IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 16, 2020, accepted August 3, 2020, date of publication August 17, 2020, date of current version August 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017210

Chaotic Dynamically Dimensioned Search Algorithm

YUN WU[1,2], JUNFENG LIU[1], AND DINGFANG LI[1]

[1]School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
[2]College of Science, Jiujiang University, Jiujiang 332005, China

Corresponding author: Dingfang Li (whudfli@163.com)

This work was supported in part by the National Key Research and Development Plan of China under Grant 2016YFC0402207 and Grant 2017YFC0405901, in part by the National Natural Science Foundation of China under Grant 51679143, in part by the Scientific and Technological Project of Jiangxi Education Office of China under Grant GJJ151078, and in part by the Scientific Research Project of Jiujiang University of China under Grant 2015LGYB18.

ABSTRACT Dynamically dimensioned search algorithm (DDS) has been proved to be an effective algorithm for solving optimization problems in actual applications, such as distributed watershed model calibration, reservoir operation, pump-and-treat problem, and Radiation Therapy (IMRT) beam angle optimization. However, it always traps in local optimal, especially for multimodal problems. To further improve the performance of DDS, this article proposes a novel chaotic dynamically dimensioned search algorithm (CDDS) by incorporating chaos theory into DDS. The improvement is performed by using three strategies: chaotic initialization, a new Gaussian mutation operator, and a chaotic search. The chaotic map generates an initial population to improve the quality of the initial solution. Chaotic initialization can enhance the exploration ability of DDS because of its intrinsic ergodic and stochastic property. A new Gaussian mutation operator is used to increase convergence speed and the exploitation ability. Meanwhile the chaotic search is utilized to jump out of the local optimal. The optimal CDDS among 13 CDDS variants is compared with other optimization algorithms on 20 classical benchmark test functions, 14 shifted rotated benchmark functions from CEC2005, and 30 benchmark functions from CEC2014 in terms of exploitation and exploration. We testify the applicability and effectiveness on a cascade reservoirs operation optimization problem in real world tasks. The experimental results and analyses demonstrate the superiority of the proposed algorithm in increasing the solution quality and accelerating the convergence.

INDEX TERMS Chaos theory, dynamically dimensioned search algorithm, exploration, exploitation.

I. INTRODUCTION

In the past decades, meta-heuristic optimization algorithms have received growing attention due to their gradient-free, simplicity, and flexibility. Some novel meta-heuristic optimization algorithms were proposed such as simulated annealing (SA) [1], tabu search (TS) [2], genetic algorithm (GA) [3], differential evolution (DE) [4], particle swarm optimization (PSO) [5], ant colony optimization (ACO) [6], harmony search (HS) [7], gravitational search algorithm (GSA) [8], cuckoo search (CS) [9], bat algorithm (BA) [10], krill herd algorithm (KH) [11], Jaya algorithm(JAYA) [12], sine cosine algorithm (SCA) [13], salp swarm algorithm (SSA) [14], and harris hawks optimization (HHO) [15]. There are also many hybrids to improve the performance of various algorithms such as self-organizing hierarchical PSO (HPSO-TVAC) [16], distance-based locally informed PSO (LIPS) [17], EPS-dPSO [18], ARA_E -SOM+BCO [19], ensemble particle swarm optimizer (EPSO) [20], and ensemble of differential evolution variants (EDEV) [21].

Generally speaking, meta-heuristic optimization algorithms can be classified into two main categories of single-solution-based and population-based algorithms. The single-solution-based algorithms generate only one solution in each iteration throughout the search process. They have the advantages of fast convergence speed, low computational cost and few function evaluations, but they may be easy to fall into local optimum. TS and SA are two typical representative single-solution-based algorithms. TS has the characteristics of a adaptive memory strategy and undertakes to jump from the local optimum by using tabu lists. SA is inspired by the annealing process and can accept a candidate solution which

The associate editor coordinating the review of this manuscript and approving it for publication was Behnam Mohammadi-Ivatloo.

is worse than the current solution with a certain probability. In recent years, iterated local search (ILS) [22], variable neighborhood search (VNS) [23] and vortex search (VS) algorithm [24] have been developed. The population-based algorithms generate an initial population including diverse solutions randomly and then improve solutions through different operators during the search process. They have the advantages of self-organization, parallelism, and the superior ability to avoid local optimum, accompanied by higher computational costs and function evaluations. The most popular population-based algorithms are: GA, which is inspired by Darwin's theory of biological evolution; PSO, which imitates the intelligent social behavior of birds flocking; DE, which finds the optimum by differential mutation operator. Recently, numerous novel population-based algorithms were proposed, some representative algorithms are as follow: grey wolf optimizer (GWO) [25], which is inspired by the leadership and hunting behavior of grey wolves in nature; whale optimization algorithm (WOA) [26], which mimics the hunting behavior of humpback whales; wild goats algorithm [27], which mimics wild goats' mountain climbing; HHO, which imitates the cooperative behavior and chasing style of Harris' hawks.

Every meta-heuristic optimization algorithm can divide the search process into exploration and exploitation phases. "Exploration phase is the process of visiting entirely new regions of the search space, while exploitation phase is the process of visiting those regions of a search space within the neighborhood of previously visited points" [28]. In fact, an optimization algorithm randomly explores the search space as vast as possible in the exploration phase, and efficiently exploits the most promising regions in exploitation phase. Notice that emphasizing on exploration will be unable to find the global optimal and slow down the convergence rate, while emphasizing on exploitation will make the algorithm getting stick into local optimal. Owing to that exploration and exploitation conflict each other, a successful meta-heuristic optimization algorithm needs to properly balance exploration and exploitation. Many strategies have been employed to improve the performance and overcome shortcomings, of which chaos theory is an effective method to boost both exploration and exploitation.

Chaos is a bounded unstable dynamic behavior and has been applied in many scientific fields such as mathematics, geology, biology, computer science, economics, and engineering. Introducing chaotic theory into the optimization methods is one of the most famous applications. Chaos optimization algorithms (COA) [29], [30] is proposed as a novel optimization algorithm by using chaotic variables instead of random variables in random-based optimization algorithm. The search process of COA consists of the coarse search and the fine search, which just plays the role of exploration and exploitation. The chaos theory has also been combined with meta-heuristic optimization algorithms by using chaotic maps for parameter adaptation, such as chaos embedded particle swarm optimization algorithms (CEPSOAs) [31], chaotic bee colony algorithm [32], chaotic harmony search algorithm [33], chaotic bat algorithm [34], chaotic krill herd algorithm [35], and chaotic gravitational search algorithm [36]. In addition, chaotic initialization and chaotic search (chaotic perturbation) are utilized instead of random initialization and random perturbation in [37]–[40] to enhance the performance of the meta-heuristics algorithms.

A novel meta-heuristic optimization algorithm named dynamically dimensioned search (DDS) algorithm was proposed to find good global solutions for automatic calibration of watershed simulation models [41]. DDS is a simple single-solution-based optimization algorithm with no algorithm parameter tuning, furthermore, it has the advantages of fast convergence speed, low computational cost and few function evaluations. It has been proved its superior performance compared to some well-known algorithms [42], [43]. According to the analysis above, DDS is particularly suitable for actual applications, such as distributed watershed model calibration [41], [44]–[49], reservoir operation [50], pump-and-treat problem [51], and Radiation Therapy (IMRT) beam angle optimization [52].

The search process of DDS is composed by two stages: initial stage and updating stage. In initial stage, an initial solution is obtained from initial population which is generated using uniform distribution. Then a candidate solution is updated through mutation operator and reflection operator in later stage. Mutation probability, which is decreasing with the growing iterations, plays a significant role in balancing exploration (global search) and exploitation (local search). The conversion from global search to local search is executed by dynamically and probabilistically reducing the number of dimensions to be mutated of the current best solution. To address the discrete and mixed optimization problems, discrete dynamically dimensioned search (D-DDS) and hybrid discrete dynamically dimensioned search (HD-DDS) algorithm were presented in [53], [54].

In addition, there are some strategies employed to improve the performance of DDS. In [55], the only algorithm parameter r was set to 0.2 at the beginning of the search process to highlight exploration, and then it decreased gradually to 0.05 in the light of different mutation probability values to enhance its exploitation. [45] updated the current best solution according to a certain dynamic probability while new best solution quality is no better than the previous one. To improve the exploitation, mutation probability was integrated into r and the value of r reduces from 0.3 to 0.05 to accelerate the rate of convergence. DYCORS-DDSRBF, an RBF-assisted modification of DDS, was originally proposed in [44] for surrogate-based optimization of high-dimensional expensive black-box functions. Numerical results verified the effectiveness of DYCORS-DDSRBF on a 14-D watershed calibration problem, eleven 30-D and 200-D test problems. A dynamic change in the current standard deviation based on successive successful or unsuccessful iterations was introduced in [52]. In [46], two selection of sampling distributions and different sampling ranges were utilized in DDS for hydrologic and water quality predictions to improve the performance of DDS. Then the DDS algorithm combined with filter method to solve nonlinear, non-smooth and non-convex constrained global optimization problems in [56]. In [43], a new mutation probability function and a Gaussian mutation operator were employed in DDS to boost the exploitation of DDS, respectively.

Based on the above researches, we can find that the performance and convergence of DDS are affected by several factors. For example, the proper initial solution can help fast find the global optimal solution, mutation probability controls the transition between exploration and exploitation, different mutation operators and parameter settings enable DDS to perform different exploitation and exploration capabilities. We have utilized some mechanisms in the preliminary experiments to find a appropriate approach to improve the performance of DDS. Then, we compared the experimental result obtained by DDS with different mechanisms under the same experimental condition. By analyzing the experimental data, we found that a new DDS variant with chaotic initialization, a new Gaussian mutation operator, and chaotic search performs best. Firstly, we use a new Gaussian mutation operator to improve the exploitation ability and the local search of DDS based on the previous research [43]. The experiments prove that the new Gaussian mutation operator can obtain good performance but may be relaxed to fall into local optimum for some multi-modal functions. Thus, we need to increase the ability of exploration of DDS. According to the mechanism of DDS, we consider the improvement in two ways: initial stage and updating stage. For the improvement of the initial solution, chaotic initialization performs best. Chaotic search (chaotic perturbation) can help the algorithm to jump from local optimum and enhance the ability of exploration. Therefore, we select these three technologies to improve DDS.

In this article, a novel chaotic dynamically dimensioned search algorithm (CDDS) is proposed. As we know, this is the first attempt to introduce chaos theory into DDS to improve the performance. Considering the non-repetition and stochastic of the chaos, we use a chaotic map for generating an initial population to enhance the exploration of DDS, then an initial solution is selected from the initial population. To enhance the exploitation of DDS, a new Gaussian mutation operator is employed to make full use of the information of current best solution for perturbation and obtains high accuracy solution and fast convergence. Because strong exploitation ability makes the algorithm easily trapped in local optimal, chaotic search (chaotic perturbation) is utilized to escape from the local optimal and improve the exploration ability.

To verify the efficiency of the CDDS algorithm, we choose 13 one-dimensional chaotic maps to construct CDDS, and find Neuron map is the best chaotic map for improving the performance of DDS significantly. The optimal CDDS is compared with some well-known optimization algorithms on 20 classical benchmark test functions, the first 14 benchmark test functions from IEEE CEC2005 and 30 benchmark functions from CEC2014, respectively. The results of experiments demonstrate that the overall performance of CDDS is superior to that of the compared algorithms in terms of solution accuracy and convergence speed. We also study the influence of three strategies through two aspects by a series of experiments. In order to investigate the availability and effectiveness of CDDS, a cascade reservoirs operation optimization problem in the real world is conducted as a case study.

The remainder of the paper is organized as follows. Section II presents the main inspiration and motivation of the proposed algorithm. Section III presents a brief introduction to the DDS algorithm and chaotic maps. The detail of the proposed algorithm is provided in Section IV and a set of experimental results with various statistical analyses are given in Section V. Section VI discusses the effect of three strategies through a series of experiments. The optimal operation problem of cascade hydropower stations is solved by CDDS in Section VII. Eventually, Section VIII includes the conclusions and future works.

II. NEED FOR RESEARCH

Hydropower is a type of clean, stable, renewable and green energy. As cascade reservoirs develop, hydropower is significant for national economy construction. The operation of cascaded hydropower stations do not only base on compensation coordination among the reservoirs, but also on each hydro-generating unit's efficiency. It is an intricate task affected by many factors. Thus, cascade reservoirs operation optimization (CROO) has become a popular research area all over the world.

Numerous conventional optimization methods have been employed for CROO, such as Linear Programming (LP), Nonlinear Programming (NLP), Dynamic Programming (DP), Quadratic Programming (QP). However, with the growing size of reservoirs, the "curse of dimensionality" is presented in the solving procedure. In recent years, various meta-heuristic optimization algorithms have been applied to avoid the "curse of dimensionality", such as GA, PSO, ACO, and DE. Nevertheless, the performance of these meta-heuristic algorithms is unsatisfactory in some cases. Consequently, researchers constantly develop new algorithms to seek better performance in terms of accuracy and efficiency. In view of the fact that CROO is a high-dimension, nonlinear constrained, complicated optimization problem, it is necessary to introduce other strong feasible, practicable and applicable meta-heuristic algorithms.

III. PRELIMINARIES

This section briefly provides an introduction about the dynamically dimensioned search (DDS) algorithm and the chaotic maps.

A. THE DDS ALGORITHM

DDS is a single-solution based heuristic neighborhood search algorithm and has been produced by

Tolson and Shoemaker [41]. It is designed for finding good global solutions within a specified maximum function evaluation limit, and requires no algorithm parameter tuning. It searches globally at the start of the search and transits to a more local search as the number of function evaluations increases. The adjustment from global to local search is achieved by dynamically and probabilistically reducing the number of dimensions in the neighborhood. The candidate solution is created by perturbing the current best solution in the randomly selected dimensions only. Reflection operator in Algorithm 1 ensures decision variable boundaries are more easily respected. DDS is a greedy algorithm because the best solution is never updated with a candidate solution that has an inferior value of the objective function.

Algorithm 1 Reflection Operator of DDS

Input: x_j^{new}, x_j^{min} and x_j^{max}
if ($x_j^{new} < x_j^{min}$) **then**
  if (U(0,1)≤0.5) **then**
    $x_j^{new} = x_j^{min} + (x_j^{min} - x_j^{new})$;
  else
    $x_j^{new} = x_j^{min}$;
  end if
  if $x_j^{new} > x_j^{max}$ **then**
    $x_j^{new} = x_j^{min}$;
  end if
end if
if ($x_j^{new} > x_j^{max}$) **then**
  if (U(0,1)≤0.5) **then**
    $x_j^{new} = x_j^{max} - (x_j^{new} - x_j^{max})$;
  else
    $x_j^{new} = x_j^{max}$;
  end if
  if $x_j^{new} < x_j^{min}$ **then**
    $x_j^{new} = x_j^{max}$;
  end if
end if
Return x_j^{new}.

Over the course of iteration, the search process consists of mutation operator, reflection operator, and selection operator. The mutation operator, controlled by a mutation probability (P), plays an important role in the search process. As to the current best solution $x^{best} = (x_1^{best}, x_2^{best}, \cdots, x_D^{best})$, a candidate solution $x^{new} = (x_1^{new}, x_2^{new}, \cdots, x_D^{new})$ is generated according to

$$x_j^{new} = \begin{cases} x_j^{best} + & r(x_j^{max} - x_j^{min})N(0,1), \\ & if\ r(j) < P(i)\ or\ j = rn(i) \\ x_j^{best}, & otherwise \end{cases} \tag{1}$$

In (1), r is recommended as 0.2, $N(0, 1)$ is a standard normally distributed random number. $r(j) \in [0, 1]$ is the jth evaluation of a uniform random number in the ith iteration. $P(i) = 1 - ln(i)/ln(m) \in [0, 1]$ is a monotonically decreasing function of iteration number i and m is the maximum number of iterations. $rn(i)$ is a randomly chosen index $\in \{1, 2, \cdots, D\}$ which ensures that x^{new} is different from x^{best} in at least one dimension.

In order to ensure that each one-dimensional mutative value in a candidate solution respects the bounds, the minimum and maximum values of decision variable act as reflecting boundaries. These details can be described as shown in Algorithm 1.

To decide whether or not x^{best} should be updated, the current best solution x^{best} is compared to the candidate solution x^{new} using the greedy criterion as follows.

$$x^{best} = \begin{cases} x^{new}, & if \quad F(x^{new}) \le F(x^{best}) \\ x^{best}, & otherwise \end{cases} \tag{2}$$

Finally, the DDS algorithm is terminated by satisfying a stop criterion.

B. CHAOTIC MAPS

The chaotic systems are nonlinear deterministic dynamical systems that they have important characteristics, including (1) ergodicity, (2) stochasticity, (3) regularity, (4) initial value sensitivity. Due to these characteristics, chaos is a reliable source of randomness, which is non-period, non-converging and bounded. In recent years, a lot of chaotic maps were utilized for solving optimization problems. One-dimensional non-invertible chaotic maps are the simplest systems with capability of providing chaotic behaviors. Here, we will introduce some of well-known one-dimensional chaotic maps to be used in later experiments.

1) LOGISTIC MAP

This classic logistic map was proposed by Robert M.May in 1976 [57], and can be written as

$$x_{k+1} = ax_k(1 - x_k) \tag{3}$$

In the equation, x_k is the kth chaotic number where k denotes the iteration number. Obviously, $x_k \in (0, 1)$ under the conditions that the initial $x_0 \in (0, 1)$. In later experiments, $a = 4$ is used and $x_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$.

This map generates sequences in (0, 1).

2) CHEBYSHEV MAP

The Chebyshev map can be defined as follows [34]:

$$x_{k+1} = cos(kcos^{-1}(x_k)) \tag{4}$$

This map generates sequences in (-1, 1).

3) CIRCLE MAP

This map is formulated as [58]:

$$x_{k+1} = x_k + b - (\frac{a}{2\pi})sin(2\pi x_k)mod(1) \tag{5}$$

For $a = 0.5$ and $b = 0.2$, it generates chaotic sequence in (0, 1).

4) GAUSS/MOUSE MAP

The Gauss map is represented by [59]:

$$x_{k+1} = \begin{cases} 0, & x_k = 0 \\ \dfrac{1}{x_k mod(1)}, & otherwise \end{cases} \tag{6}$$

and $\frac{1}{x_k mod(1)} = \frac{1}{x_k} - [\frac{1}{x_k}]$.

This map generates sequences in (0, 1).

5) ITERATIVE MAP

The interative map is defined as follows [36]:

$$x_{k+1} = sin(\frac{a\pi}{x_k}) \tag{7}$$

where $a \in (0, 1)$ is a suitable parameter, and $a = 0.7$ has been used in the experiments.

This map generates sequences in $(-1, 1)$.

6) PIECEWISE MAP

The piecewise map [60] can be written as:

$$x_{k+1} = \begin{cases} \dfrac{x_k}{a}, & 0 \leq x_k < a \\ \dfrac{x_k - a}{0.5 - a}, & a \leq x_k < 0.5 \\ \dfrac{1 - a - x_k}{0.5 - a}, & 0.5 \leq x_k < 1 - a \\ \dfrac{1 - x_k}{a}, & 1 - a \leq x_k < 1 \end{cases} \tag{8}$$

where $a \in (0, 0.5)$ is a control parameter, and $a = 0.4$ has been used in the experiments.

This map generates sequences in (0, 1).

