Received July 9, 2019, accepted July 21, 2019, date of publication July 30, 2019, date of current version August 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931910

Modified Dolphin Swarm Algorithm Based on Chaotic Maps for Solving High-Dimensional Function Optimization Problems

WEIBIAO QIAO[1,2] AND ZHE YANG[3]
[1]School of Environmental and Municipal Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China
[2]Sinopec Petroleum Engineering Zhongyuan Company Ltd., Zhengzhou 450046, China
[3]School of Computer Science, The University of Manchester, Manchester M13 9PL, U.K.

Corresponding author: Weibiao Qiao (wbq0408@163.com)

This work was supported by the High-Level Talents Start-Up Project of North China University of Water Resources and Electric Power.

ABSTRACT In 2016, dolphin swarm algorithm (DSA) that has received sustained research interest due to its simplicity and effectiveness was proposed. However, when solving high-dimensional function optimization problems, DSA is prone to fall into local optimization problems, which leads to low optimization accuracy or even failure. In this paper, to solve this problem, chaotic mapping is introduced into DSA, and chaotic dolphin swarm algorithm (CDSA) is successfully proposed. Based on high-dimensional Rastrigin function, the optimal chaotic map is determined among eight chaotic maps (e.g., Logistic). Then, in view of high-dimensional Levy function, Rotated Hyper-Ellipsoid function and Sum Squares function respectively, the performance of CDSA and that of the state-of-the-art algorithms (e.g. (whale optimization algorithm) WOA) are compared. The results show that the performance of CDSA based on Kent map is best and the performance of CDSA outperform that of the state-of-the-art algorithms considered to be compared. Finally, it is concluded that such a new meta-heuristic algorithm could help to improve the shortcomings of DSA and increase the applied range of DSA.

INDEX TERMS Chaotic maps, dolphin swarm algorithm, high-dimensional function, optimization.

I. INTRODUCTION

As an important branch in the field of optimization, the optimization problems of high-dimensional functions have always been a hot issue for scholars at home and abroad [1]. High-dimensional function optimization has a significant application in theory and engineering field. Many practical optimization problems which may be solved can be transformed into optimization problems of high-dimensional functions through certain transformations, such as multi-variable function fitting [2].

However, in the process of optimizing high-dimensional functions, with the increase of their dimensions, the scale of search space increases exponentially, this lead to that traditional optimization methods can't meet the needs of solving [3]. Therefore, in recent years, scholars at home and abroad have tried to use the meta-heuristic algorithm to solve the optimization problem of high-dimensional functions, these research results are shown in TABLE 1.

From TABLE 1, we can see that all kinds of meta-heuristic algorithms are inspired by particle swarm optimization (PSO). Although the above-mentioned meta-heuristic algorithms obtain the optimal solution of the function in a certain scale, the algorithms above still have the problem of easily falling into the local optimum. At the same time, the algorithms above are still limited to scale problems, e.g., with less than 30 decision variables. Therefore, some improved algorithms are proposed, which are shown in TABLE 2.

For TABLE 2, some scholars have proposed some improved meta-heuristic algorithms, but these enhanced meta-heuristic algorithms only solve functions in low-dimensional space, for example, some functions have only two or three variables, so it is very significant to find meta-heuristic algorithms that can be solved in high-dimensional space.

The associate editor coordinating the review of this article and approving it for publication was Seyedali Mirjalili.

TABLE 1. Some mete-heuristic algorithms for solving high-dimensional fuction optimization problem in recent years.