7) SINE MAP

As a unimodal map, the sine map can be defined by [36]:

$$x_{k+1} = \frac{a}{4} sin(\pi x_k) \tag{9}$$

where $a \in (0, 4]$ and $a = 4$ is used in later experiments.

This map generates sequences in (0, 1).

8) SINGER MAP

The following equation defines the singer map [61]:

$$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4) \tag{10}$$

where $\mu \in (0.9, 1.08)$ is a control parameter, and $\mu = 1.07$ has been used in the experiments.

This map generates sequences in (0, 1).

9) SINUSOIDAL MAP

This map can be given by [31], [62]

$$x_{k+1} = ax_k^2 sin(\pi x_k) \tag{11}$$

In later experiments, $a = 2.3$ is used.

This map generates sequences in (0, 1).

10) TENT MAP

The map is represented by [35]:

$$x_{k+1} = \begin{cases} \dfrac{x_k}{0.7}, & x_k < 0.7 \\ \dfrac{10}{3}(1 - x_k), & x_k \geq 0.7 \end{cases} \tag{12}$$

This map generates sequences in (0, 1).

11) CUBIC MAP

The equation of the cubic map [63] can be represented as:

$$x_{k+1} = \rho x_k(1 - x_k^2) \tag{13}$$

$\rho = 2.59$ has been used in the experiments.

This map generates sequences in (0, 1).

12) SINUSOIDAL MAP

This map can be represented as [33]:

$$x_{k+1} = 2.3x_k^2 sin(\pi x_k) \tag{14}$$

This map generates sequences in (0, 1).

13) NEURON MAP

This map is defined as [30]:

$$x_{k+1} = \eta - 2tanh(\gamma)exp(-3x_k^2) \tag{15}$$

where η denotes the attenuation factor($0 \leq \eta \leq 1$), γ is the proportionality factor. $\eta = 0.5$ and $\gamma = 5$ have been used in the experiments.

This map generates sequences in $(-1.5, 0.5)$.

IV. CHAOTIC DYNAMICALLY DIMENSIONED SEARCH ALGORITHM

The chaotic dynamically dimensioned search algorithm (CDDS) is conducted through chaotic initialization, a new Gaussian mutation operator, and a chaotic search. 13 chaotic maps generate an initial population to enhance the exploration of the algorithm and improve the quality of the initial solution, respectively. The new Gaussian mutation operator focuses on local search around the current best solution to enhance the exploitation ability for higher precision solution and faster convergence rate. Moreover, CDDS employs the chaotic search in the process of search, which prevents the algorithm from falling into local optima to some extent. The stages of the proposed algorithm are described in Figure 1.

A. INITIAL STAGE

In this stage, the initial solution is generated by iterating the selected chaotic map as shown in Algorithm 2.

As the chaos has the characteristics of ergodicity and non-repetition, the chaotic sequences provided by the selected chaotic map were embedded into the original DDS algorithm to generate the initial population. Therefore, instead of generating the population using stochastic search,

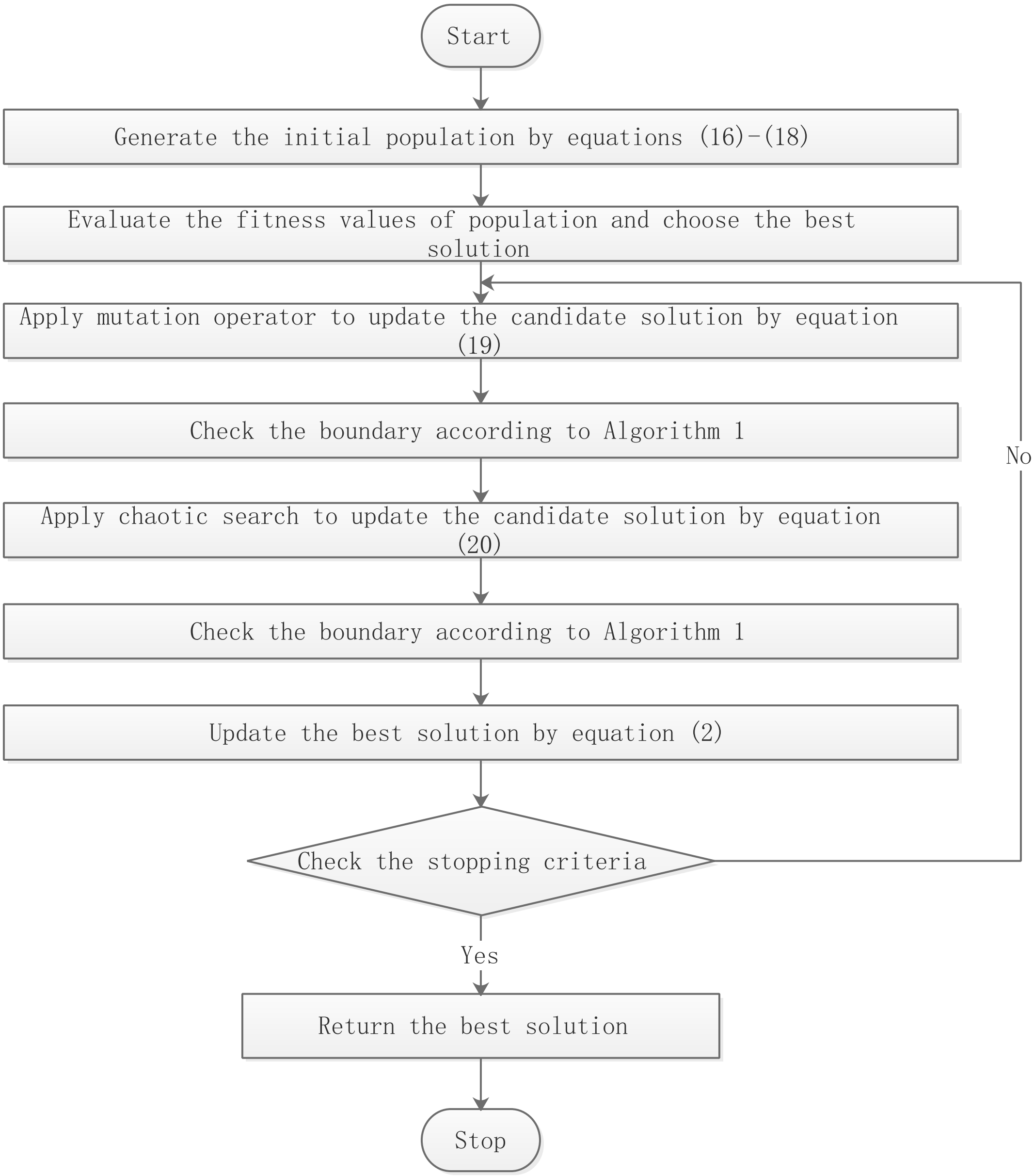


FIGURE 1. **Flowchart of the CDDS algorithm.**

13 one-dimensional chaotic maps are used to implement global search using the following equations:

$$x_i^{(k)} = x_i^{min} + (x_i^{max} - x_i^{min})c_i^{(k)} \tag{16}$$

$$x_i^{(k)} = \frac{(x_i^{max} + x_i^{min})}{2} + \frac{(x_i^{max} - x_i^{min})}{2}c_i^{(k)} \tag{17}$$

$$x_i^{(k)} = \frac{(1.5x_i^{max} + 0.5x_i^{min})}{2} + \frac{(x_i^{max} - x_i^{min})}{2}c_i^{(k)} \tag{18}$$

Equations (16)-(18) are utilized to map the chaotic variables $c_i^{(k)}$ into the optimization variables $x_i^{(k)}$ in the light of the chaotic map with different scaled values, where $i = 1, 2, \cdots, D$ and $k = 1, 2, \cdots, p$. (16) is suitable for most chaotic maps except Chebyshev, Iterative and Neuron; (17) is suitable for Chebyshev and Iterative, meanwhile (18) is suitable for Neuron.

After that, the fitness function value of each initial variable is computed, and the best variable that has the smallest fitness function value is selected from the initial population to represent the initial solution.

B. UPDATING STAGE

The solution is updated in this stage through two steps. A new Gaussian mutation operator is used in the first step, and a chaotic search is used in the second step.

1) UPDATING USING A NEW GAUSSIAN MUTATION OPERATOR

The new Gaussian mutation adds a random Gaussian vector to the current best solution by the following

Algorithm 2 Chaotic Initialization of CDDS

p = the maximum number of chaotic initialization iteration
Randomly initialize the first chaotic variable
for $k = 1 : p$ **do**
  Generate initial solution by chaotic variable using (16)-(18)
  Update chaotic variable c^k according to the selected chaotic map
end for
Sort the initial population and choose the best solution x^{best}
Set $x^0 = x^{best}$.

equation:

$$x_j^{new} = \begin{cases} x_j^{best} + & r \cdot x_j^{best} N(0, 1), \\ & if\ r(j) < P(i)\ or\ j = rn(i) \\ x_j^{best}, & otherwise \end{cases} \quad (19)$$

r is the scalar neighborhood size perturbation parameter assigning the same variation to each decision variable, which is recommended as 0.2.

The perturbation variance of CDDS in the randomly selected dimensions makes full use of the information of the current best solution for disturbance. The new Gaussian mutation operator focuses on local search around the current best solution. It can enhance the quality of candidate solution, improve the search speed and exhibit good local search ability.

2) UPDATING USING CHAOTIC SEARCH (CHAOTIC PERTURBATION)

In general, the new Gaussian mutation operator lends itself strongly to exploitation. However, it could not always implement global search well. Sometimes it fails to find global optimal solution in some cases [43]. Thus, the chaotic search (chaotic perturbation) is used to enhance the exploration ability. More details of the chaotic search are shown in Algorithm 3.

Algorithm 3 Chaotic Search of CDDS

q = the maximum number of chaotic search iteration.
Randomly initialize the first chaotic variable.
for $k = 1 : q$ **do**
  Chaotic iteration variable is transformed into optimization variable using (16)-(18)
  Add chaotic perturbation to the candidate solution using (20)
  Update chaotic variable c^k according to the selected chaotic map
end for
Evaluate the fitness value of the candidate solution x^{new}.

Chaotic search, the fine search of COA, is a random movement with ergodicity, pseudo-randomness and regularity. It could escape from local minima and approach the global minima. In the updating process of the current best solution, the candidate solution would be perturbed sufficiently to jump out local optimization using the following equation:

$$x_j^{new} = x_j^{new} + \beta \cdot \delta_j \quad (20)$$

where β is a small positive number and δ_j is the chaotic perturbation. We set β to be 0.00001 by some experiments tests.

V. NUMERICAL SIMULATION AND RESULTS

In order to evaluate the performance of the proposed algorithm, 64 benchmark functions are employed in this section. Tables 1-4 present these functions, where "Search range" is the ranges of the variables, "Minimum" is the optimum and D is the dimension. The test functions in Tables 1-2 are classical test problems, which are classified in two groups: unimodal and multimodal. Notice that the unimodal functions have only one global optimum so they are suitable for investigating the exploitation capability, and the multimodal functions with several local optima are suitable to test the exploration ability. CEC2005 and CEC2014 are shown in Tables 3-4 to further testify the effectivity of CDDS.

A. CDDS WITH DIFFERENT CHAOTIC MAPS

Different chaotic maps have different mechanisms, thus different chaotic DDS variants have different performance. In order to choose the optimal chaotic DDS, 13 chaotic DDS variants are benchmarked using 20 well-known benchmark functions. We set the dimension and the maximum number of function evaluations(FEs) to 30 and 10000, respectively. Tables 5 and 6 illustrate the average (mean), standard deviation (std), success rate (SR) of the obtained best fitness function values averaging over 100 independent runs. The best results are indicated in bold type.

Success rate (SR) is defined as follows

$$SR = 100 \times \frac{N_{successful}}{N_{all}} \quad (21)$$

where N_{all} is the number of runs, and $N_{successful}$ is the number of runs which can find the solution successfully. A run is considered as a successful run if the solution meets $|f_{best} - f^*| < \varepsilon$, where f_{best} is the optimum provided by the algorithm and f^* is the theoretical optimum. The value-to-reach (VTR) is set to be 10^{-5}. To evaluate the performance of the algorithms, we ranked the algorithms in Tables 5 and 6, which also offered the best/worst ranking, rank sum and overall rank of each algorithm. Note that CDDS1 to CDDS13 utilized Logistic, Chebyshev, Circle, Gauss/Mouse, Iterative, Piecewise, Sine, Singer, Sinusoidal, Tent, Cubic, Sinus and Neuron, respectively.

As depicted in Tables 5 and 6, CDDS13 (Neuron map) provides the best performance on the majority of the benchmark

TABLE 1. Summary of classical unimodal benchmark functions.

Name	Equation	Search range	Minimum
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-10, 10]^D$	0
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	0
Step	$f_3(x) = \sum_{i=1}^{D}[x_i + 0.5]^2$	$[-10, 10]^D$	0
Sum-Square	$f_4(x) = \sum_{i=1}^{D} ix_i^2$	$[-10, 10]^D$	0
Schwefel's 1.2	$f_5(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$	0
Schwefel's 2.21	$f_6(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	$[-100, 100]^D$	0
Schwefel's 2.22	$f_7(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]^D$	0
Quartic with noise	$f_8(x) = \sum_{i=1}^{D} ix_i^4 + random[0, 1)$	$[-1.28, 1.28]^D$	0
Sum-Power	$f_9(x) = \sum_{i=1}^{D} \lvert x_i\rvert^{(i+1)}$	$[-1, 1]^D$	0

TABLE 2. Summary of classical multimodal benchmark functions.

Name	Equation	Search range	Minimum
Griewank	$f_{10}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-500, 700]^D$	0
Ackley	$f_{11}(x) = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - exp(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)) + 20 + e$	$[-2, 2]^D$	0
Rastrigin	$f_{12}(x) = \sum_{i=1}^{D}[x_i^2 - 10cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	0
Levy	$f_{13}(x) = sin^2(\pi\omega_1) + \sum_{i=1}^{D-1}(\omega_i - 1)^2[1 + 10sin^2(\pi\omega_i + 1)]$ $+(\omega_D - 1)^2[1 + sin^2(2\pi\omega_D)]$, where $\omega_i = 1 + \frac{x_i - 1}{4}$	$[-10, 10]^D$	0
Penalty ♯1	$f_{14}(x) = \frac{\pi}{n}\{10sin^2(\pi\omega_1) + \sum_{i=1}^{D-1}(\omega_i - 1)^2[1 + 10sin^2(\pi\omega_{i+1})]$ $+(\omega_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$, where $\omega_i = 1 + \frac{x_i + 1}{4}$	$[-50, 50]^D$	0
Penalty ♯2	$f_{15}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{D}(x_i - 1)^2[1 + sin^2(3\pi x_{i+1})]$ $+(x_D - 1)^2[1 + sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$	0
Zakharov	$f_{16}(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5ix_i)^2 + (\sum_{i=1}^{D} 0.5ix_i)^4$	$[-5, 10]^D$	0
Elliptic	$f_{17}(x) = \sum_{i=1}^{D}(10^6)^{(i-1)/(D-1)}x_i^2$	$[-100, 100]^D$	0
Schwefel's 2.26	$f_{18}(x) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin(\lvert x_i\rvert^{\frac{1}{2}})$	$[-500, 500]^D$	0
Alpine	$f_{19}(x) = \sum_{i=1}^{D} \lvert x_i sin(x_i) + 0.1 \cdot x_i\rvert$	$[-10, 10]^D$	0
Schaffer	$f_{20}(x) = 0.5 + \frac{sin^2(\sqrt{\sum_{i=1}^{D} x_i^2}) - 0.5}{(1 + 0.001(\sum_{i=1}^{D} x_i^2))^2}$	$[-100, 100]^D$	0

In benchmark function f_{14} and f_{15}, the definition of the function $u(x_i, a, k, m)$ is as follows:

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

TABLE 3. Summary of the CEC2005 test functions.

Test functions	Initialization Range	Search range	Minimum
F_1: Shifted Sphere Function	$[-100, 100]^D$	$[-100, 100]^D$	-450
F_2: Shifted Schwefel's Problem 1.2	$[-100, 100]^D$	$[-100, 100]^D$	-450
F_3: Shifted Rotated High Conditioned Elliptic Function	$[-100, 100]^D$	$[-100, 100]^D$	-450
F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness	$[-100, 100]^D$	$[-100, 100]^D$	-450
F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds	$[-100, 100]^D$	$[-100, 100]^D$	-310
F_6: Shifted Rosenbrock's Function	$[-100, 100]^D$	$[-100, 100]^D$	390
F_7: Shifted Rotated Griewank's Function without Bounds	$[0, 600]^D$	$[-600, 600]^D$	-180
F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds	$[-32, 32]^D$	$[-32, 32]^D$	-140
F_9: Shifted Rastrigin's Function	$[-5, 5]^D$	$[-5, 5]^D$	-330
F_{10}: Shifted Rotated Rastrigin's Function	$[-5, 5]^D$	$[-5, 5]^D$	-330
F_{11}: Shifted Rotated Weierstrass Function	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	90
F_{12}: Schwefel's Problem 2.13	$[-100, 100]^D$	$[-100, 100]^D$	-460
F_{13}: Expanded Extended Griewank's plus Rosenbrock's Function	$[-3, 1]^D$	$[-3, 1]^D$	-130
F_{14}: Shifted Rotated Expanded Scaffer's F6	$[-100, 100]^D$	$[-100, 100]^D$	-300

functions and obtains the highest rank. For unimodal benchmark functions, CDDS13 performs best in 5 out 9 functions, and provides two second-best results. The success rate of CDDS13 is higher than those of other variants. Inspecting the results in Table 5, all the CDDS variants significantly provide better results than DDS in 7 out of 9 functions. The success rates of different CDDS variants are higher than the success rate of DDS, meaning that the chaotic maps can improve the "exploitation" of DDS.

For multimodal benchmark functions, CDDS13 performs best in 4 out 11 functions and obtains the highest rank. The success rate of CDDS13 is in the top three of all the

TABLE 4. Summary of the CEC2014 test functions.