Algorithm name	Occurrence	Results or conclusion	Reference
Decimal-coding small world optimization algorithm (DSWOA)	2009	DSWOA can acquire a satisfactory solution, also has better stability and a fast convergence rate.	[4]
Improving particle swarm optimization (IPSO)	2010	Simulation results demonstrate that the proposed method is more stable and efficient than several other existing ways.	[5]
Disturbance chaotic ant swarm algorithm (DCASA)	2011	The results show that the proposed algorithm is useful as well as efficient for the complex high-dimensional optimization problems.	[6]
Chaotic gaussian particle swarm optimization (CGPSO)	2011	It can be safely concluded that the proposed CGPSO is an efficient optimization scheme for solving high-dimensional problems	[7]
Modified Hestenes and Stiefel conjugate gradient (MHSCG)	2013	Numerical results show that the proposed algorithm is advantageous to existing CG methods for large-scale optimization problems	[8]
Cooperative coevolution orthogonal artificial bee colony (CCOABC)	2013	the simulation results demonstrate that CCOABC is a highly competitive algorithm for solving high-dimensional function.	[9]
Multi-scale quantum harmonic oscillator algorithm (MSQHOA)	2013	The results show that the multi-scale quantum harmonic oscillator algorithm gets precise global optimum for high-dimensional function.	[10]
Improved glowworm swarm optimization algorithm (IGSOA)	2013	The experimental results indicate that IGSO has better ability of global optimization and higher success ratio.	[11]
Artificial bee colony algorithm (ABCA)	2013	The experimental results show that the algorithm is useful to solve the high-dimensional complex optimization problem.	[12]
Differential evolution algorithm (DEA)	2014	The results show that the new algorithm is effective and efficient for high-dimensional optimization	[13]
Greedy randomized adaptive search procedure (GRASP).	2015	Experimental results show the supremacy of the proposed method over previous versions of GRASP for feature selection.	[14]
Surrogate-assisted cooperative swarm optimization (SACSO)	2017	Empirical studies demonstrate that the proposed algorithm can find high-quality solutions for high-dimensional problems	[15]
Surrogate-assisted hierarchical particle swarm optimization (SAHPSO)	2018	Our experimental results demonstrate that the proposed method is competitive compared with the state-of-the-art algorithms under a limited computational budget.	[16]
Improved grey wolf optimization algorithm (IGWOA)	2018	The results show that the proposed algorithm can find more accurate solutions and has a higher convergence rate	[17]
Incremental gravitational search algorithm (IGSA)	2018	It is observed that the IGSA-3 algorithm is better than the IGSA-1 and two algorithms in that its appropriateness, stability, and duration.	[18]

TABLE 2. Some imoroved meta-heuristic algorithms in recent years.

Algorithm name	Occurrence	Reference
Particle swarm ant colony optimization (PSACO)	2007	[19]
Particle swarm optimization gravitational search algorithm (PSOGSA).	2014	[20]
Modified differential evolution whale optimization algorithm (MDEWOA)	2018	[21]
Particle swarm grey wolf optimization algorithm (PSGWOA)	2018	[22]
Hybrid firefly particle swarm optimization (HFPSO)	2018	[23]
Genetic particle swarm optimization algorithm (GPSOA)	2018	[24]
Simulated annealing moth flame optimization (SAMFO)	2018	[25]
Wolf pack search local search (WPSLS)	2019	[26]

In 2016, a novel meta-heuristic algorithm ([27]–[29]) called dolphin swarm algorithm (DSA) is proposed and applied. But like other meta-heuristic algorithms, DSA still has the problem of an optimal balance between exploration and exploitation, Therefore, to solve this problem and enhance the convergence speed and the ability to obtain the global optimal solution of DSA, a new algorithm named chaotic dolphin swarm algorithm (CDSA) is put forward by introducing chaotic map into DSA in this study.

The rest of the paper is organized as follows: Chaotic theory and chaotic map are presented in Section II; The DSA is explained in detail in Section III; Also, the combination of DSA and chaotic map is meticulously described in Section IV. Based on Section II, Section III, and Section IV, Section V gives result and discussion; Last, the conclusions and future work are provided in Section VI.

II. CHAOTIC THEORY AND CHAOTIC MAP

A. CHAOTIC THEORY

Chaos refers to seemingly random irregular motions occurring in deterministic systems. The behavior of a system described by deterministic theory can be expressed as uncertainty, which is the chaotic phenomenon. Furthermore, the theory of studying chaos is called chaos theory. Chaotic systems are called chaotic systems, and chaotic systems are highly sensitive to initial conditions. In other words, for deterministic descriptive systems, chaos can also occur [30].

B. CHAOTIC MAP

To improve the global convergence ability of DSA, chaos which are shown in TABLE 3 [31] is introduced into DSA and CDSA is developed. Then we simulate the distribution and

TABLE 3. Eight different chaotic maps.

No.	Name	Definition
1	Logistic	$x_{i+1}=u\ x_i\ (1+x_i)$
2	Kent	$x_{i+1}=0.9\text{-}1.9\ x_i$
3	Tent	$x_{i+1}= x_i\ /0.4,\ x_i\leq 0.4;\ (1\text{-}x_i)\ /0.6, x_i>0.6$
4	Ushiki	$x_{i+1}= (3.7\text{-}\ x_i\text{-}0.1y_i)\ x_i;\ y_{i+1}= (3.7\text{-}\ 0.15x_i\text{-}y_i)\ y_i$
5	Lozi	$x_{i+1}= 1\text{-}1.75\lvert\ x_i\ \rvert+y_i;\ y_{i+1}=0.3\ x_i$
6	Henon	$x_{i+1}= 1+y_i\text{-}1.4x_i^2;\ y_{i+1}= 0.3x_i$
7	Wien	$dx_i=\text{-}x_i+2.5(y_i\text{-}z_i);\ dy_i=\text{-}x_i+1.5y_i\text{-}2.5z_i;\ dz_i=5u\ (y_i\text{-}1)z_i;\ u=1,y_i\geq 0;u=0,y_i<0$
8	Lorenz	$dx_i=10(y_i\text{-}x_i);\ dy_i=34x_i+x_i\,z_i\ \text{-}y_i;\ dz_i= x_i\,y_i\ \text{-}8/3z_i$

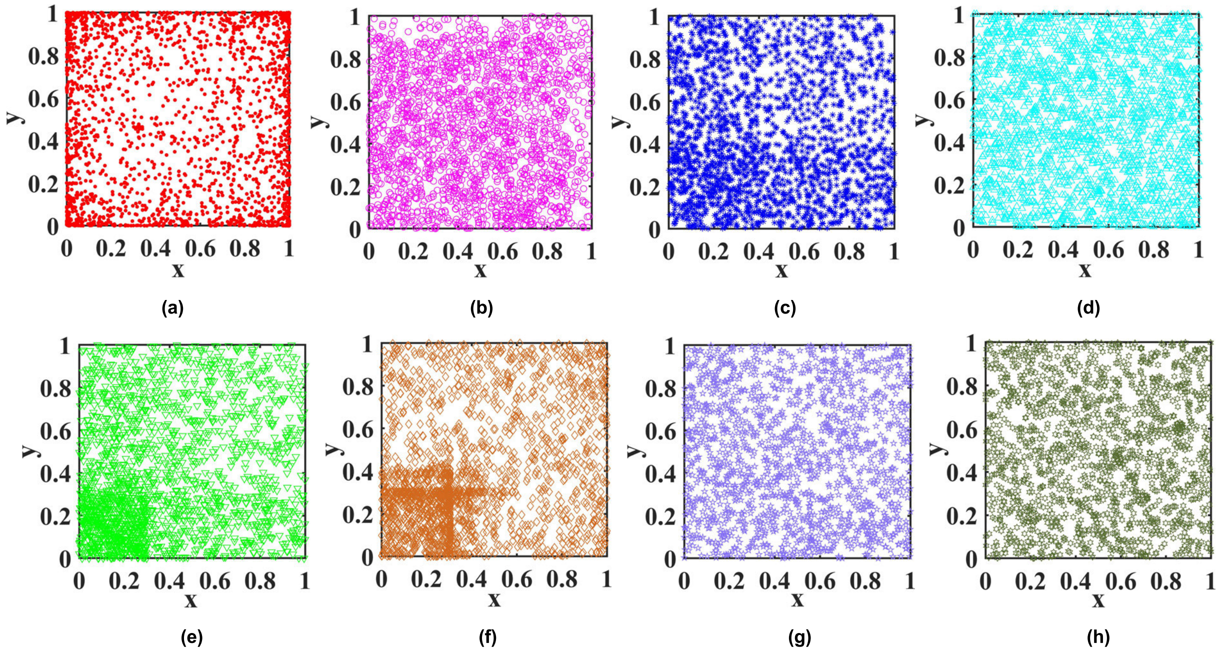

FIGURE 1. Distribution of solutions to eight chaotic maps: (a) Logistic map; (b) Kent map; (c) Tent map; (d) Ushiki map; (e) Lozi map; (f) Henon map; (g) Wien map; (h) Lorenz map.

proportion of solutions of eight chaotic maps in FIGURE 1. and FIGURE 2.