Types	Test functions	Search range	Minimum
Unimodal Functions	$F1_{CEC2014}$:Rotated High Conditioned Elliptic Function	$[-100, 100]^D$	100
	$F2_{CEC2014}$:Rotated Bent Cigar Function	$[-100, 100]^D$	200
	$F3_{CEC2014}$:Rotated Discus Function	$[-100, 100]^D$	300
Simple Multimodal Functions	$F4_{CEC2014}$:Shifted and Rotated Rosenbrock's Function	$[-100, 100]^D$	400
	$F5_{CEC2014}$:Shifted and Rotated Ackley's Function	$[-100, 100]^D$	500
	$F6_{CEC2014}$:Shifted and Rotated Weierstrass Function	$[-100, 100]^D$	600
	$F7_{CEC2014}$:Shifted and Rotated Griewank's Function	$[-100, 100]^D$	700
	$F8_{CEC2014}$:Shifted Rastrigin's Function	$[-100, 100]^D$	800
	$F9_{CEC2014}$:Shifted and Rotated Rastrigin's Function	$[-100, 100]^D$	900
	$F10_{CEC2014}$:Shifted Schwefel's Function	$[-100, 100]^D$	1000
	$F11_{CEC2014}$:Shifted and Rotated Schwefel's Function	$[-100, 100]^D$	1100
	$F12_{CEC2014}$:Shifted and Rotated Katsuura Function	$[-100, 100]^D$	1200
	$F13_{CEC2014}$:Shifted and Rotated HappyCat Function	$[-100, 100]^D$	1300
	$F14_{CEC2014}$:Shifted and Rotated HGBat Function	$[-100, 100]^D$	1400
	$F15_{CEC2014}$:Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	$[-100, 100]^D$	150
	$F16_{CEC2014}$:Shifted and Rotated Expanded Scaffer's F6 Function	$[-100, 100]^D$	1600
Hybrid Function 1	$F17_{CEC2014}$:Hybrid Function 1 (N=3)	$[-100, 100]^D$	1700
	$F18_{CEC2014}$:Hybrid Function 2 (N=3)	$[-100, 100]^D$	1800
	$F19_{CEC2014}$:Hybrid Function 3 (N=4)	$[-100, 100]^D$	1900
	$F20_{CEC2014}$:Hybrid Function 4 (N=4)	$[-100, 100]^D$	2000
	$F21_{CEC2014}$:Hybrid Function 5 (N=5)	$[-100, 100]^D$	2100
	$F22_{CEC2014}$:Hybrid Function 6 (N=5)	$[-100, 100]^D$	2200
Composition Functions	$F23_{CEC2014}$:Composition Function 1 (N=5)	$[-100, 100]^D$	2300
	$F24_{CEC2014}$:Composition Function 2 (N=3)	$[-100, 100]^D$	2400
	$F25_{CEC2014}$:Composition Function 3 (N=3)	$[-100, 100]^D$	2500
	$F26_{CEC2014}$:Composition Function 4 (N=5)	$[-100, 100]^D$	2600
	$F27_{CEC2014}$:Composition Function 5 (N=5)	$[-100, 100]^D$	2700
	$F28_{CEC2014}$:Composition Function 6 (N=5)	$[-100, 100]^D$	2800
	$F29_{CEC2014}$:Composition Function 7 (N=3)	$[-100, 100]^D$	2900
	$F30_{CEC2014}$:Composition Function 8 (N=3)	$[-100, 100]^D$	3000

algorithms on multimodal benchmark functions except f_{13}, f_{15} and f_{16}. It evidences that CDDS13 is capable of avoiding local minima and can improve the "exploration" of DDS. Comprehensive consideration suggests that we choose Neuron map as the final optimal map to form the best chaotic DDS (CDDS).

B. GENERAL PERFORMANCE OF CDDS ON CLASSICAL BENCHMARK FUNCTIONS

In order to have an all-sided investigation of the performance of the CDDS algorithm, three test sets were used to carry out experiments. Firstly, we compare it with nine optimization algorithms for solving 20 well-known benchmark functions to prove the superiority of the proposed CDDS algorithm. The compared algorithms are DE [4], PSO [5], BA [10], SCA [13], GSA [8], KH [11], JAYA [12], EPSO [20], and EDEV [21], noticing that JAYA is a parameter-less algorithm. The parameter settings for these algorithms are shown in Table 7. In experiments, population size and maximum number of function evaluations (FEs) were set to 50 and 10000, respectively. The results after 100 runs were presented in Tables 8 and 9. The best results were highlighted in bold.

Non-parametric Wilcoxon signed-rank test at 5% significance level was conducted between the results of proposed CDDS and the results of the compared algorithms. The symbols "+" or "−" represent that CDDS is significantly better than or significantly worse than the compared algorithms, respectively. The symbol "=" represents that there is no significant difference between CDDS and the compared algorithms. The algorithms are ranked according to their mean error values. The number of best/worst ranking and the number of "+" were counted for each algorithm. These statistical results and the overall rank for each algorithm were both summarized in the last four rows in Tables 8 and 9.

Table 8 shows that CDDS performs best in 8 out 9 functions (f_1-f_7 and f_9), which proves CDDS to have the good exploitation ability. KH gets the best performance on f_8. As to Wilcoxon signed rank test, CDDS is significantly better than DE, PSO, BA, SCA, GSA, KH, JAYA, EPSO and EDEV on 9, 8, 9, 9, 7, 8, 9, 9 and 8 test functions, respectively.

Table 9 shows that CDDS outperforms other algorithms in 5 out 11 functions. GSA and EDEV both perform best on two of the eleven functions. KH and EPSO have the best performance on f_{15} and f_{20}, respectively. With regard to Wilcoxon signed rank test, CDDS significantly outperforms DE, PSO, BA, SCA, GSA, KH, JAYA, EPSO and EDEV on 9, 9, 11, 10, 8, 6, 11, 7 and 6 test functions, respectively. Taken together, CDDS performs better than other algorithms on the majority of test functions, and has the highest rank. EPSO and EDEV are the second most effective.

To investigate the convergence speed of the algorithms, the convergence curves were depicted in Figures 2 and 3. The values in these figures are the average best objective function values achieved from 100 independent runs. It should

TABLE 5. Comparison of experimental results by CDDS algorithm with thirteen different chaotic maps on classical unimodal benchmark functions.

Func	Criteria	DDS	CDDS1	CDDS2	CDDS3	CDDS4	CDDS5	CDDS6	CDDS7	CDDS8	CDDS9	CDDS10	CDDS11	CDDS12	CDDS13
f_1	mean	8.37e-02	2.46e-42	1.70e-41	4.97e-42	1.47e-41	2.56e-42	3.78e-40	**1.51e-42**	3.78e-40	3.12e-40	1.82e-40	2.82e-42	1.41e-40	1.75e-42
	std	3.48e-02	1.56e-41	1.04e-40	1.18e-41	4.91e-41	1.86e-41	1.71e-39	6.73e-42	1.08e-39	1.15e-39	9.87e-40	9.35e-42	4.81e-40	**5.02e-42**
	SR	0	100	100	100	100	100	100	100	100	100	100	100	100	100
	rank	14	3	8	6	7	4	13	1	12	11	10	5	9	2
f_2	mean	4.77e+02	**1.33e+00**	1.21e+01	1.77e+01	2.18e+01	6.43e+00	2.69e+01	5.05e+00	2.85e+01	3.54e+01	2.36e+01	2.68e+00	2.90e+01	1.04e+01
	std	5.34e+02	1.87e+00	1.81e+01	9.02e-01	1.88e+01	2.55e+01	2.33e+01	1.83e+00	4.81e+01	6.38e+01	**7.10e-01**	1.32e+01	3.84e+01	2.62e+01
	SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	rank	14	1	6	7	8	4	10	3	11	13	9	2	12	5
f_3	mean	7.67e-02	5.51e-08	2.50e-01	3.23e-08	4.55e-08	4.94e-08	5.75e-08	5.55e-08	6.35e-08	6.32e-08	4.73e-08	5.43e-08	8.32e-08	**3.00e-08**
	std	2.73e-02	7.02e-09	6.32e-07	3.72e-09	6.03e-09	4.92e-09	6.39e-09	7.23e-09	8.17e-09	9.82e-09	6.77e-09	6.93e-09	1.26e-08	**3.71e-09**
	SR	0	100	0	100	100	100	100	100	100	100	100	100	100	100
	rank	13	7	14	2	3	5	9	8	11	10	4	6	12	1
f_4	mean	1.11e+00	6.28e-41	9.09e-41	1.14e-40	4.30e-40	4.55e-41	7.31e-39	7.78e-41	6.32e-39	5.10e-39	4.56e-39	8.87e-41	1.90e-39	**4.43e-41**
	std	4.97e-01	3.87e-40	4.30e-40	2.92e-40	2.42e-39	2.19e-40	3.47e-38	5.35e-40	1.69e-38	1.48e-38	2.75e-38	3.44e-40	5.32e-39	**1.82e-40**
	SR	0	100	100	100	100	100	100	100	100	100	100	100	100	100
	rank	14	3	6	7	8	2	13	4	12	11	10	5	9	1
f_5	mean	6.19e+03	1.09e-05	6.62e+00	2.65e-03	1.39e-05	9.39e-04	6.72e-01	1.09e-05	1.78e+00	8.79e-01	2.11e-01	8.39e-02	4.19e+03	**8.81e-06**
	std	1.78e+03	5.58e-06	2.38e+01	2.18e-02	8.58e-06	9.30e-03	3.05e+00	5.52e-06	5.14e+00	3.34e+00	1.11e+00	8.39e-01	1.99e+04	**4.95e-06**
	SR	0	53	0	45	36	55	0	46	0	1	2	42	0	62
	rank	14	2	12	6	4	5	9	3	11	10	8	7	13	1
f_6	mean	6.83e+00	6.28e-10	4.54e-09	9.83e-09	5.11e-08	**5.31e-10**	9.95e-07	7.66e-10	1.12e-06	4.19e-08	2.50e-06	3.45e-09	5.53e+01	9.31e-09
	std	1.31e+00	7.65e-10	6.45e-09	1.32e-08	6.26e-08	**5.97e-10**	1.06e-06	9.27e-10	1.42e-06	6.20e-08	2.93e-06	5.79e-09	6.35e+00	1.29e-08
	SR	0	100	100	100	100	100	100	100	99	100	96	100	0	100
	rank	13	2	5	7	9	1	10	3	11	8	12	4	14	6
f_7	mean	1.02e+00	5.92e-23	9.09e-23	1.00e-22	1.27e-22	**5.33e-23**	3.63e-22	6.20e-23	4.30e-22	4.41e-22	2.98e-22	8.41e-23	2.78e-22	5.36e-23
	std	1.86e-01	7.78e-23	9.83e-23	1.20e-22	1.43e-22	**6.30e-23**	3.81e-22	8.18e-23	4.78e-22	4.89e-22	3.82e-22	1.02e-22	3.49e-22	6.79e-23
	SR	0	100	100	100	100	100	100	100	100	100	100	100	100	100
	rank	14	3	6	7	8	1	11	4	12	13	10	5	9	2
f_8	mean	1.10e-01	7.69e-02	7.72e-02	7.89e-02	7.79e-02	7.83e-02	8.03e-02	8.04e-02	7.95e-02	7.90e-02	7.81e-02	7.99e-02	6.49e-02	**6.16e-02**
	std	4.02e-02	2.15e-02	2.30e-02	2.20e-02	1.91e-02	2.24e-02	2.09e-02	2.41e-02	2.10e-02	2.17e-02	2.31e-02	2.12e-02	**1.86e-02**	1.91e-02
	SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	rank	14	3	4	8	5	7	12	13	10	9	6	11	2	1
f_9	mean	2.96e-06	2.98e-79	2.93e-77	8.43e-83	2.42e-82	5.70e-80	1.94e-81	3.34e-78	1.48e-78	4.41e-79	2.11e-79	2.03e-78	8.29e-76	**1.99e-83**
	std	5.22e-06	2.50e-78	2.00e-76	6.67e-82	1.11e-81	5.16e-79	1.87e-80	3.33e-77	1.47e-77	4.30e-78	1.79e-78	2.03e-77	8.27e-75	**1.89e-82**
	SR	92	100	100	100	100	100	100	100	100	100	100	100	100	100
	rank	14	7	12	2	3	5	4	11	9	8	6	10	13	1
best/worst ranking		0/7	1/0	0/1	0/0	0/0	2/0	0/0	1/0	0/0	0/0	0/0	0/0	0/1	5/0
rank sum		124	31	73	52	55	34	91	50	99	93	75	55	93	20
Overall rank		14	2	8	5	6	3	10	4	13	11	9	6	11	1

be noted that the most convergence curves are plotted in a semi-logarithmic coordinate system except f_5, f_6, f_9, and f_{20}.

For unimodal test functions, CDDS is significantly superior to the other algorithms. As can be seen in Figure 2, the quality of the initial solution obtained by CDDS is worse than that of other algorithms. This phenomenon is caused due to the difference between single-solution based search algorithm and population-based search algorithm. CDDS searches the optimal solution with the fastest convergence speed in the early stage of the search process on f_1-f_2, f_4-f_7 and f_9. Then CDDS acheives VTR except f_2 and f_8. For f_3 and f_5, GSA shows a faster convergence speed initially and converges slowly as the search proceeds. For f_8, KH and CDDS have the fastest convergence speed at the beginning of search process, and then KH finds the best solution at the end of the process. In particular, we can find that the convergence curves of CDDS on f_1, f_3, f_4 and f_7 disappeared after some function evaluations. The reason is that the error values obtained by CDDS are close to zero, then the values go to negative infinity after taking logarithm. This illustrates that CDDS has fast convergence rate and high accuracy. To sum up, CDDS performs best in terms of accuracy and convergence speed for unimodal functions. It proves that CDDS improves the ''exploitation'' of DDS.

As shown in Figure 3, CDDS has the fastest convergence speed on f_{10} and f_{15}-f_{17}. Meanwhile, GSA shows the fastest convergence speed on f_{11}-f_{13} and f_{19}. For f_{14}, CDDS has the fastest convergence speed initially, then KH converges fastest, EDEV finds the best solution on this function. With regard to f_{17}, the optimal value obtained by CDDS acheives VTR rapidly, and therefore the convergence curve disappeared after about 1500 function evaluations. For f_{18}, the average best objective function values of DE, PSO, SCA, JAYA, EPSO and CDDS are positive numbers, meanwhile those of BA, KH, and GSA are negative numbers. Especially, the average best objective function values of EDEV are positive numbers in the early stage of the search process, and turn negative as the search proceeds. f_{18} is composed of a great number of peaks and valleys, and it has a second best minimum far from the global minimum. This may be the

TABLE 6. Comparison of experimental results by CDDS algorithm with thirteen different chaotic maps on classical multimodal benchmark functions.

Func	Criteria	DDS	CDDS1	CDDS2	CDDS3	CDDS4	CDDS5	CDDS6	CDDS7	CDDS8	CDDS9	CDDS10	CDDS11	CDDS12	CDDS13
f_{10}	mean	1.07e+00	7.88e-03	1.04e-02	6.79e-03	6.99e-03	6.31e-03	7.41e-03	8.34e-03	5.47e-03	7.66e-03	6.92e-03	6.62e-03	1.32e-02	**4.98e-03**
	std	2.89e-02	9.98e-03	8.18e-03	9.17e-03	8.85e-03	7.68e-03	8.80e-03	1.09e-02	7.13e-03	8.64e-03	8.56e-03	8.36e-03	1.38e-02	**6.32e-03**
	SR	0	43	20	51	52	52	44	44	55	41	50	49	28	55
	rank	14	10	12	5	7	3	8	11	2	9	6	4	13	1
f_{11}	mean	**4.43e-02**	7.48e-01	6.57e-01	6.26e-01	3.86e-01	4.85e-01	6.31e-01	6.82e-01	6.34e-01	8.96e-01	2.86e-01	5.55e-01	2.37e+00	1.81e-01
	std	**9.86e-03**	1.12e+00	1.08e+00	1.01e+00	8.44e-01	9.59e-01	1.04e+00	1.09e+00	1.03e+00	1.17e+00	7.32e-01	1.01e+00	5.61e-01	5.56e-01
	SR	0	68	72	71	82	79	72	70	72	62	86	75	4	89
	rank	1	12	10	7	4	5	8	11	9	13	3	6	14	2
f_{12}	mean	**3.63e+00**	2.99e+01	2.91e+01	2.63e+01	2.75e+01	2.99e+01	2.94e+01	2.94e+01	2.85e+01	2.87e+01	2.64e+01	2.95e+01	2.71e+01	1.76e+01
	std	**1.18e+00**	3.99e+00	4.27e+00	3.80e+00	3.78e+00	4.40e+00	3.93e+00	4.61e+00	3.77e+00	4.24e+00	3.81e+00	4.04e+00	5.58e+00	3.92e+00
	SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	rank	1	14	9	3	6	13	10	11	7	8	4	12	5	2
f_{13}	mean	**3.84e-02**	1.33e+01	2.12e+00	1.38e+01	1.44e+01	9.75e+00	1.85e+01	1.37e+01	6.75e+00	1.44e+01	1.57e+01	1.29e+01	1.58e+00	7.46e+00
	std	**1.63e-02**	4.10e+00	4.64e+00	4.10e+00	4.79e+00	6.60e+00	4.20e+00	4.55e+00	3.36e+00	4.56e+00	4.12e+00	4.46e+00	1.20e+00	3.46e+00
	SR	0	0	10	0	0	3	0	0	1	0	0	0	8	0
	rank	1	8	3	10	12	6	14	9	4	11	13	7	2	5
f_{14}	mean	**4.45e-02**	3.36e-01	1.10e+00	2.07e-01	2.02e-01	2.98e-01	1.90e-01	2.41e-01	4.24e-01	3.08e-01	1.83e-01	3.06e-01	3.86e+00	2.18e-01
	std	**4.10e-02**	3.69e-01	8.08e-01	2.60e-01	2.50e-01	4.20e-01	3.01e-01	2.83e-01	4.26e-01	4.08e-01	2.24e-01	3.33e-01	1.25e+00	2.96e-01
	SR	0	26	0	39	33	28	45	31	18	29	36	21	0	43
	rank	1	11	13	5	4	8	3	7	12	10	2	9	14	6
f_{15}	mean	3.31e-01	3.46e-01	3.34e-02	5.23e-01	4.53e-01	2.74e-01	1.45e+00	3.36e-01	1.39e-01	6.95e-01	9.30e-01	3.31e-01	**2.30e-07**	2.43e-01
	std	1.17e-01	3.46e-01	7.33e-02	5.14e-01	4.31e-01	2.59e-01	7.81e-01	3.36e-01	1.85e-01	5.88e-01	6.77e-01	3.55e-01	**3.83e-08**	2.22e-01
	SR	0	21	52	10	12	30	0	22	44	8	2	19	100	15
	rank	7	9	2	11	10	5	14	8	3	12	13	6	1	4
f_{16}	mean	5.76e+01	3.23e+00	3.29e+00	1.70e+00	2.25e+00	2.32e+00	9.89e-01	2.76e+00	3.72e+01	**5.59e-01**	9.40e-01	2.74e+00	1.24e+00	2.10e+00
	std	1.84e+01	5.73e+00	4.77e+00	4.42e+00	4.49e+00	3.89e+00	2.07e+00	4.75e+00	1.77e+02	**1.50e+00**	1.92e+00	6.77e+00	1.11e+00	4.21e+00
	SR	0	0	0	1	0	0	24	0	0	37	3	0	1	0
	rank	14	11	12	5	7	8	3	10	13	1	2	9	4	6
f_{17}	mean	2.27e+05	1.34e-34	1.00e-34	1.65e-34	1.79e-34	7.66e-35	4.12e-34	1.27e-34	5.26e-34	5.42e-34	1.84e-34	1.52e-34	2.49e-34	**2.61e-35**
	std	1.76e+05	6.66e-34	2.68e-34	7.11e-34	7.80e-34	2.63e-34	1.67e-33	5.13e-34	1.97e-33	1.90e-33	4.73e-34	6.97e-34	9.57e-34	**6.93e-35**
	SR	0	100	100	100	100	100	100	100	100	100	100	100	100	100
	rank	14	5	3	7	8	2	11	4	12	13	9	6	10	1
f_{18}	mean	1.49e+03	4.10e+03	4.33e+03	5.83e+03	5.61e+03	4.87e+03	5.55e+03	4.30e+03	-2.34e+301	**8.75e+02**	6.39e+03	4.55e+03	-2.08e+300	4.58e+03
	std	**3.13e+02**	7.98e+02	8.47e+02	8.42e+02	6.45e+02	7.90e+02	6.94e+02	7.96e+02	Inf	6.54e+02	8.68e+02	8.29e+02	Inf	2.17e+03
	SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	rank	2	3	5	11	10	8	9	4	14	1	12	6	13	7
f_{19}	mean	1.31e-01	3.69e-02	6.18e-02	1.36e-02	2.33e-02	2.27e-02	3.76e-02	3.58e-02	5.31e-02	5.32e-02	1.86e-02	3.35e-02	1.42e-01	**4.05e-03**
	std	7.52e-02	3.59e-02	4.90e-02	1.36e-02	2.58e-02	2.13e-02	3.19e-02	2.79e-02	7.50e-02	4.16e-02	2.07e-02	2.72e-02	1.04e-01	**7.85e-03**
	SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	rank	13	8	12	2	5	4	9	7	10	11	3	6	14	1
f_{20}	mean	4.15e-01	4.49e-01	4.48e-01	4.35e-01	4.38e-01	4.42e-01	4.45e-01	4.46e-01	4.44e-01	4.48e-01	4.35e-01	4.44e-01	4.26e-01	**4.11e-01**
	std	3.22e-02	**1.47e-02**	1.55e-02	1.90e-02	2.04e-02	1.83e-02	1.58e-02	1.84e-02	1.86e-02	1.48e-02	1.88e-02	1.71e-02	2.13e-02	3.04e-02
	SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	rank	2	14	13	4	6	7	10	11	9	12	5	8	3	1
best/worst ranking		4/3	0/2	0/0	0/0	0/0	0/0	0/2	0/0	0/1	2/0	0/0	0/0	1/3	4/0
rank sum		70	105	94	70	79	69	99	93	95	101	72	79	93	36
Overall rank		3	14	10	3	6	2	12	8	11	13	5	6	8	1

TABLE 7. Parameter settings for various algorithms.