III. DOLPHIN SWARM ALGORITHM (DSA)

A. PREDATORY BEHAVIOR OF DOLPHIN SWARM

In 2016, inspired by PSO, Wu et al. began to pay attention to some behaviors of dolphins ([27]–[29]). For instance, the dolphin uses echolocation in search of prey. Except for echolocation, another behavior of dolphin swarm is cooperation and division of labor to catch prey. The third behavior of dolphin swarm is information exchanges. These three behaviors can be summarized as one dolphin discovers its prey, informs other dolphins by echolocation, and then all dolphins surround the prey and catch food.

B. MAIN DEFINITIONS

1) DOLPHIN

In the process of optimization, each dolphin represents a feasible solution. However, the expression of feasible solutions for various optimization problems is different. In this paper, to better understand the optimization process, dolphins are defined as $\mathbf{Dol}_i = [x_1, x_2, \ldots, x_D]^{\mathrm{T}} (i = 1, 2, \ldots, N)$, where N represents the number of dolphins, and x_j $(j = 1, 2, \ldots, D)$ represent the component:

2) OPTIMAL INDIVIDUAL AND NEIGHBORHOOD SOLUTION

The two variables closely related to the dolphin algorithm are optimal neighborhood solution (expressed as $\boldsymbol{K}$) and optimal individual solution (shown as $\boldsymbol{L}$). More specifically, for every $\mathbf{Dol}_i (i = 1, 2, \ldots, N)$, there are two relevant variables which are $\boldsymbol{L}_i (i = 1, 2, \ldots, N)$ and $\boldsymbol{K}_i (i = 1, 2, \ldots, N)$, respectively, where $\boldsymbol{K}_i$ is the optimal solution of what $\mathbf{Dol}_i$ finds by itself or gets from others, and $\boldsymbol{L}_i$ represents the optimal solution that $\mathbf{Dol}_i$ finds in a single time.

3) DISTANCE AND FITNESS

In DSA, there are three types of distances, which are the distance between $\mathbf{Dol}_i$ and $\mathbf{Dol}_j$, named $\mathrm{DD}_{i,j}$, the distance between $\mathbf{Dol}_i$ and $\boldsymbol{K}_i$, called DK_i, and the distance between $\boldsymbol{L}_i$ and $\boldsymbol{K}_i$, called DLK_i, respectively. The expression of the