Algorithms	Parameters
DE	The scale factor $F = 0.5$ and crossover rate $CR = 0.9$
PSO	The inertia weight ω: $0.9 - 0.4$, the learning factors $c_1 = 2$,$c_2 = 2$
BA	The loudness rate $\alpha = 0.9$, the pulse rate $\gamma = 0.9$
SCA	$a = 2$
GSA	$G_0 = 100$, $\alpha = 20$
KH	The foraging speed $V_f = 0.02$, the maximum diffusion speed $D^{max} = 0.005$, the maximum induced speed $N^{max} = 0.01$
EPSO	$g_1 = 20$ and $g_2 = 30$
EDEV	$\lambda_1 = \lambda_2 = \lambda_3 = 0.1$, $\lambda_4 = 0.7$, $ng = 20$

reason that some algorithms are trapped in local optimal. For f_{20}, GSA, KH and CDDS have the fastest convergence speed at the beginning of search process, and then they seem to trap in local optimal. According to these analyses, it can be concluded that CDDS is superior to or competitive with the other heuristic algorithms on the multimodal functions.

C. GENERAL PERFORMANCE OF CDDS ON CEC2005 BENCHMARK FUNCTIONS

In this section, the first 14 functions from the CEC2005 benchmarks [64] are used to evaluate the performance of CDDS, which are summarized in Table 3. In this test suite, the 14 functions can be divided into three categories:

TABLE 8. Comparison of experimental results by different algorithms on classical unimodal benchmark functions.

Func	Criteria	DE	PSO	BA	SCA	GSA	KH	JAYA	EPSO	EDEV	CDDS
	mean	1.03e-01	3.21e+00	1.57e+02	3.46e+00	6.29e-03	5.00e-03	3.20e+01	1.98e-04	1.11e-03	**1.75e-42**
	std	5.02e-02	1.41e+01	4.47e+01	3.75e+00	1.42e-02	5.33e-03	6.60e+00	1.67e-04	6.22e-04	**5.02e-42**
f_1	SR	0	5	0	0	40	0	0	0	0	100
	rank	6	7	10	8	5	4	9	2	3	1
	h	+	+	+	+	+	+	+	+	+	
	mean	1.49e+03	6.04e+03	2.60e+07	1.44e+06	1.10e+03	1.88e+02	2.24e+06	1.69e+02	8.09e+01	**1.04e+01**
	std	1.36e+03	1.46e+04	1.37e+07	2.09e+06	8.93e+02	1.36e+02	1.02e+06	1.16e+02	7.08e+01	**2.62e+01**
f_2	SR	0	0	0	0	0	0	0	0	0	0
	rank	6	7	10	8	5	4	9	3	2	1
	h	+	+	+	+	+	+	+	+	+	
	mean	1.08e-01	5.43e+00	1.52e+02	9.24e+00	3.40e-01	8.09e-03	4.83e+01	2.74e-04	1.22e-03	**3.00e-08**
	std	7.78e-02	1.86e+01	3.14e+01	4.37e+00	5.17e-01	1.91e-02	1.03e+01	4.43e-04	7.91e-04	**3.71e-09**
f_3	SR	0	0	0	0	68	0	0	0	0	100
	rank	5	7	10	8	6	4	9	2	3	1
	h	+	+	+	+	=	+	+	+	+	
	mean	1.34e+00	1.21e+02	2.15e+03	4.14e+01	3.58e+00	3.73e-01	3.93e+02	4.12e-03	1.44e-02	**4.43e-41**
	std	6.18e-01	1.91e+02	6.83e+02	4.87e+01	3.48e+00	4.26e-01	8.64e+01	4.41e-03	1.07e-02	**1.82e-40**
f_4	SR	0	3	0	0	0	0	0	0	0	100
	rank	5	8	10	7	6	4	9	2	3	1
	h	+	+	+	+	+	+	+	+	+	
	mean	4.66e+03	8.57e+03	4.43e+04	1.72e+04	1.63e+03	1.51e+03	4.89e+04	1.08e+03	4.85e+02	**8.81e-06**
	std	1.47e+03	5.54e+03	1.74e+04	8.48e+03	1.19e+03	5.75e+02	7.03e+03	3.99e+02	2.98e+02	**4.95e-06**
f_5	SR	0	0	0	0	0	0	0	0	0	62
	rank	6	7	10	8	5	4	9	3	2	1
	h	+	+	+	+	+	+	+	+	+	
	mean	2.17e+01	8.35e+00	5.44e+01	4.73e+01	1.01e+01	4.42e+00	5.50e+01	1.22e+01	2.49e+00	**9.31e-09**
	std	4.90e+00	4.16e+00	6.41e+00	1.12e+01	1.79e+00	1.15e+00	7.45e+00	3.74e+00	1.01e+00	**1.29e-08**
f_6	SR	0	2	0	0	0	0	0	0	0	100
	rank	7	4	10	8	5	3	9	6	2	1
	h	+	+	+	+	+	+	+	+	+	
	mean	3.65e+00	1.13e+01	1.31e+06	9.53e-01	4.41e-01	1.80e+09	5.32e+01	6.58e-02	8.28e-02	**5.36e-23**
	std	1.99e+00	9.27e+00	1.15e+07	8.70e-01	5.76e-01	4.54e+09	2.16e+01	7.31e-02	6.22e-02	**6.79e-23**
f_7	SR	0	0	0	0	5	0	0	0	0	100
	rank	6	7	9	5	4	10	8	2	3	1
	h	+	+	+	+	+	+	+	+	+	
	mean	1.13e-01	1.25e-01	1.06e+01	6.60e-01	6.11e-02	**2.11e-02**	1.62e+00	8.97e-02	3.43e-02	6.16e-02
	std	3.22e-02	3.31e-01	4.38e+00	9.08e-01	3.30e-02	**1.49e-02**	6.36e-01	3.16e-02	1.28e-02	1.91e-02
f_8	SR	0	0	0	0	0	0	0	0	0	0
	rank	6	7	10	8	3	1	9	5	2	4
	h	+	=	+	+	=	-	+	+	-	
	mean	9.39e-09	2.58e-07	8.39e-07	2.16e-03	1.09e-07	7.23e-07	2.27e-03	1.98e-15	1.47e-17	**1.99e-83**
	std	6.01e-08	6.20e-07	5.22e-07	4.89e-03	2.05e-07	1.00e-06	2.84e-03	6.20e-15	6.02e-17	**1.89e-82**
f_9	SR	100	100	100	8	100	100	0	100	100	100
	rank	4	6	8	9	5	7	10	3	2	1
	h	+	+	+	+	+	+	+	+	+	
+		9	8	9	9	7	8	9	9	8	
best/worst ranking		0/0	0/0	0/5	0/0	0/0	1/0	0/3	0/0	0/0	8/0
Overall rank		6	7	10	8	5	4	9	3	2	1

F_1-F_5 are shifted unimodal functions; F_6-F_{12} are shifted multimodal functions; F_{13}-F_{14} are the expanded multimodal functions. CDDS is compared with seven algorithms including HPSO-TVAC [16], jDE [65], LIPS [17], EPS-dPSO [18], ARA$_E$ -SOM+BCO [19], EPSO [20] and EDEV [21]. We reported the mean and the standard deviation of error values over 25 independent runs in Table 10 where FEs is set as 300000 and $D = 30$. For comparison purpose, the results of the other compared algorithms were taken from the results reported in relevant published literatures [18]–[21]. The best results among them are marked in bold.

Table 10 shows that our algorithm is better than HPSO-TVAC for 9 functions (F_2-F_8 and F_{11}-F_{12}), better than jDE for 6 functions (F_2-F_3, F_7-F_8 and F_{11}-F_{12}), better than LIPS for 8 functions (F_2-F_8 and F_{12}), better than EPS-dPSO for 5 functions (F_3, F_7-F_8 and F_{11}-F_{12}), better than ARA$_E$ -SOM+BCO for 5 functions (F_3, F_7-F_8 and F_{11}-F_{12}), better than EPSO for 4 functions (F_3, F_7-F_8 and F_{12}), and better than EDEV for 3 functions (F_3 and F_7-F_8). In fact, CDDS yields the best solution on F_3, F_7-F_8, while HPSO-TVAC, jDE, LIPS, EPS-dPSO, ARA$_E$ -SOM+BCO, EPSO, EDEV obtained the best values in 1 (F_1), 2 (F_1 and F_9), 1 (F_1),1 (F_1), 3 (F_1, F_4 and F_6), 4 (F_1-F_2, F_9 and F_{14}), 7 (F_1, F_5 and F_9-F_{13}), respectively. As for the expanded multi-modal functions, the performance of CDDS is the worst among the compared algorithms. Taken together, the overall

TABLE 9. Comparison of experimental results by different algorithms on classical multimodal benchmark functions.

Func	Criteria	DE	PSO	BA	SCA	GSA	KH	JAYA	EPSO	EDEV	CDDS
f_{10}	mean	1.10e+00	1.23e+01	2.16e+02	4.24e+00	2.65e+02	2.08e+01	2.87e+01	6.86e-02	2.90e-01	**4.98e-03**
	std	5.86e-02	2.85e+01	5.13e+01	4.16e+00	3.33e+01	1.45e-01	5.81e+00	5.38e-02	1.43e-01	**6.32e-03**
	SR	0	0	0	0	0	0	0	0	0	55
	rank	4	6	9	5	10	7	8	2	3	1
	h	+	+	+	+	+	+	+	+	+	
f_{11}	mean	5.77e-02	8.10e-01	2.07e+00	1.68e-01	**1.98e-04**	5.05e-03	2.33e+00	5.74e-02	6.07e-03	1.81e-01
	std	1.53e-02	7.45e-01	3.32e-01	1.40e-01	**1.34e-03**	4.63e-03	2.37e-01	2.13e-01	1.75e-03	5.56e-01
	SR	0	4	0	0	93	0	0	0	0	89
	rank	5	8	10	6	1	2	9	4	3	7
	h	-	+	+	-	-	-	+	-	-	
f_{12}	mean	2.17e+02	6.83e+01	8.96e+01	8.00e+01	1.92e+01	1.79e+01	2.85e+02	4.62e+01	9.21e+01	**1.76e+01**
	std	1.13e+01	3.45e+01	2.95e+01	4.47e+01	4.92e+00	1.56e+01	2.18e+01	1.35e+01	1.16e+01	**3.92e+00**
	SR	0	0	0	0	0	0	0	0	0	0
	rank	9	5	7	6	3	2	10	4	8	1
	h	+	+	+	+	+	+	+	+	+	
f_{13}	mean	6.38e-01	5.41e+00	3.85e+01	1.23e+01	2.40e-01	8.68e-01	1.97e+01	1.46e+00	**2.78e-02**	7.46e+00
	std	2.59e-01	4.40e+00	1.34e+01	9.19e+00	2.03e-01	3.74e-01	7.90e+00	1.38e+00	**7.96e-02**	3.46e+00
	SR	0	0	0	0	0	0	0	0	0	0
	rank	3	6	10	8	2	4	9	5	1	7
	h	-	-	+	+	-	-	+	-	-	
f_{14}	mean	4.10e+00	4.13e+00	2.60e+07	1.59e+06	2.89e+00	5.60e-01	1.06e+06	2.92e+00	**2.32e-03**	2.18e-01
	std	3.87e+00	4.02e+00	2.21e+07	6.44e+06	1.42e+00	1.95e+00	2.20e+06	2.29e+00	**1.05e-02**	2.96e-01
	SR	0	0	0	0	0	88	0	0	0	43
	rank	6	7	10	9	4	3	8	5	1	2
	h	+	+	+	+	+	+	+	+	-	
f_{15}	mean	5.58e+01	3.60e+01	7.51e+07	4.19e+06	2.85e+01	**4.23e-09**	4.76e+06	4.29e+00	3.71e-03	2.43e-01
	std	2.81e+02	1.07e+02	5.24e+07	8.70e+06	8.64e+00	**1.31e-08**	2.63e+06	4.43e+00	3.76e-03	2.22e-01
	SR	0	0	0	0	0	100	0	0	0	15
	rank	7	6	10	8	5	1	9	4	2	3
	h	+	+	+	+	+	-	+	+	-	
f_{16}	mean	1.66e+02	2.59e+02	5.26e+07	6.54e+01	7.23e+01	2.02e+02	3.30e+02	1.86e+01	1.97e+01	**2.10e+00**
	std	4.19e+01	1.84e+02	1.59e+08	2.61e+01	1.80e+01	6.85e+01	6.08e+01	8.16e+00	7.92e+00	**4.21e+00**
	SR	0	0	0	0	0	0	0	0	0	0
	rank	6	8	10	4	5	7	9	2	3	1
	h	+	+	+	+	+	+	+	+	+	
f_{17}	mean	2.20e+04	3.48e+07	6.31e+08	1.18e+05	8.33e+06	1.36e+06	4.26e+06	2.13e+03	1.93e+03	**2.61e-35**
	std	1.78e+04	5.55e+07	2.87e+08	1.74e+05	6.89e+06	1.06e+06	1.24e+06	2.52e+03	1.98e+03	**6.93e-35**
	SR	0	0	0	0	0	0	0	0	0	100
	rank	4	9	10	5	8	6	7	3	2	1
	h	+	+	+	+	+	+	+	+	+	
f_{18}	mean	7.74e+03	4.98e+03	9.70e+03	8.88e+03	**-2.72e+03**	-4.36e+03	8.01e+03	4.10e+03	-3.46e+18	4.58e+03
	std	4.92e+02	9.51e+02	3.68e+02	3.51e+02	**4.76e+02**	1.46e+03	4.53e+02	6.49e+02	2.32e+19	2.17e+03
	SR	0	0	0	0	0	0	0	0	0	0
	rank	6	5	9	8	1	3	7	2	10	4
	h	+	=	+	+	-	=	+	-	+	
f_{19}	mean	5.16e+00	3.72e+00	2.45e+01	4.14e+00	9.94e-02	1.64e-01	3.26e+01	4.39e-02	1.10e+00	**4.05e-03**
	std	2.91e+00	3.26e+00	4.96e+00	3.72e+00	3.67e-01	3.45e-01	4.45e+00	5.56e-02	1.26e+00	**7.85e-03**
	SR	0	1	0	0	0	0	0	0	0	0
	rank	8	6	9	7	3	4	10	2	5	1
	h	+	+	+	+	+	+	+	+	+	
f_{20}	mean	4.77e-01	4.28e-01	4.99e-01	4.20e-01	4.38e-01	3.66e-01	4.99e-01	**3.08e-01**	3.47e-01	4.11e-01
	std	1.12e-02	9.17e-02	4.06e-04	7.20e-02	2.60e-02	6.21e-02	4.15e-04	**5.41e-02**	6.53e-02	3.04e-02
	SR	0	2	0	0	0	0	0	0	0	0
	rank	8	6	10	5	7	3	9	1	2	4
	h	+	+	+	+	+	-	+	-	-	
+		9	9	11	10	8	6	11	7	6	
best/worst ranking		0/0	0/0	0/6	0/0	2/1	1/0	0/3	1/0	2/1	5/0
Overall rank		6	8	10	7	5	4	9	2	3	1

performance of EDEV is the best, followed by EPSO, ARA_E -SOM+BCO and CDDS.

Based on the deeply study of the performance of CDDS, CDDS exhibits competitive overall performance for CEC2005, but it does not constitute an overwhelming superiority. It should be noticed that CDDS performs slightly worse than EPSO and EDEV for CEC2005, contrary to the classical benchmark functions. This may be because the exploration and exploitation ability of CDDS can't be well controlled for complex test functions. It is mostly likely that CDDS just search one solution with strong randomness in every iteration and is easy to trap in local optimum when handling the complex problems.

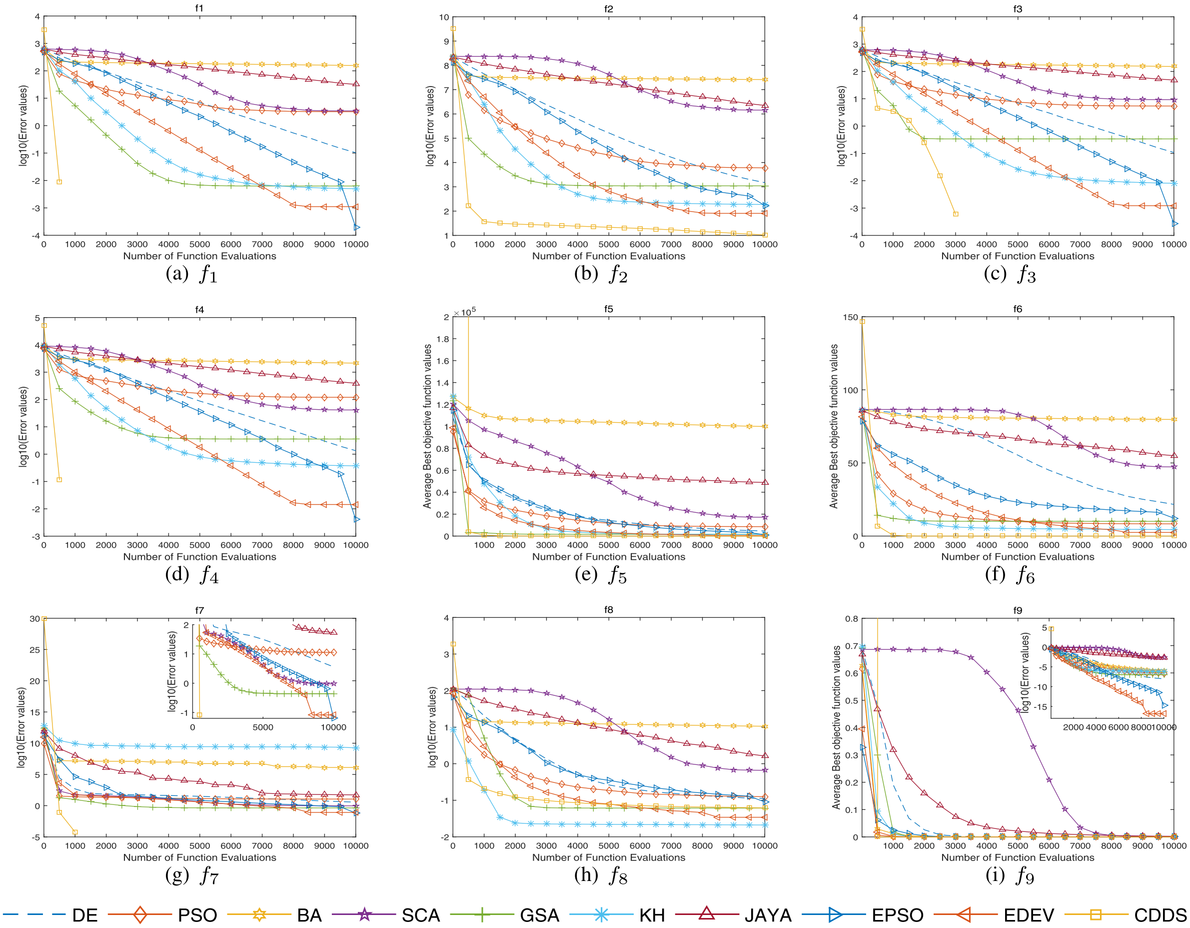


FIGURE 2. The convergence curves for unimodal benchmark functions.