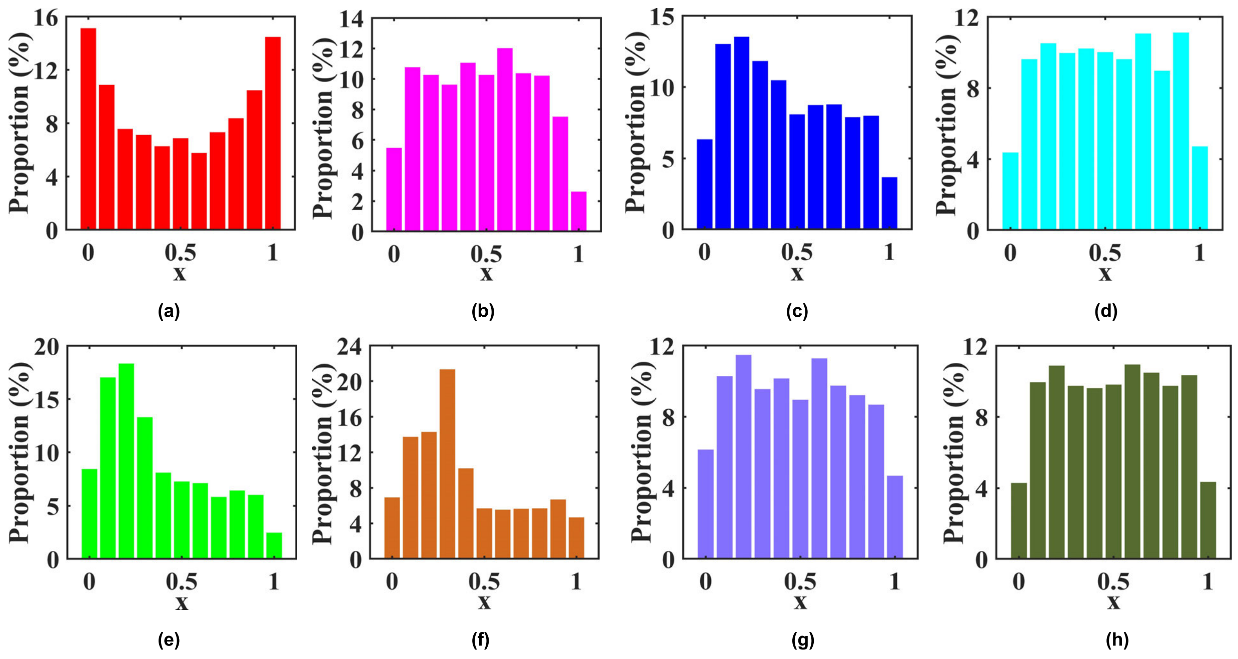

FIGURE 2. Proportion of solutions to eight chaotic maps: (a) Logistic map; (b) Kent map; (c) Tent map; (d) Ushiki map; (e) Lozi map; (f) Henon map; (g) Wien map; (h) Lorenz map.

above three distances is as follows:

$$\mathrm{DD}_{i,j} = \left\|\mathbf{Dol}_i - \mathbf{Dol}_j\right\|, \quad i,j = 1,2,\ldots \mathrm{N},\ i \neq j. \tag{1}$$

$$\mathrm{DK}_i = \|\mathbf{Dol}_i - \boldsymbol{K}_i\|, \quad i = 1,2,\ldots N. \tag{2}$$

$$\mathrm{DKL}_i = \|\boldsymbol{L}_i - \boldsymbol{K}_i\|, \quad i = 1,2,\ldots N. \tag{3}$$

Fitness is based on judging whether the solution is good or bad. In DSA, E is calculated by fitting function (i.e., Rastrigin function in (25) in Section V). For this function, the closer the E value is to 0, the better the solution is obtained. Because different fitting functions of various optimization problems are different, Fitness (X) is used to represent the fitting functions in this paper. Examples of specific fitting functions may be found in (25), (26), (28), and (29) in Section V.

C. CRITICAL STAGES

The DSA is split into six stages, which are the initialization, search, call, reception, predation, and termination stage. Since the initial stage is only the initialization of the population, the final stage only gives a termination condition; therefore, in this subsection, search, call, reception, and predation stage are mainly used. The four stages are described in detail as follows:

1) SEARCH STAGE

When searching for prey, each dolphin usually makes a sound in M directions in the area near the dolphin. In order to qualitatively describe the process of each dolphin's search for prey, sound is defined as $\boldsymbol{V}_i = [v_1, v_2, \ldots, v_D]$ $(i = 1, 2, \ldots, M)$ in this paper, where $v_j(j = 1, 2, \ldots, D)$ represents the component of each dimension, namely the direction attribute of the sound and M represents the number of sounds. Besides, sound must satisfy $||\boldsymbol{V}_i|| = \text{speed}$ $(i = 1, 2, \ldots, M)$, where 'speed' is the speed attribute of sound. To prevent dolphins from falling into the search phase, a maximum search time T_1 is set. In the range of 0 to T_1, the sound $\boldsymbol{V}_j$ that $\mathbf{Dol}_i$ $(i = 1, 2, \ldots, N)$ makes at time t will find a new solution $\boldsymbol{X}_{ijt}$. The definition of $\boldsymbol{X}_{ijt}$ is as follows.

$$\mathrm{X}_{ijt} = \mathbf{Dol}_i + \boldsymbol{V}_j t \tag{4}$$

For $\boldsymbol{X}_{ijt}$ that $\mathbf{Dol}_i$ obtains, its fitness value E_{ijt} is expressed as follows:

$$\mathrm{E}_{ijt} = \mathit{Fitness}\left(\boldsymbol{X}_{ijt}\right). \tag{5}$$