D. GENERAL PERFORMANCE OF CDDS ON CEC2014 BENCHMARK FUNCTIONS

In this section, 30 rotated unimodal, shifted and rotated multimodal, hybrid, and composition functions from the CEC2014 benchmarks [66] are utilized to evaluate the effectiveness of CDDS, which are summarized in Table 4. In the test suite, the 30 functions can be divided into four categories: $F1_{CEC2014}$-$F3_{CEC2014}$ are unimodal functions; $F4_{CEC2014}$-$F16_{CEC2014}$ are simple multimodal functions; $F17_{CEC2014}$-$F22_{CEC2014}$ are hybrid functions; $F23_{CEC2014}$-$F30_{CEC2014}$ are composition functions.

In this experiment, CDDS was compared with SOS [67], GSA [8], CS [9], CMA-ES [68], HAPS-PS [69], and L-SHADE [70] on the CEC2014 test suit with $D = 30$. The maximal number of function evaluations (maxFEs) was set to $10000D$. To reduce the randomness, the experimental results were calculated from 51 independent runs. The mean and standard deviation of function error value $f(X_{best}) - f(X_*)$ were used to evaluate the optimization performance, where X_{best} is the best solution found by the algorithm in a run, and X_* is the theoretical global optimum of each function. For convenience, the results of the other compared algorithms were cited from the published literatures [69]. The experimental results are shown in Table 11, and the overall ranking values were recorded in the last row.

From the overall rank in Table 11, we can find CDDS ranks the top three among 30 CEC2014 functions while HAPS-PS and L-SHADE rank second and first, respectively. Again 6 compared algorithms, CDDS is better or equal on 24, 16, 21, 27, 9 and 4 functions than SOS, GSA, CS, CMA-ES, HAPS-PS, and L-SHADE, respectively. The detailed comparison results shows that CDDS was superior to SOS, GSA, CS, and CMA-ES on most test functions. CDDS is the best performer only for $F3_{CEC2014}$. CDDS shows the best performance for 6 functions($F1_{CEC2014}$, $F3_{CEC2014}$-$F4_{CEC2014}$, $F17_{CEC2014}$, $F21_{CEC2014}$ and $F30_{CEC2014}$), besides L-SHADE. With regard to HAPS-PS, it is a quite recently proposed algorithm and provides the best results on 6 functions($F12_{CEC2014}$, $F23_{CEC2014}$-$F26_{CEC2014}$ and $F28_{CEC2014}$). L-SHADE ranks first for 22 CEC2014 functions ($F1_{CEC2014}$-$F4_{CEC2014}$, $F6_{CEC2014}$-$F10_{CEC2014}$, $F13_{CEC2014}$, $F15_{CEC2014}$-$F22_{CEC2014}$, $F25_{CEC2014}$-$F27_{CEC2014}$ and $F30_{CEC2014}$) because it is the winner of the CEC2014 competition.

According to the above intensive study, CDDS has a good comprehensive performance for CEC2014. In order

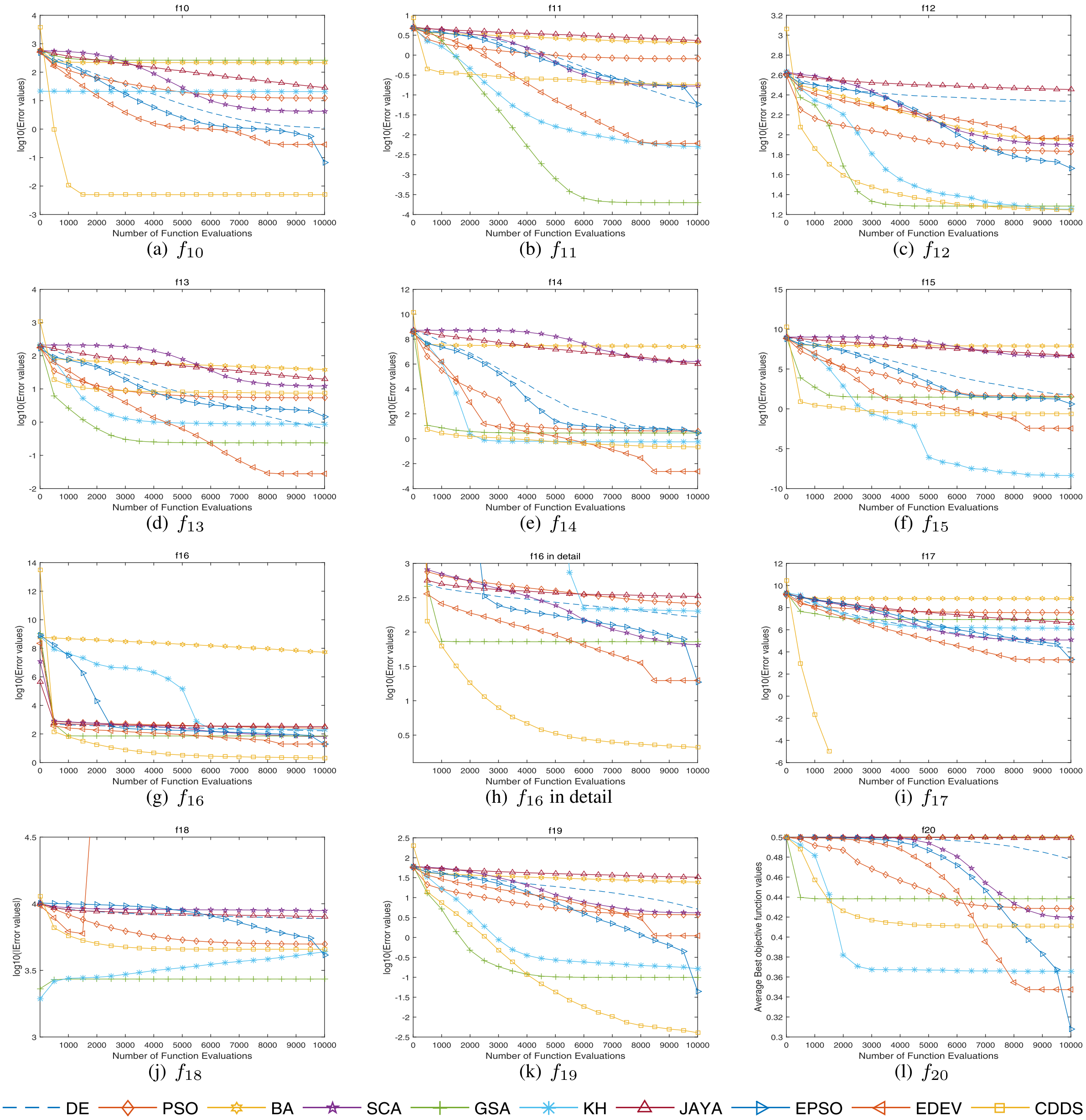


FIGURE 3. The convergence curves for multimodal benchmark functions.

to analyze the deep reasons, we provide a detailed analysis. Firstly, we can find that CDDS confirms its superiority on unimodal functions due to its strong exploitation ability. As a single-solution-based optimization algorithm, CDDS searches the design space with a single point so it makes full use of global optimal solution information, which enhance the exploitation capability. For simple multimodal functions, CDDS obtains the top three mean values for six functions ($F4_{CEC2014}$-$F6_{CEC2014}$, $F12_{CEC2014}$, $F14_{CEC2014}$, and $F16_{CEC2014}$). The superiority may be because CDDS has good exploration capability due to chaotic initialization and chaotic search. Then, with regard to hybrid functions, CDDS gets the top three mean values for three functions ($F17_{CEC2014}$, $F20_{CEC2014}$, and $F21_{CEC2014}$). It means that CDDS can jump from local optimal solution when dealing with complex problems. This may be because of chaotic search which can increase the probability of CDDS escaping local optima. Finally, CDDS obtains the top three mean values for five functions ($F23_{CEC2014}$-$F24_{CEC2014}$, $F26_{CEC2014}$, $F29_{CEC2014}$, and $F30_{CEC2014}$) for composition functions. In summary, CDDS can effectively balance exploration and exploitation, thereby it is a competitive algorithm for CEC2014.

VI. THE EFFECT OF THREE STRATEGIES

In this section we will discuss the effect of chaotic initialization, Gaussian mutation operator and chaotic search through two aspects.

TABLE 10. Comparison of experimental results of the CDDS algorithm for the 30 dimensional CEC2005 benchmark functions.

Func	Criteria	HPSO-TVAC	jDE	LIPS	EPS-dPSO	ARA_E -SOM+BCO	EPSO	EDEV	CDDS
F_1	mean	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	1.16e-06
	std	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	3.74e-07
F_2	mean	3.79e-06	3.45e-06	8.35e+01	3.59e-13	3.43e-23	**0.00e+00**	3.39e-28	3.44e-06
	std	2.82e-06	2.76e-06	3.27e+02	1.39e-12	1.10e-22	**1.88e-12**	1.22e-28	8.21e-07
F_3	mean	7.72e+05	2.44e+05	1.00e+07	1.31e+04	4.12e+04	3.62e+05	2.16e+04	**7.45e+03**
	std	2.96e+05	3.22e+05	1.35e+07	0.00e+00	2.49e+04	1.35e+05	1.58e+04	**5.41e+03**
F_4	mean	2.48e+04	4.78e-02	3.32e+04	2.17e+01	**2.61e-09**	2.38e+02	2.81e-08	2.07e+04
	std	5.71e+03	2.12e-01	8.20e+03	5.50e-01	**6.45e-09**	1.51e+02	3.24e-08	5.90e+03
F_5	mean	9.20e+03	5.56e+02	9.35e+03	5.42e+03	3.28e+03	4.63e+03	**6.14e-02**	8.07e+03
	std	1.81e+03	5.62e+02	1.75e+03	1.89e+02	7.02e+02	8.08e+02	**2.14e-02**	2.31e+03
F_6	mean	5.04e+01	2.65e+01	1.06e+02	2.19e+01	**1.60e-01**	3.36	4.97e-01	3.41e+01
	std	5.05e+01	2.32e+01	2.06e+02	1.06e+00	**7.97e-01**	4.15	5.65e-01	2.49e+01
F_7	mean	0.01	1.14e-02	8.54e+01	1.17e-02	1.73e-02	0.02	1.01e-02	**9.87e-03**
	std	0.01	7.28e-03	4.54e+01	1.05e-02	2.01e-02	0.01	6.28e-03	**3.08e-06**
F_8	mean	20.71	2.09e+01	20.92	2.01e+01	2.00e+01	20.91	2.01e+01	**2.00e+01**
	std	0.15	7.28e-03	0.12	1.30e-01	3.20e-02	0.08	6.28e-03	**4.13e-04**
F_9	mean	10.71	**0.00e+00**	4.95e+01	3.23e+01	2.84e-16	**0.00e+00**	**0.00e+00**	1.37e+02
	std	4.96	**0.00e+00**	1.24e+01	1.56e+00	9.84e-16	**0.00e+00**	**0.00e+00**	2.17e+01
F_{10}	mean	2.76e+02	5.46e+01	6.72e+01	2.90e+02	1.19e+02	54.56	**2.51e+01**	4.17e+02
	std	4.55e+01	8.85e+00	1.92e+01	1.16e+01	4.18e+01	16.98	**6.54e+00**	5.59e+01
F_{11}	mean	3.06e+01	2.88e+01	2.11e+01	2.87e+01	2.82e+01	22.23	**1.25e+01**	2.76e+01
	std	2.58e+00	2.61e+00	2.68e+00	5.12e-01	4.23e+00	2.58	**5.11e+00**	3.37e+00
F_{12}	mean	4.47e+03	8.23e+03	8.01e+04	2.37e+05	1.66e+03	4.10e+03	**7.17e+02**	1.39e+03
	std	4.66e+03	8.54e+03	7.02e+04	4.51e+04	2.54e+03	4.19e+03	**3.21e+02**	1.99e+03
F_{13}	mean	3.85e+00	1.67e+00	4.08e+00	2.26	1.26e+00	1.80	**1.09e+00**	6.36e+00
	std	1.11e+00	1.56e-01	1.20e+00	7.33e-01	2.46e-01	0.43	**8.21e-01**	6.69e+00
F_{14}	mean	1.23e+01	1.30e+01	12.35	1.26e+01	1.25e+01	**11.86**	1.21e+01	1.31e+01
	std	0.37	2.23e-01	0.50	3.61e-01	4.37e-01	**0.77**	5.21e-01	2.78e-01

A. THE ADVANTAGES OF THREE STRATEGIES (Chaotic INITIALIZATION, GAUSSIAN MUTATION OPERATOR AND CHAOTIC SEARCH)

First of all, we investigated the advantages of chaotic initialization, Gaussian mutation operator and chaotic search. We removed one strategy and remained the other two strategies to obtain three modified variants of CDDS, which were compared with the full version on the previous 20 well-known benchmark functions. The experimental parameter settings are the same as before. The average (mean), run times (Time), standard deviation (std), and success rate (SR) of the optimal solutions are shown in Table 12.

In the first case, chaotic initialization was removed and the initial population was randomly generated, while the new Gaussian mutation operator and the chaotic search were used to balance the exploration and exploitation abilities. The results of this case are shown in Table 12 named CI. According to the results, the mean, Std and SR of the proposed algorithm without chaotic initialization decrease on the large majority of the test functions, meanwhile the CPU time increases in most cases. It means that chaotic initialization can improve the quality of initial population.

In the second case, the new Gaussian mutation operator was removed and replaced by original mutation operator while the chaotic map was utilized to generate initial population and the chaotic search was used in the updating stage. The results of this case marked MO. In general, the performance of the proposed algorithm in this case is worse than the performance of CDDS. The mean, Std and SR of the proposed algorithm in this case are the worst on unimodal test functions except f_3, which is contrary to some multimodal test functions. Meanwhile, the CPU time increases compared to CDDS. This means the new Gaussian mutation operator can enhance the exploitation of the proposed algorithm.

In the last case, the chaotic search was removed, while chaotic initialization and new Gaussian mutation operator were employed. The tag of this case is CS. From the results in Table 12, the mean, Std, SR and the CPU time of the proposed algorithm in this case decrease. It means that chaotic search can effectively jump from local optimal and enhance the exploration ability, but the chaotic search may be time consuming.

B. INFLUENCE OF THE STARTING ITERATION OF THE CHAOTIC SEARCH

According to the analyses in Subsection VI-A, we find that the starting iteration of the chaotic search is important for CDDS. The results of f_3, f_5, f_{15} and f_{19} in Table 12 indicate that the chaotic search plays a major role when the search is trapped in local minima. In order to find the best starting iteration of the chaotic search, we have tested 20 benchmark functions in Tables 1 and 2 with different starting iteration of the chaotic search for CDDS. The results were shown in Tables 13 and 14.

Table 13 includes the average of 20 classical benchmark function values with different starting iteration for CDDS. Here "0.0*FEs" means the chaotic search is used at the beginning of the search process, and "1.0*FEs" indicates

TABLE 11. Comparison of experimental results of the CDDS algorithm for the 30 dimensional CEC2014 benchmark functions.

Func	Criteria	SOS	GSA	CS	CMA-ES	HAPS-PS	L-SHADE	CDDS
$F1_{CEC2014}$	mean	6.94e+07	1.56e+07	2.28e+05	9.42e+04	5.24e+06	0.00e+00	9.00e+03
	std	2.91e+07	3.57e+06	7.85e+02	7.88e+04	1.95e+06	0.00e+00	5.36e+03
	rank	7	6	4	3	5	1	2
$F2_{CEC2014}$	mean	3.61e+09	8.88e+03	1.38e+02	2.55e+10	5.80e+03	0.00e+00	2.12e+04
	std	7.75e+08	1.62e+03	4.71e+01	3.85e+09	1.17e+03	0.00e+00	1.29e+04
	rank	6	4	2	7	3	1	5
$F3_{CEC2014}$	mean	2.37e+04	7.31e+04	1.04e+04	1.44e+04	7.86e+00	0.00e+00	0.00e+00
	std	1.35e+04	4.12e+03	5.53e-01	5.66e+03	3.64e+00	0.00e+00	0.00e+00
	rank	6	7	4	5	3	1	1
$F4_{CEC2014}$	mean	4.14e+02	3.68e+02	7.07e+01	2.52e+03	4.38e+01	0.00e+00	6.26e-01
	std	7.32e+01	2.98e+01	2.13e+01	5.35e+02	4.49e+01	0.00e+00	1.46e+00
	rank	6	5	4	7	3	1	2
$F6_{CEC2014}$	mean	2.48e+01	2.87e+01	3.23e+01	4.08e+01	3.72e+00	1.40e-07	2.45e+01
	std	1.93e+00	2.49e+00	3.27e+00	2.12e+00	3.18e+00	9.90e-07	3.75e+00
	rank	4	5	6	7	2	1	3
$F7_{CEC2014}$	mean	3.38e+01	1.99e-04	1.79e+00	2.31e+02	6.93e-04	0.00e+00	1.31e-02
	std	8.02e+00	2.01e-05	2.19e+00	2.82e+01	1.66e-03	0.00e+00	1.81e-02
	rank	6	2	5	7	3	1	4
$F8_{CEC2014}$	mean	8.09e+01	1.40e+02	7.11e+01	2.83e+02	4.21e+00	0.00e+00	9.97e+01
	std	7.89e+00	8.41e+00	1.12e+01	2.20e+01	2.28e+00	0.00e+00	2.22e+01
	rank	4	6	3	7	1	1	5
$F9_{CEC2014}$	mean	1.72e+02	1.64e+02	1.78e+02	3.28e+02	3.07e+01	6.80e+00	1.78e+02
	std	1.89e+01	1.20e+01	3.47e+01	7.65e+01	7.18e+00	1.50e+00	2.61e+01
	rank	3	6	4	7	2	1	4
$F10_{CEC2014}$	mean	2.39e+03	3.35e+03	2.02e+03	2.61e+02	1.51e+02	1.60e-02	1.70e+03
	std	1.19e+02	3.18e+02	1.93e+02	1.05e+02	1.07e+02	1.60e-02	4.63e+02
	rank	6	7	5	3	2	1	4
$F11_{CEC2014}$	mean	4.48e+03	4.05e+03	4.49e+03	1.68e+02	2.14e+03	1.20e+03	3.43e+03
	std	4.08e+02	4.24e+02	3.38e+02	1.98e+02	2.71e+02	1.80e+02	6.05e+02
	rank	6	5	7	1	3	2	4
$F12_{CEC2014}$	mean	7.37e-01	9.10e-02	8.11e-01	3.03e-01	2.89e-02	1.60e-01	1.57e-01
	std	2.85e-01	1.22e-03	2.78e-01	2.17e+00	7.25e-02	2.30e-02	5.00e-02
	rank	6	2	7	5	1	4	3
$F13_{CEC2014}$	mean	6.85e-01	4.02e-01	4.17e-01	5.50e+00	3.61e-01	1.20e-01	6.91e-01
	std	2.07e+00	3.55e-02	4.46e-02	3.07e-01	5.44e-02	1.70e-02	1.24e-01
	rank	5	3	4	7	2	1	6
$F14_{CEC2014}$	mean	8.38e+00	2.30e-01	5.17e-01	7.53e+01	4.27e-01	2.40e-01	2.42e-01
	std	4.28e+00	2.19e-02	2.65e-02	8.07e+00	2.21e-01	3.00e-02	3.83e-02
	rank	6	1	5	7	4	2	3
$F15_{CEC2014}$	mean	2.56e+02	1.25e+01	1.31e+01	1.02e+04	5.41e+00	2.10e+00	1.12e+02
	std	2.10e+02	1.98e+00	1.86e+00	3.24e+04	1.56e+00	2.50e-01	7.45e+01
	rank	6	3	4	7	2	1	5
$F16_{CEC2014}$	mean	1.20e+01	1.48e+01	1.23e+01	1.37e+01	9.84e+00	8.50e+00	1.19e+01
	std	3.69e-01	2.44e-01	1.59e-01	5.31e-01	3.62e-01	4.60e-01	5.99e-01
	rank	4	7	5	6	2	1	3
$F17_{CEC2014}$	mean	5.58e+06	7.28e+05	1.48e+06	5.48e+03	7.10e+05	1.90e+02	1.41e+03
	std	3.68e+06	1.24e+05	1.21e+06	3.62e+03	6.03e+05	7.50e+01	3.99e+02
	rank	7	5	6	3	4	1	2
$F18_{CEC2014}$	mean	4.86e+05	3.86e+02	1.42e+03	1.51e+09	2.80e+02	5.90e+00	2.45e+03
	std	2.24e+05	1.14e+02	2.42e+01	3.92e+08	3.61e+01	2.90e+00	2.78e+03
	rank	6	3	4	7	2	1	5
$F19_{CEC2014}$	mean	4.06e+01	1.56e+02	1.18e+01	2.98e+02	5.67e+00	3.70e+00	3.46e+01
	std	2.23e+01	2.48e+01	5.31e-01	4.25e+01	1.42e+00	6.801e-01	1.74e+01
	rank	6	4	3	7	2	1	5
$F20_{CEC2014}$	mean	1.59e+04	8.24e+04	3.93e+04	4.61e+03	5.88e+01	3.10e+00	1.11e+02
	std	1.01e+04	1.34e+04	2.20e+04	3.88e+03	3.48e+01	1.50e+00	2.97e+01
	rank	5	7	6	4	2	1	3
$F21_{CEC2014}$	mean	7.85e+05	1.78e+05	3.54e+05	6.86e+03	9.80e+04	8.70e+01	7.99e+02
	std	6.05e+05	3.19e+04	3.48e+05	2.75e+03	6.62e+04	9.00e+01	3.34e+02
	rank	7	5	6	3	4	1	2
$F22_{CEC2014}$	mean	5.45e+02	9.51e+02	3.11e+02	1.61e+03	2.03e+02	2.80e+01	8.08e+02
	std	1.82e+02	1.82e+02	9.15e+01	2.92e+02	1.26e+02	1.80e+01	3.41e+02
	rank	4	6	3	7	2	1	5
$F23_{CEC2014}$	mean	3.42e+02	2.00e+02	3.43e+02	5.79e+02	2.00e+02	3.20e+02	3.14e+02
	std	2.18e+01	2.34e-05	8.19e-02	4.94e+01	8.42e-04	0.00e+00	4.02e-13
	rank	5	1	6	7	1	4	3
$F24_{CEC2014}$	mean	2.36e+02	2.00e+02	2.21e+02	2.12e+02	2.00e+02	2.20e+02	2.03e+02
	std	9.89e+00	7.05e-02	1.45e+ 00	7.49e+00	3.48e-09	1.10e+00	6.99e+00

TABLE 11. *(Continued.)* **Comparison of experimental results of the CDDS algorithm for the 30 dimensional CEC2014 benchmark functions.**

Func	Criteria	SOS	GSA	CS	CMA-ES	HAPS-PS	L-SHADE	CDDS
	rank	7	1	6	4	1	5	3
$F25_{CEC2014}$	mean	2.14e+02	2.00e+02	2.09e+02	2.12e+02	2.00e+02	2.00e+02	2.16e+02
	std	3.49e+00	4.92e-07	6.11e-01	2.96e+00	2.82e-03	2.00e+02	8.47e+00
	rank	6	1	4	5	1	1	7
$F26_{CEC2014}$	mean	1.01e+02	1.69e+02	1.01e+02	1.25e+02	1.00e+02	1.00e+02	1.01e+02
	std	2.90e-02	2.90e+01	4.18e-02	5.51e+01	3.15e+01	1.60e-02	1.05e-01
	rank	3	7	3	6	1	1	3
$F27_{CEC2014}$	mean	4.96e+02	7.69e+02	4.18e+02	1.07e+03	5.60e+02	3.00e+02	5.68e+02
	std	1.51e+02	5.68e+02	5.68e+00	2.30e+02	1.40e+02	0.00e+00	2.87e+02
	rank	3	6	2	7	4	1	5
$F28_{CEC2014}$	mean	1.32e+03	7.65e+02	3.49e+03	2.79e+03	1.15e+02	8.40e+02	2.79e+03
	std	1.05e+02	3.08e+02	5.48e+02	5.91e+02	2.33e+00	1.40e+01	5.05e+02
	rank	4	2	7	5	1	3	5
$F29_{CEC2014}$	mean	2.15e+04	2.00e+02	5.44e+05	3.52e+04	1.39e+03	7.20e+02	1.08e+03
	std	1.18e+03	4.57e-02	2.61e+06	5.33e+03	4.82e+03	5.10e+00	4.93e+02
	rank	5	1	7	6	4	2	3
$F30_{CEC2014}$	mean	3.73e+04	2.31e+04	2.49e+04	6.47e+05	3.26e+03	1.20e+03	1.42e+03
	std	2.12e+04	2.43e+04	2.26e+04	1.31e+05	9.64e+02	6.20e+02	4.12e+02
	rank	6	4	5	7	3	1	2
Overall rank		6	4	5	7	2	1	3

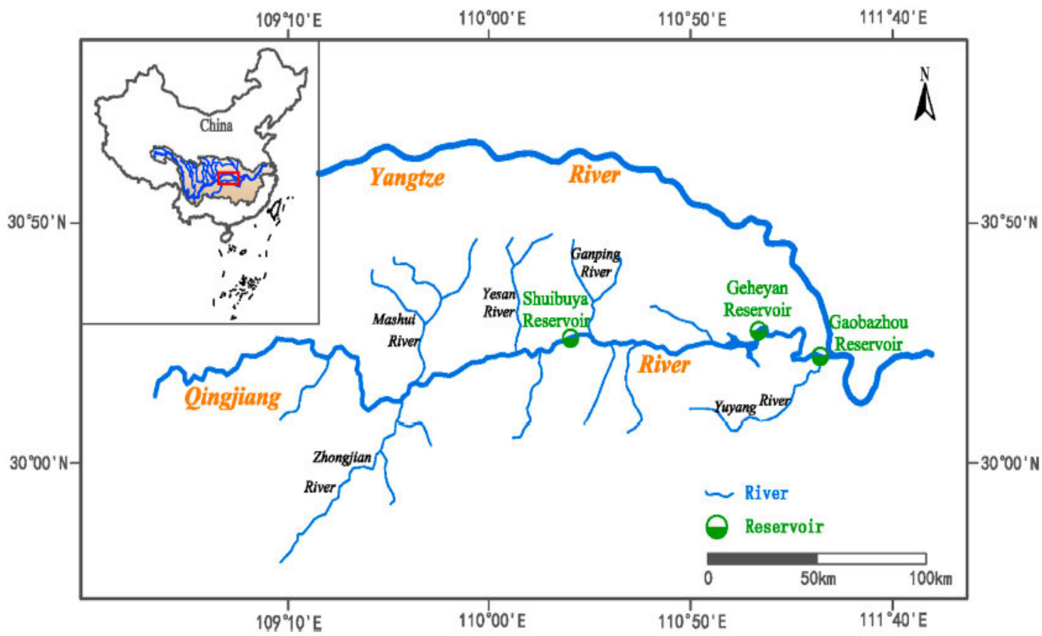


FIGURE 4. The location of cascade reservoirs in the Qingjiang River.

that the chaotic search is removed. To have a better comparison, we ranked the results in Table 14. From Table 14, CDDS obtains the best rank on 8 functions (f_1, f_3-f_4, f_7, f_9, f_{11}, f_{15} and f_{17}) when the starting iteration is "0.0*FEs"; CDDS obtains the best rank on 5 functions (f_2, f_5, f_{13}, f_{16} and f_{19}) when the starting iteration is "0.1*FEs"; CDDS obtains the best rank on 2 functions (f_{10} and f_{14}) when the starting iteration is "0.4*FEs"; CDDS obtains the best rank on 2 functions (f_8 and f_{20}) when the starting iteration is "1.0*FEs"; CDDS obtains the best rank on 1 functions (f_{18}) when the starting iteration is "0.2*FEs"; CDDS obtains the best rank on 1 functions (f_6) when the starting iteration is "0.3*FEs"; CDDS obtains the best rank on 1 functions (f_{12}) when the starting iteration is "0.7*FEs". It is noteworthy that the starting iteration of "0.1*FEs" has the best overall rank, followed by the starting iteration of "0.0*FEs". To have an integrative consideration, we select "0.0*FEs" as the best starting iteration of the chaotic search.

Another point worth noticing is that the best rank is obtained mostly when the chaotic search is used in the early stage of the search, while CDDS only performs best on 3 functions if chaotic search is utilized in the later stage of

TABLE 12. The results of testing the influence of removing one component.

	Mean				Time			
	CI	MO	CS	CDDS	CI	MO	CS	CDDS
f_1	3.40e-42	4.46e-08	7.86e-39	1.75e-42	6.9473	9.5793	4.3070	7.3077
f_2	1.10e+01	1.01e+02	3.50e+01	1.04e+01	8.9824	10.283	2.6798	9.3348
f_3	2.96e-08	4.56e-08	3.81e+00	3.00e-08	7.5472	8.6098	2.3864	7.4976
f_4	1.23e-40	1.42e-06	9.39e-38	4.43e-41	7.1617	8.5572	4.8421	7.0498
f_5	7.82e-04	5.30e+03	4.71e+00	8.81e-06	19.3009	24.688	3.1509	3.0172
f_6	1.07e-08	2.71e+00	1.90e-09	9.31e-09	6.7706	15.616	7.4655	9.6627
f_7	8.07e-23	1.33e-01	1.66e-21	5.36e-23	6.7581	11.341	4.6757	8.9220
f_8	7.63e-02	1.19e-01	3.79e-02	6.16e-02	7.5285	7.6775	1.2266	7.7625
f_9	9.20e-83	6.52e-14	1.48e-79	1.99e-83	8.2968	8.7654	6.4990	7.9554
f_{10}	6.85e-03	1.05e+00	7.46e-03	4.98e-03	7.0691	8.6799	6.0346	6.7914
f_{11}	5.44e-01	3.04e-05	2.36e-01	1.81e-01	7.3689	8.8783	5.9401	7.5403
f_{12}	2.74e+01	6.17e-01	1.69e+01	1.76e+01	8.8016	9.3196	2.2761	8.3749
f_{13}	1.11e+01	2.70e-08	7.59e+00	7.46e+00	12.468	11.725	1.8596	11.743
f_{14}	3.27e-01	9.72e-03	8.80e-01	2.18e-01	14.223	14.025	2.4103	13.265
f_{15}	2.74e-01	2.24e-02	2.11e+00	2.43e-01	12.988	13.071	2.2634	11.677
f_{16}	2.42e+00	5.60e+01	2.19e+00	2.10e+00	7.2362	6.5633	2.2205	6.6821
f_{17}	1.06e-34	4.94e+01	1.48e-33	2.61e-35	9.0128	10.504	4.3089	8.8029
f_{18}	5.48e+03	6.79e+02	4.56e+03	4.58e+03	8.6351	10.324	1.3179	9.4709
f_{19}	1.42e-02	3.58e-02	2.33e-01	4.05e-03	7.8711	7.6013	2.0304	7.1958
f_{20}	4.41e-01	4.14e-01	3.90e-01	4.11e-01	7.8774	7.8933	1.1378	7.7125
	Std				**SR**			
	CI	MO	CS	CDDS	CI	MO	CS	CDDS
f_1	9.71e-42	8.65e-09	2.46e-38	5.02e-42	100	100	100	100
f_2	2.28e+01	2.79e+02	4.35e+01	2.62e+01	0	0	0	0
f_3	4.34e-09	8.11e-09	7.31e-01	3.71e-09	100	100	0	100
f_4	4.45e-40	8.09e-07	2.86e-37	1.82e-40	100	100	100	100
f_5	7.60e-03	1.47e+03	1.37e+01	4.95e-06	44	0	0	62
f_6	1.33e-08	6.31e-01	3.63e-09	1.29e-08	100	0	100	100
f_7	9.16e-23	8.56e-02	1.86e-21	6.79e-23	100	0	100	100
f_8	2.01e-02	3.89e-02	1.33e-02	1.91e-02	0	0	0	0
f_9	7.96e-82	2.22e-13	1.47e-78	1.89e-82	100	100	100	100
f_{10}	9.97e-03	2.10e-02	8.50e-03	6.32e-03	53	0	45	55
f_{11}	9.61e-01	2.72e-06	6.20e-01	5.56e-01	74	0	86	89
f_{12}	4.00e+00	7.64e-01	4.06e+00	3.92e+00	0	43	0	0
f_{13}	4.12e+00	5.88e-09	3.73e+00	3.46e+00	0	100	0	0
f_{14}	3.47e-01	3.64e-02	4.49e-01	2.96e-01	26	84	0	43
f_{15}	2.69e-01	3.05e-02	5.71e-01	2.22e-01	23	18	0	15
f_{16}	3.14e+00	1.85e+01	2.50e+00	4.21e+00	0	0	0	0
f_{17}	4.92e-34	3.08e+01	3.53e-33	6.93e-35	100	0	100	100
f_{18}	6.56e+02	5.92e+02	2.26e+03	2.17e+03	0	11	0	0
f_{19}	1.57e-02	1.52e-02	1.38e-01	7.85e-03	0	0	0	0
f_{20}	1.86e-02	3.20e-02	3.35e-02	3.04e-01	0	0	0	0

the search process. These analyses prove the effectiveness of the chaotic search.

VII. THE APPLICATION OF OPTIMAL OPERATION OF CASCADE RESERVOIRS

In this section, the hydropower stations on the Qingjiang River in China are used to evaluate the practical feasibility and the capability of CDDS for solving the cascade reservoirs operation optimization (CROO). CROO needs to consider the hydraulic connection between upstream and downstream reservoirs. It has a lot of equality and inequality constraints which is a challenging task when the objective function directly affects the candidate solution of the updating stage. Thus, how to obtain the global optimal solution of CROO is representative to demonstrate the feasibility and effectiveness of CDDS for solving real-world optimal problems [71], [72].

A. QINGJIANG

The Qingjiang River [73]–[75], which is 423 km long, is the largest tributary of the middle Yangtze River. It flows through Enshi, Badong, Changyang and Yidu located in Hubei province, China. The Qingjiang River Basin has a moderate climate, the basin area of 17,000 km^2, an average annual rainfall of about 1400 mm, and an average flow of 440 m$^3/s$. We select three hydropower stations as a case study, including Gaobazhou hydropower station, Geheyan hydropower station, and Shuibuya hydropower station, which are in the midstream and downstream of the Qingjiang River Basin. Three cascade reservoirs: Gaobazhou (GBZ) reservoir, Geheyan (GHY)reservoir, and Shuibuya (SBY) reservoir are shown in Figures 4 and 5.

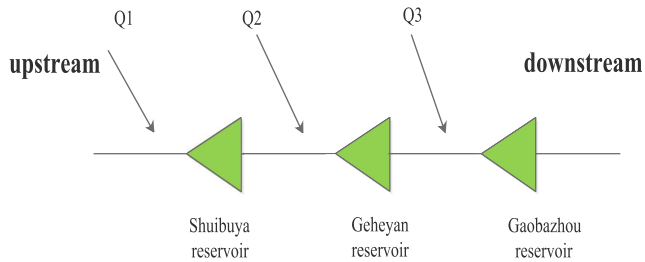


FIGURE 5. Schematic diagram of cascade reservoirs in the Qingjiang River.

In Figure 5, Q1 is the streamflow into Shuibuya reservoir, Q2 is the lateral inter-zone inflow between Shuibuya and Geheyan, and Q3 is the lateral inter-zone inflow between Geheyan and Gaobazhou. Notice the GBZ is a daily regulating and small storage reservoir, which plays an important role in the reverse regulation of upstream Geheyan. The optimal operation model used in this section is only applied to the SBY and GHY. The main features of the two reservoirs are shown in Table 15.

B. MATHEMATICAL MODELING OF THE CROO PROBLEM

Power generation is an important method to obtain the economic benefit from a cascade reservoirs system. The benefit from power generation depends on the utilization ratio of water. Thus, the optimal operation of a cascade reservoirs system is a vital and complex task for the society needs and economic development.

CROO aims at maximizing the annual power generation while satisfying all physical and operational constraints. In the operation optimization, the whole operation time is divided into 12 periods, which ranges from May to April of the next year. The length of the operation period is set to one month. Because the water level can be converted into the inflow, the outflow and reservoir capacity according to some curve equations, we choose the water level as the decision variable.

The objective function can be expressed as follows:

$$\begin{aligned} Maximize\ F &= \sum_{t=1}^{T}\sum_{m=1}^{M} N_{m,t}\,\Delta t \\ &= \sum_{t=1}^{T}\sum_{m=1}^{M} k_m q_{m,t} h_{m,t}\,\Delta t \end{aligned}$$

where T is the number of operation periods, M is the number of the hydro plants, $N_{m,t}$ is the power output of m-th plant

TABLE 13. The average values of the classical benchmark function with different starting iteration for CDDS.