If

$$\begin{aligned} E_{iab} &= \min\nolimits_{j=1,2,\ldots,\mathrm{M};t=1,2,\ldots,T_1} E_{ijt} \\ &= \min\nolimits_{j=1,2,\ldots,\mathrm{M};t=1,2,\ldots,T_1} \mathrm{Fitness}\left(\boldsymbol{X}_{ijt}\right) \end{aligned} \tag{6}$$

Then the optimal individual solution $\boldsymbol{L}_i$ of $\mathbf{Dol}_i$ is defined as

$$\boldsymbol{L}_i = \boldsymbol{X}_{iab} \tag{7}$$

If

$$\mathrm{Fitness}\left(\boldsymbol{L}_i\right) < \mathrm{Fitness}\left(\boldsymbol{K}_i\right) \tag{8}$$

Then $\boldsymbol{K}_i$ is displaced by $\boldsymbol{L}_i$; otherwise, $\boldsymbol{K}_i$ does not change.

After all the $\mathbf{Dol}_i$ $(i = 1, 2, \ldots, N)$ update their $\boldsymbol{L}_i$ and $\boldsymbol{K}_i$, DSA enters call stage.

2) RECEPTION STAGE

In DSA, in order, the reception stage occurs after the call stage, but to better understand the call stage, in this subsection, the reception stage is first described in detail. The quantitative description of information exchange between dolphins and dolphins can be expressed by an $N \times N$-order matrix which is named 'transmission time matrix' (**TS** = ($\mathrm{TS}_{ij}(i = 1, 2, \ldots, N; j = 1, 2, \ldots, N)$)), where TS_{ij} is the rest of the time for the sound of moving from $\mathbf{Dol}_i$ to $\mathbf{Dol}_j$.

When DSA get into the reception stage, that all elements TS_{ij} $(i = 1, 2, \ldots, N; \mathrm{j} = 1, 2, \ldots, N)$ in the **TS** will decrease demonstrate that the sounds spread on any element TS_{ij} in the **TS**, and if

$$\mathrm{TS}_{i,j} = 0 \tag{9}$$

This means that the sound, which will be received by $\mathbf{Dol}_i$, sent from $\mathbf{Dol}_j$ to $\mathbf{Dol}_i$. Next, TS_{ij} will be displaced by a new search time, which is called 'maximum transmission time' (T_2). By this process, we will know the relevant sound has been received. Furthermore, comparing $\boldsymbol{K}_i$ and $\boldsymbol{K}_j$, if

$$Fitness\,(\boldsymbol{K}_i) > Fitness\,(\boldsymbol{K}_j) \tag{10}$$

Then $\boldsymbol{K}_j$ replaces $\boldsymbol{K}_i$, or $\boldsymbol{K}_i$ does not change. Next, DSA gets into the predation stage.

3) CALL STAGE

Based on the search stage, at this stage, each dolphin makes sounds for the sake of informing other dolphins of their search results, containing whether an optimal global solution is found and where it is located. Next, the transmission time matrix **TS** should be updated according to the following inequality.

For $\boldsymbol{K}_i$, $\boldsymbol{K}_j$, and $\mathrm{TS}_{i,j}$, if

$$Fitness\,(\boldsymbol{K}_i) > Fitness\,(\boldsymbol{K}_j) \tag{11}$$

$$\mathrm{TS}_{i,j} > \left\lceil \frac{\mathrm{DD}_{i,j}}{\mathrm{A} \cdot \mathrm{speed}} \right\rceil \tag{12}$$

where A, which is a constant, represents the acceleration. Next, $\mathrm{TS}_{i,j}$ will be updated according to the following equation:

$$\mathrm{TS}_{i,j} = \left\lceil \frac{\mathrm{DD}_{i,j}}{\mathrm{A} \cdot \mathrm{speed}} \right\rceil \tag{13}$$

After all the $\mathrm{TS}_{i,j}$ is updated, DSA gets into the reception stage.