	0.0*FEs	0.1*FEs	0.2*FEs	0.3*FEs	0.4*FEs	0.5*FEs	0.6*FEs	0.7*FEs	0.8*FEs	0.9*FEs	1.0*FEs
f_1	1.75e-42	3.63e-40	3.67e-39	7.20e-39	1.08e-38	1.31e-38	4.09e-39	4.65e-39	4.57e-39	4.58e-39	7.86e-39
f_2	1.04e+01	7.21e+00	1.54e+01	1.12e+01	1.91e+01	1.64e+01	2.40e+01	2.21e+01	2.58e+01	3.13e+01	3.50e+01
f_3	3.00e-08	3.06e-08	3.06e-08	3.25e-08	3.35e-08	3.57e-08	3.91e-08	4.67e-08	3.72e-04	1.49e+00	3.81e+00
f_4	4.43e-41	5.08e-39	5.15e-38	6.20e-38	9.20e-38	1.30e-37	5.31e-38	9.10e-38	6.40e-38	6.98e-38	9.39e-38
f_5	8.81e-06	8.18e-06	2.17e-03	8.30e-06	1.16e-05	3.03e-02	9.91e-03	7.96e-02	3.55e-01	1.65e+00	4.71e+00
f_6	9.31e-09	3.42e-09	1.83e-09	1.18e-09	1.55e-09	1.60e-09	1.33e-09	1.25e-09	1.65e-09	1.69e-09	1.90e-09
f_7	5.36e-23	1.09e-21	1.38e-21	1.40e-21	1.71e-21	1.72e-21	1.27e-21	1.35e-21	1.53e-21	1.36e-21	1.66e-21
f_8	6.16e-02	4.87e-02	4.56e-02	4.46e-02	4.25e-02	4.36e-02	4.07e-02	4.23e-02	4.21e-02	4.02e-02	3.79e-02
f_9	1.99e-83	2.95e-81	1.90e-77	1.85e-78	5.63e-78	1.94e-75	1.65e-78	2.28e-77	6.15e-75	1.70e-79	1.48e-79
f_{10}	4.98e-03	6.42e-03	8.64e-03	6.48e-03	4.95e-03	7.46e-03	7.19e-03	7.07e-03	6.80e-03	7.34e-03	7.46e-03
f_{11}	1.81e-01	3.15e-01	3.76e-01	2.50e-01	3.92e-01	2.85e-01	2.12e-01	2.34e-01	3.40e-01	2.00e-01	2.36e-01
f_{12}	1.76e+01	1.85e+01	1.69e+01	1.71e+01	1.81e+01	1.78e+01	1.77e+01	1.67e+01	1.75e+01	1.73e+01	1.69e+01
f_{13}	7.46e+00	6.96e+00	7.10e+00	7.37e+00	7.62e+00	7.36e+00	7.36e+00	7.12e+00	7.48e+00	7.22e+00	7.59e+00
f_{14}	2.18e-01	1.45e-01	1.45e-01	1.53e-01	1.17e-01	1.85e-01	1.85e-01	2.59e-01	2.88e-01	4.89e-01	8.80e-01
f_{15}	2.43e-01	3.23e-01	2.63e-01	2.98e-01	3.47e-01	2.96e-01	3.30e-01	3.37e-01	4.36e-01	8.04e-01	2.11e+00
f_{16}	2.10e+00	1.51e+00	1.81e+00	1.75e+00	1.99e+00	1.58e+00	1.81e+00	2.61e+00	2.40e+00	2.25e+00	2.19e+00
f_{17}	2.61e-35	6.46e-34	7.21e-34	1.37e-33	1.74e-33	2.30e-33	7.06e-34	1.26e-33	1.33e-33	1.15e-33	1.48e-33
f_{18}	4.58e+03	4.36e+03	3.89e+03	4.62e+03	4.13e+03	4.42e+03	4.32e+03	4.55e+03	4.62e+03	4.63e+03	4.56e+03
f_{19}	4.05e-03	2.79e-03	4.38e-03	4.51e-03	5.22e-03	5.34e-03	3.96e-03	4.37e-03	6.13e-03	1.06e-02	2.33e-01
f_{20}	4.11e-01	3.94e-01	3.91e-01	3.96e-01	3.97e-01	3.92e-01	3.93e-01	3.96e-01	3.96e-01	4.00e-01	3.90e-01

TABLE 14. Rank of the average values with different starting iteration for CDDS.

	0.0*FEs	0.1*FEs	0.2*FEs	0.3*FEs	0.4*FEs	0.5*FEs	0.6*FEs	0.7*FEs	0.8*FEs	0.9*FEs	1.0*FEs
f_1	1	2	3	8	10	11	4	7	5	6	9
f_2	2	1	4	3	6	5	8	7	9	10	11
f_3	1	3	2	4	5	6	7	8	9	10	11
f_4	1	2	3	5	9	11	4	8	6	7	10
f_5	3	1	5	2	4	7	6	8	9	10	11
f_6	11	10	8	1	4	5	3	2	6	7	9
f_7	1	2	6	7	10	11	3	4	8	5	9
f_8	11	10	9	8	6	7	3	5	4	2	1
f_9	1	2	8	6	7	10	5	9	11	4	3
f_{10}	2	3	11	4	1	9	7	6	5	8	10
f_{11}	1	8	10	6	11	7	3	4	9	2	5
f_{12}	7	11	2	4	10	9	8	1	6	5	3
f_{13}	8	1	2	7	11	5	6	3	9	4	10
f_{14}	7	3	2	4	1	5	6	9	8	10	11
f_{15}	1	5	2	4	8	3	6	7	9	10	11
f_{16}	7	1	4	3	6	2	5	11	10	9	8
f_{17}	1	2	4	8	10	11	3	6	7	5	9
f_{18}	8	4	1	10	2	5	3	6	9	11	7
f_{19}	3	1	5	6	7	8	2	4	9	10	11
f_{20}	11	5	2	6	9	3	4	8	7	10	1
best ranking	8	5	1	1	2	0	0	1	0	0	2
rank sum	88	77	93	106	137	140	96	123	155	145	160
Overall rank	2	1	3	5	7	8	4	6	10	9	11

during operation period t(MW), Δt is the operation period(h), k_m is the power coefficient for m-th plant, $q_{m,t}$ is the release in operation period t from plant m(m^3/s) and $h_{m,t}$ is hydraulic head of m-th plant during t-th operation period (m).

C. CONSTRAINS

There are several constraints proposed on reservoir operation as follows.

(1)Water connection of cascade reservoirs:

$$I_{m+1,t} = Q_{m,t} + q_{m,t}$$

where $I_{m+1,t}$ is the inflow of $(m+1)$-th reservoir in t-th operation period(m^3/s), $Q_{m,t}$ is the outflow of m-th reservoir in t-th operation period(m^3/s), $q_{m,t}$ is the inter-zone inflow into $(m+1)$-th reservoir in t-th operation period(m^3/s).

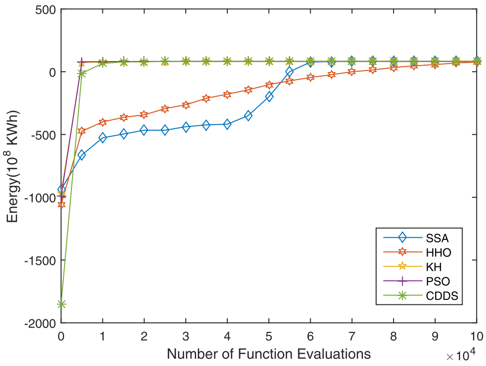
(2) Water balance constraint:

$$V_{m,t+1} = V_{m,t} + (I_{m,t} - Q_{m,t} - S_{m,t}) \times \Delta t$$

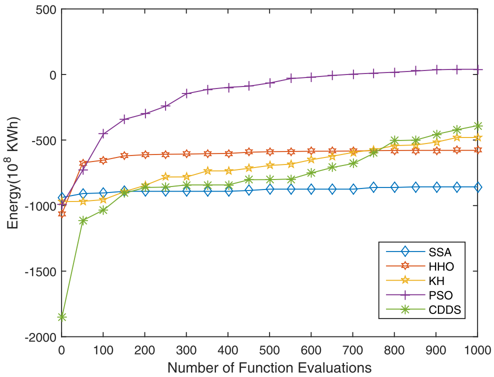
where $V_{m,t+1}$ and $V_{m,t}$ are the initial storage volume of m-th reservoir in the beginning of $(t+1)$ and t-th operation period(m^3/s), respectively; $S_{m,t}$ is the water spill of m-th reservoir in t-th operation period(m^3/s).

(3)Water level constraint:

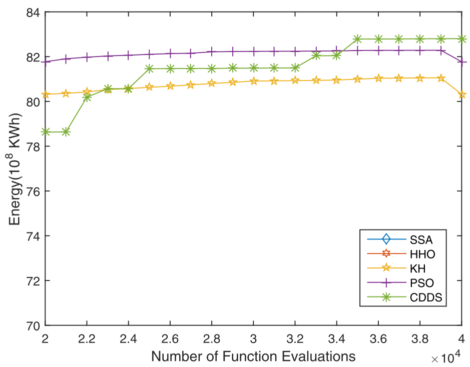
$$Z_{m,t}^{min} \leq Z_{m,t} \leq Z_{m,t}^{max}$$



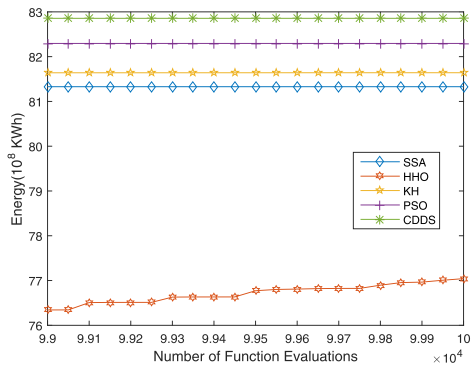
(a) The whole search process



(b) The early stage of search process



(c) The middle stage of search process



(d) The final stage of search process

FIGURE 6. The convergence curves for the CROO problem.

TABLE 15. Parameters of the Shuibuya and Geheyan hydropower stations in Qingjiang River.

Reservoir items	Units	SBY	GHY
Dead water level	m	350	175
Normal water level	m	400	200
Flood control level	m	392	194
Total storage	10^8 m^3	45.89	30.18
Beneficial reservoir capacity	10^8 m^3	23.83	19.75
Installed capacity	MW	1840	1200
Guaranteed output	MW	310	187
Regulation ability		multi-year	annual
Power coefficient		8.5	8.5

where $Z_{m,t}^{min}$ and $Z_{m,t}^{max}$ are the minimum and maximum limits of water level for m-th reservoir in t-th operation period(m), respectively.

(4)Outflow constraint:

$$Q_{m,t}^{min} \leq Q_{m,t} \leq Q_{m,t}^{max}$$

where $Q_{m,t}^{min}$ and $Q_{m,t}^{max}$ are the minimum and maximum limits of outflow for m-th reservoir in t-th operation period(m$^3/s$), respectively.

(5) Power output constraint:

$$N_{m,t}^{min} \leq N_{m,t} \leq N_{m,t}^{max}$$

where $N_{m,t}^{min}$ and $N_{m,t}^{max}$ are the minimum and maximum limits of the power output for m-th hydro plant in t-th operation period(MW), respectively.

(6)Boundary constraint:

$$Z_{m,1} = Z_{m,b}, \quad Z_{m,T+1} = Z_{m,e}$$

where $Z_{m,b}$ is the water level of m-th reservoir in the beginning of the whole operation time(m); $Z_{m,e}$ is the water level of m-th reservoir at end of the whole operation time(m).

D. CONSTRAINT HANDLING AND PARAMETER SETTINGS

1) CONSTRAINT HANDLING

Due to complex constraint conditions of the whole operation time, we choose penalty function methods to handle constraint. The fitness function can be expressed as follows:

$$F = -\sum_{t=1}^{T}\sum_{m=1}^{M}[N_{m,t} - H\delta \cdot (N_{m,f} - N_{m,t})^k]\Delta t$$

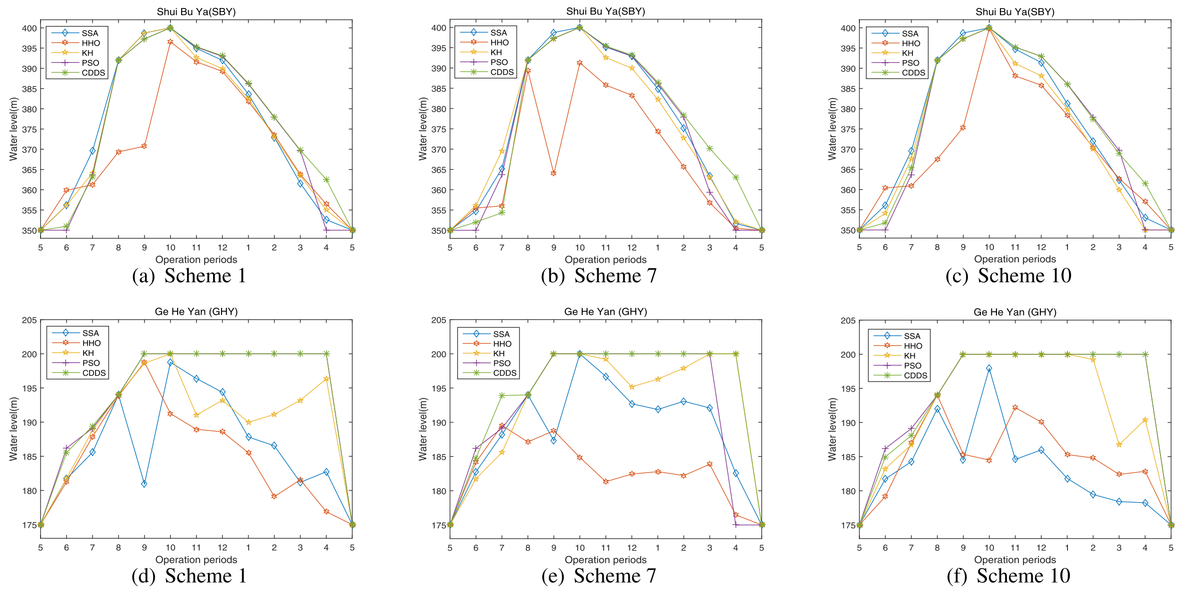
and

$$\delta = \begin{cases} 1, & N(m,t) < N_{m,f} \\ 0, & N(m,t) \geq N_{m,f} \end{cases}$$

where H, δ and k are penalty parameters and $N_{m,f}$ is the guaranteed output for m-th hydro plant. Note that $N_{m,f} - N_{m,t}$ must be positive. Furthermore, k usually takes the value of either 1 or 2. In our experiments it is set to 1, and H is set to 1.

TABLE 16. The optimal operation results of Qingjiang cascade hydropower station by using GA, BA, KH and CDDS.

Run index	SSA	HHO	KH	PSO	CDDS
1	81.08020	73.11664	82.18316	82.89737	82.88440
2	81.41665	81.26605	81.81009	81.74482	83.00378
3	81.44476	80.14619	81.68512	82.07968	82.87252
4	80.96646	80.79414	81.30341	82.78816	82.71648
5	81.47258	79.60977	81.44673	82.89737	82.81155
6	81.67930	80.07626	81.10175	82.07311	82.66900
7	81.58144	75.81833	82.83041	82.24390	82.64748
8	81.90717	75.50025	79.40675	82.67621	83.04438
9	81.41306	71.86648	82.22762	80.65473	82.93714
10	80.32426	72.22710	82.41153	82.89737	82.97515
Best	81.90717	81.26605	82.83041	82.89737	83.04438
Worst	80.32426	71.86648	79.40675	80.65473	82.64748
Med	81.43071	77.71405	81.74760	82.46005	82.87846
Mean	81.32859	77.04212	81.64066	82.29527	82.85619
Std	0.44376	3.75069	0.94935	0.71247	0.14106
p	0.00195(+)	0.00195(+)	0.02148(+)	0.34375(=)	



(a) Scheme 1 (b) Scheme 7 (c) Scheme 10

(d) Scheme 1 (e) Scheme 7 (f) Scheme 10

FIGURE 7. Monthly water level process of Shuibuya and Geheyan obtained by different algorithms for three representative schemes.

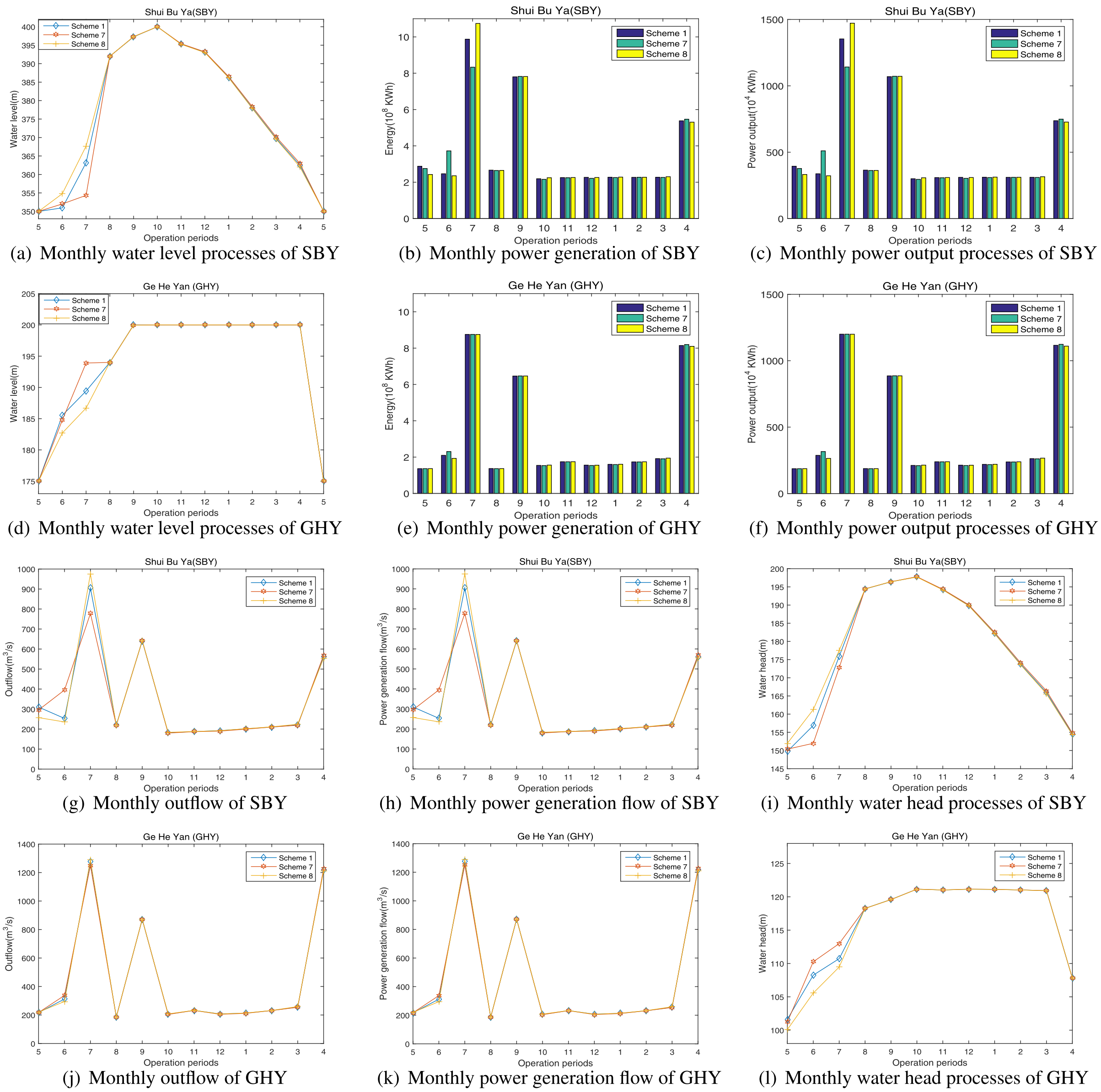
2) PARAMETER SETTINGS

To demonstrate the practicability and effectiveness of CDDS in the application of the CROO problem, CDDS is compared with SSA [14], HHO [15], KH [11], and PSO [5].

The maximum number of function evaluation (FEs), here set to 10^5, is selected as the stopping criterion for all algorithms. The population size is 50 for all population-based algorithms. The other control parameters of the compared algorithms are given as follows: For SSA, parameter c_1 is defined as in the paper [14], c_2 and c_3 are random numbers uniformly generated in the interval of [0,1]. For HHO, parameters r_1, r_2, r_3, r_4 and q are random numbers inside (0,1), which are updated in each iteration. For KH, the foraging speed is $V_f = 0.02$, the maximum diffusion speed is $D^{max} = 0.005$, the maximum induced speed is $N^{max} = 0.01$. For PSO, the inertia weight ω: $0.9 - 0.2$, the learning factors $c_1 = 2$,$c_2 = 2$.

E. RESULTS ANALYSIS AND DISCUSSION

Table 16 shows the results for the optimal operation of the Qingjiang cascade hydropower stations in 10 runs by using CDDS, SSA, HHO, KH, and PSO. It can be observed that the mean annual power generation obtained by CDDS is evidently excellent. Compared with SSA, HHO, KH, and PSO, the mean annual power generation obtained by CDDS increased by 1.88%, 7.55%, 1.49%, 0.68%. The results indicate that CDDS can maximize the power generation by hydropower station and increase the economic benefit. This conclusion that the standard deviation (Std) of CDDS is smaller than that of the other four algorithms reveals CDDS's good stability. It's remarkable that HHO is unstable in terms of the Std of HHO. The Wilcoxon signed-rank test with a 5% significant level is used to further study the differences among the compared algorithms, and the statistical results are shown in the last row of Table 16. According to Wilcoxon signed



(a) Monthly water level processes of SBY
(b) Monthly power generation of SBY
(c) Monthly power output processes of SBY
(d) Monthly water level processes of GHY
(e) Monthly power generation of GHY
(f) Monthly power output processes of GHY
(g) Monthly outflow of SBY
(h) Monthly power generation flow of SBY
(i) Monthly water head processes of SBY
(j) Monthly outflow of GHY
(k) Monthly power generation flow of GHY
(l) Monthly water head processes of GHY

FIGURE 8. Reservoir operation processes obtained by CDDS for three representative schemes.

rank test, CDDS significantly outperforms SSA, HHO, and KH, while it has the same performance as PSO. The comparison results reveal that CDDS can effectively handle the problem with a series of equality and inequality constraints.

In order to further investigate the performance of five algorithms, the convergence curves of mean annual power generation for reservoirs are shown in Figure 6. The convergence curve of CDDS ascends fast in the early stage and has rapid convergence during the middle and late stage. At last, the convergence curve reaches the best in the convergence stage. It reveals the improvement for CDDS can effectively improve performance. In initial stage, the search with a single point leads to poor inital solution. The new Gaussian mutation operator, which improves the exploitation of CDDS, leads to deeply search so that search curve presents the behavior of rapid rise. The quality of the optimal solution obtained by CDDS wins the second place of the five algorithms after 1000 FEs. But CDDS reduces to the third owing to strong local search ability during middle stage. Then the chaotic search is used to jump from local optimal solution. CDDS is pulled up from the third to the second place at 22000 FEs to about 26000 FEs, and it maintains the first in the search process after 32000 FEs. Finally, the performance of CDDS is still the best in the late stage. In sum, CDDS can effectively enhance the exploration and exploitation ability so as to obtain high-quality solution.

Then, the monthly water level processes of each reservoir obtained by different algorithms on three representative operation results (i.e., scheme 1, scheme 7, and scheme 10) are shown in Figure 7. It can be observed that PSO, KH, and CDDS operate at high water level for three schemes, respectively. The high water level can take advantage of the higher hydraulic head to improve power generation. The data in Table 16 can confirm that fact. With regard to three schemes, the performances of HHO are both the worst primarily because the water levels obtained by HHO are low.

To further verify the effectiveness of CDDS for handling the CROO Problem, the reservoir operation processes of each reservoir for three representative schemes (i.e., scheme 1, scheme 7, and scheme 8) obtained by CDDS are presented in Figure 8. From Table 16, in the light of annual power generation, there is slight difference between the three schemes. With regard to 10 runs, scheme 8 gets the best annual power generation, scheme 7 obtains the worst annual power generation.

First of all, the monthly water levels are all in the boundaries of constraints, which means the complex constraints are solved effectively. From Figure 8, the monthly water levels of the two reservoirs for three representative schemes are essentially uniform except for June and July. The principal reason is that we have a rainy season from April to October and the reservoir inflow is abundant which is conducive to optimal operation.

Whereafter, we can analyze the power generation processes and the power output processes. The power generation and the power output present out the same change trend. As can be noticed, the power output and the power generation from April to September are all obvious more than those from October to March. In fact, these months are in the rainy season thus the reservoirs can increase the inflow to increase the power generation.

Finally, we pay attention to the outflow processes, the power generation flow processes, and the water head processes. The outflow processes are consistent with the power generation flow processes, which means no water spill for two reservoirs. We also realize that the trend of the power generation processes is in line with that of the outflow processes. The outflow of SBY in May to July, September, and next April is higher than that of other months, so a large quantity of water is discharged from reservoirs to increase power generation, which can be seen from the monthly power generation processes. The outflow of GHY in June, July, September, and next April also can get more power generation. As regards the water head processes, there is reasonable concordance between the monthly water level processes and the water head processes. It's obvious that the water head processes satisfy the practical requirement.

Thus, the experiment on cascade reservoirs operation optimization problem shows that CDDS can perform well in the constrained practical problems. CDDS is an effective method in solving the constrained problems because it can effectively harmonize the exploration and exploitation propensities.

VIII. CONCLUSION

In this article, we proposed a chaotic-DDS based on chaotic initialization, a new Gaussian mutation operator and a chaotic search. Using the chaotic map to generate the initial population can enhance the exploration of the proposed algorithm and improve the quality of initial solution. The strategy updating the candidate solution by a new Gaussian mutation operator can improve the exploitation ability of the proposed algorithm and conduce to high accuracy solution and fast convergence. Utilizing a chaotic search to jump out of the local optimal can enhance the exploration. The performance of CDDS and the influence of three strategies were testified by a set of experiment series. 13 chaotic maps were employed in CDDS on 20 classical benchmark functions to choose the best version of CDDS. Then the best chaotic DDS was compared with the state-of-art algorithms through 20 benchmark functions, CEC2005, CEC 2014. Additionally, a series of experiments are conducted to study the effect of three strategies. The experimental results reveal that the proposed CDDS algorithm outperformed the compared algorithms in terms of solution accuracy and convergence speed, well balanced exploration and exploitation. The statistical results prove the enhancements of CDDS than other algorithms were significant. Thereafter, a cascade reservoirs operation optimization problem was used to further investigate the feasibility and effectiveness. The experimental and graphical results indicate that CDDS is a competitive algorithm for the constrained practical problems.

We designed the algorithm by incorporating three strategies into DDS. Due to the no free lunch theorem, our algorithm gets better performance at little cost of time consuming. In future works, other strategies and ideas can be employed to overcome this weakness.

ACKNOWLEDGMENT

The authors would like thank to anonymous reviewers for providing their constructive comments.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[2] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

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

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Nov. 2002, pp. 1942–1948.

[6] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[7] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2008.

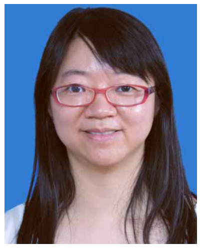
[8] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[9] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput.*, 2010, pp. 210–214.

[10] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperat. Strategies Optim.*, vol. 284, pp. 65–74, Jan. 2010.

[11] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4831–4845, Dec. 2012.
[12] R. Venkata Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *Int. J. Ind. Eng. Computations*, vol. 7, no. 1, pp. 19–34, 2016.
[13] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
[14] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
[15] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[16] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.
[17] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.
[18] P. Agarwalla and S. Mukhopadhyay, "Efficient player selection strategy based diversified particle swarm optimization algorithm for global optimization," *Inf. Sci.*, vols. 397–398, pp. 69–90, Aug. 2017.
[19] Q. Jiang, L. Wang, X. Hei, J. Cheng, X. Lu, Y. Lin, and G. Yu, "ARAe-SOM+BCO: An enhanced artificial raindrop algorithm using self-organizing map and binomial crossover operator," *Neurocomputing*, vol. 275, pp. 2716–2739, Jan. 2018.
[20] N. Lynn and P. N. Suganthan, "Ensemble particle swarm optimizer," *Appl. Soft Comput.*, vol. 55, pp. 533–548, Jun. 2017.
[21] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Inf. Sci.*, vol. 423, pp. 172–186, Jan. 2018.
[22] H. Lourenço, O. Martin, and T. Stützle, *Iterated Local Search: Framework Appl.*, vol. 146, pp. 363–397, Sep. 2010.
[23] P. Hansen, N. Mladenović, and J. A. M. Pérez, "Variable neighborhood search," *Eur. J. Oper. Res.*, vol. 191, no. 3, pp. 593–595, 2008.
[24] B. Doğan and T. Ölmez, "A new Metaheuristic for numerical function optimization: Vortex search algorithm," *Inf. Sci.*, vol. 293, pp. 125–145, Feb. 2015.
[25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, no. 3, pp. 46–61, 2014.
[26] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[27] A. Shefaei and B. Mohammadi-Ivatloo, "Wild goats algorithm: An evolutionary algorithm to solve the real-world optimization problems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2951–2961, Jul. 2018.
[28] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, no. 3, p. 35, 2013.
[29] B. Li and W. Jiang, "Chaos optimization method and its application," *Control Theory Appl.*, vol. 4, pp. 613–615, Aug. 1997.
[30] J. Feng, J. Zhang, X. Zhu, and W. Lian, "A novel chaos optimization algorithm," *Multimedia Tools Appl.*, vol. 76, pp. 17405–17436, Aug. 2017.
[31] B. Alatas, E. Akin, and A. B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons Fractals*, vol. 40, no. 4, pp. 1715–1734, May 2009.
[32] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Syst. Appl.*, vol. 37, no. 8, pp. 5682–5687, Aug. 2010.
[33] B. Alatas, "Chaotic harmony search algorithms," *Appl. Math. Comput.*, vol. 216, no. 9, pp. 2687–2699, Jul. 2010.
[34] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *J. Comput. Sci.*, vol. 5, no. 2, pp. 224–232, Mar. 2014.
[35] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Inf. Sci.*, vol. 274, pp. 17–34, Aug. 2014.
[36] S. Mirjalili and A. H. Gandomi, "Chaotic gravitational constants for the gravitational search algorithm," *Appl. Soft Comput.*, vol. 53, pp. 407–419, Apr. 2017.
[37] W.-L. Xiang and M.-Q. An, "An efficient and robust artificial bee colony algorithm for numerical optimization," *Comput. Oper. Res.*, vol. 40, no. 5, pp. 1256–1265, May 2013.
[38] R. A. Ibrahim, M. A. Elaziz, and S. Lu, "Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Syst. Appl.*, vol. 108, pp. 1–27, Oct. 2018.
[39] X. Zhang and T. Feng, "Chaotic bean optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 67–77, Jan. 2018.
[40] X. Xu, H. Rong, M. Trovati, M. Liptrott, and N. Bessis, "CS-PSO: Chaotic particle swarm optimization algorithm for solving combinatorial optimization problems," *Soft Comput.*, vol. 22, no. 3, pp. 783–795, Feb. 2018.
[41] B. A. Tolson and C. A. Shoemaker, "Dynamically dimensioned search algorithm for computationally efficient watershed model calibration," *Water Resour. Res.*, vol. 43, no. 1, pp. 208–214, Jan. 2007.
[42] R. Arsenault, A. Poulin, P. Côté, and F. Brissette, "Comparison of stochastic optimization algorithms in hydrological model calibration," *J. Hydrologic Eng.*, vol. 19, no. 7, pp. 1374–1384, Jul. 2014.
[43] Y. Wu, D. Li, and B. Ming, "Novel dynamically dimensioned search algorithms for very high dimensional optimization problems," unpublished.
[44] R. G. Regis and C. A. Shoemaker, "Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization," *Eng. Optim.*, vol. 45, no. 5, pp. 529–555, May 2013.
[45] X. L. Huang and J. Xiong, "Parameter optimization of multi-tank model with modified dynamically dimensioned search algorithm," in *Proc. 3rd Int. Symp. Comput. Sci. Comput. Technol.*, 2010, vol. 5978, no. 1, pp. 283–288.
[46] H. Yen, J. Jeong, and D. R. Smith, "Evaluation of dynamically dimensioned search algorithm for optimizing SWAT by altering sampling distributions and searching range," *J. Amer. Water Resour. Assoc.*, vol. 52, no. 2, pp. 443–455, Apr. 2016.
[47] J. Chu, Y. Peng, W. Ding, and Y. Li, "A heuristic dynamically dimensioned search with sensitivity information (HDDS-S) and application to river basin management," *Water*, vol. 7, no. 12, pp. 2214–2238, May 2015.
[48] X. Huang, W. Liao, X. Lei, Y. Jia, Y. Wang, X. Wang, Y. Jiang, and H. Wang, "Parameter optimization of distributed hydrological model with a modified dynamically dimensioned search algorithm," *Environ. Model. Softw.*, vol. 52, pp. 98–110, Feb. 2014.
[49] F. Lespinas, A. Dastoor, and V. Fortin, "Performance of the dynamically dimensioned search algorithm: Influence of parameter initialization strategy when calibrating a physically based hydrological model," *Hydrol. Res.*, vol. 49, no. 4, pp. 971–988, Aug. 2018.
[50] S. Kang, S. Lee, and T. Kang, "Development and application of storage-zone decision method for long-term reservoir operation using the dynamically dimensioned search algorithm," *Water Resour. Manage.*, vol. 31, no. 1, pp. 1–14, 2017.
[51] L. S. Matott, K. Leung, and J. Sim, "Application of MATLAB and Python optimizers to two case studies involving groundwater flow and contaminant transport modeling," *Comput. Geosci.*, vol. 37, no. 11, pp. 1894–1899, Nov. 2011.
[52] J. M. Dias, H. Rocha, B. Ferreira, and M. D. C. Lopes, "IMRT beam angle optimization using DDS with a cross-validation approach for configuration selection," in *Proc. ICCSA, II*, Guimarães, Portugal, 2014, pp. 1–26.
[53] B. A. Tolson, M. A. Esfahani, A. C. Zecchin, and H. R. Maier, "A new algorithm for water distribution system optimization: Discrete dynamically dimensioned search," in *Proc. World Environ. Water Resour. Congr.*, May 2008, pp. 1–4.
[54] B. A. Tolson, M. Asadzadeh, H. R. Maier, and A. Zecchin, "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization: HD-DDS ALGORITHM," *Water Resour. Res.*, vol. 45, no. 12, Dec. 2009.
[55] J. Xiong, Y. Ohnish, K. Takahashi, and T. Koyama, "Parameter determination of multi-tank model with dynamically dimensioned search," in *Proc. Symp. Rock Mech. Jpn.*, vol. 38, Jan. 2009, pp. 19–24.
[56] M. J. F. Macêdo, M. F. P. Costa, A. M. A. Rocha, and E. W. Karas, "Combining filter method and dynamically dimensioned search for constrained global optimization," in *Proc. ICCSA, III*, Trieste, Italy, 2017, pp. 119–134.
[57] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.
[58] P. L. Boyland, "Bifurcations of circle maps: Arnol'd tongues, bistability and rotation intervals," *Commun. Math. Phys.*, vol. 106, no. 3, pp. 353–381, Sep. 1986.
[59] M. Bucolo, R. Caponetto, L. Fortuna, M. Frasca, and A. Rizzo, "Does chaos work better than noise?" *IEEE Circuits Syst. Mag.*, vol. 2, no. 3, pp. 4–19, Mar. 2002.
[60] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, Jan. 2013.
[61] H. O. Peitgen, H. Jürgens, and D. Saupe, *Chaos Fractals*. Berlin: Germany: Springer-Verlag, 1992.

[62] R. Caponetto, L. Fortuna, S. Fazzino, and M. G. Xibilia, ''Chaotic sequences to improve the performance of evolutionary algorithms,'' *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 289–304, Jun. 2003.
[63] X. Yuan, J. Zhao, Y. Yang, and Y. Wang, ''Hybrid parallel chaos optimization algorithm with harmony search algorithm,'' *Appl. Soft Comput.*, vol. 17, pp. 12–22, Apr. 2014.
[64] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, ''Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,'' Nanyang Technol. Univ., Singapore, Kanpur Genetic Algorithms Lab., IIT Kanpur, Kanpur, India, Tech. Rep. 2005005, May 2005.
[65] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, ''Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems,'' *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
[66] J. J. Liang, B. Y. Qu, and P. N. Suganthan, ''Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization,'' Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep. 201311, Dec. 2013.
[67] M.-Y. Cheng and D. Prayogo, ''Symbiotic organisms search: A new Metaheuristic optimization algorithm,'' *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.
[68] N. Hansen, S. D. Müller, and P. Koumoutsakos, ''Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES),'' *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003.
[69] P. Agarwalla and S. Mukhopadhyay, ''Hybrid advanced player selection strategy based population search for global optimization,'' *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112825.
[70] R. Tanabe and A. S. Fukunaga, ''Improving the search performance of shade using linear population size reduction,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.
[71] B. Ming, J.-X. Chang, Q. Huang, Y.-M. Wang, and S.-Z. Huang, ''Optimal operation of multi-reservoir system based-on cuckoo search algorithm,'' *Water Resour. Manage.*, vol. 29, no. 15, pp. 5671–5687, Dec. 2015.
[72] B. Ming, Q. Huang, Y. M. Wang, D. F. Liu, and T. Bai, ''Cascade reservoir operation optimization based-on improved cuckoo search,'' *Shui Li Xue Bao*, vol. 46, no. 3, pp. 341–349, 2015.
[73] L. Su, K. Yang, H. Hu, and Z. Yang, ''Long-term hydropower generation scheduling of large-scale cascade reservoirs using chaotic adaptive multi-objective bat algorithm,'' *Water*, vol. 11, no. 11, p. 2373, Nov. 2019.
[74] Z. Jiang, H. Qin, C. Ji, Z. Feng, and J. Zhou, ''Two dimension reduction methods for multi-dimensional dynamic programming and its application in cascade reservoirs operation optimization,'' *Water*, vol. 9, no. 9, p. 634, Aug. 2017.
[75] Z. Yang, K. Yang, H. Hu, and L. Su, ''The cascade reservoirs multi-objective ecological operation optimization considering different ecological flow demand,'' *Water Resour. Manage.*, vol. 33, no. 1, pp. 207–228, Jan. 2019.



YUN WU received the B.S. degree from Central China Normal University, Wuhan, China, in 2004, and the M.S. degree from Capital Normal University, Beijing, China, in 2010. She is currently pursuing the Ph.D. degree in mathematics with Wuhan University, Wuhan.

Since 2004, she has been a Lecturer with the School of Science, Jiujiang University, China. Her research interests include computational intelligence and evolutionary algorithms.

JUNFENG LIU received the B.S. degree from Central Xinyang Normal University, Xinyang, China, in 2012, and the M.S. degree from Wenzhou University, Wenzhou, China, in 2016. He is currently pursuing the Ph.D. degree in mathematics with Wuhan University, Wuhan, China. His research interests include computational intelligence and evolutionary algorithms.

DINGFANG LI received the B.S. degree from Xiamen University, Xiamen, China, and the M.S. and Ph.D. degrees from Wuhan University, Wuhan, China.

He is currently a Professor with the School of Mathematics and Statistics, Wuhan University. He has published more than 50 international peer-reviewed journal/conference papers. His research interests include computational intelligence, evolutionary algorithms, and machine learning.