4) PREDATION STAGE

In the search phase, reception stage, call stage and predation stage, predation stage is the most critical and important stage. Next, we describe the predation stage in detail. In this stage, each dolphin preys within a certain surrounding radius, which is defined as R_2. Also, R_2 determine the distance between the dolphin's optimal neighborhood solution and its position after the predation obtains a new position. Furthermore, the search radius R_1, which is the maximum range in the search stage, can be calculated as follows:

$$R_1 = T_1 \times \mathrm{speed} \tag{14}$$

Next, $\mathbf{Dol}_i$ $(i = 1, 2, \ldots, N)$ is regarded as an example to describe the calculation of R_2 and update the dolphin's position.

(a) For $\mathbf{Dol}_i$ $(i = 1, 2, \ldots, N)$, if

$$DK_i \leq R_1 \tag{15}$$

Then, R_2 will be calculated according to (16).

$$R_2 = \left(1 - \frac{2}{e}\right) DK_i, \quad e > 2 \tag{16}$$

where e represents the radius reduction coefficient.

After getting R_2, $\mathbf{Dol}_i$'s new position $\mathbf{newDol}_i$ can be obtained:

$$\boldsymbol{newDol}_i = \boldsymbol{K}_i + \frac{\boldsymbol{Dol}_i - \boldsymbol{K}_i}{\mathrm{DK}_i} R_2. \tag{17}$$

(b) For $\mathbf{Dol}_i$ $(i = 1, 2, \ldots, N)$, if

$$\mathrm{DK}_i > R_1 \tag{18}$$

and

$$\mathrm{DK}_i \geq \mathrm{DKL}_i \tag{19}$$

Then, R_2 will be calculated according to (20).

$$R_2 = \left(1 - \frac{\frac{\mathrm{DK}_i}{\mathrm{Fitness}(\boldsymbol{K}_i)} + \frac{\mathrm{DK}_i - \mathrm{DKL}_i}{\mathrm{Fitness}(\boldsymbol{L}_i)}}{e \cdot DK_i \frac{1}{\mathrm{Fitness}(\boldsymbol{K}_i)}}\right) \mathrm{DK}_i, \quad e > 2 \tag{20}$$

After getting R_2, $\mathbf{Dol}_i$'s new position $\mathbf{newDol}_i$ can be obtained:

$$\mathbf{newDol}_i = \mathbf{K}_i + \frac{\mathbf{Random}}{\|\mathbf{Random}\|} R_2 \tag{21}$$

(c) For $\mathbf{Dol}_i$ $(i = 1, 2, \ldots, N)$, if it satisfies (18) and

$$\mathrm{DK}_i < \mathrm{DKL}_i \tag{22}$$

Then, R_2 will be calculated according to (23).

$$R_2 = \left(1 - \frac{\frac{\mathrm{DK}_i}{\mathrm{Fitness}(\boldsymbol{K}_i)} + \frac{\mathrm{DKL}_i - \mathrm{DK}_i}{\mathrm{Fitness}(\boldsymbol{L}_i)}}{e \cdot DK_i \frac{1}{\mathrm{Fitness}(\boldsymbol{K}_i)}}\right) \mathrm{DK}_i, \quad e > 2 \tag{23}$$

After getting R_2, $\mathbf{Dol}_i$'s new position $\mathbf{newDol}_i$ can be obtained by (21).

After $\mathbf{Dol}_i$ moves to the position $\mathbf{newDol}_i$, comparing $\mathbf{newDol}_i$ with $\boldsymbol{K}_i$ in terms of fitness, if

$$\mathrm{Fitness}\,(\boldsymbol{newDol}_i) > \mathrm{Fitness}\,(\boldsymbol{K}_j) \tag{24}$$

Then $\mathbf{newDol}_i$ replaces $\boldsymbol{K}_i$, or $\boldsymbol{K}_i$ does not change.

Finally, if the iterative termination condition is satisfied, DSA enters the termination stage, or, DSA enters the search stage again.

IV. CHAOTIC DOLPHIN SWARM ALGORITHM (CDSA)

The algorithm searches the optimal solution of the target problem by simulating dolphin behavior. Each iteration updates all individual dolphins, and selects the current optimal position, repeating the process until the end condition is satisfied.

The flow chart of CDSA is shown in FIGURE 3.

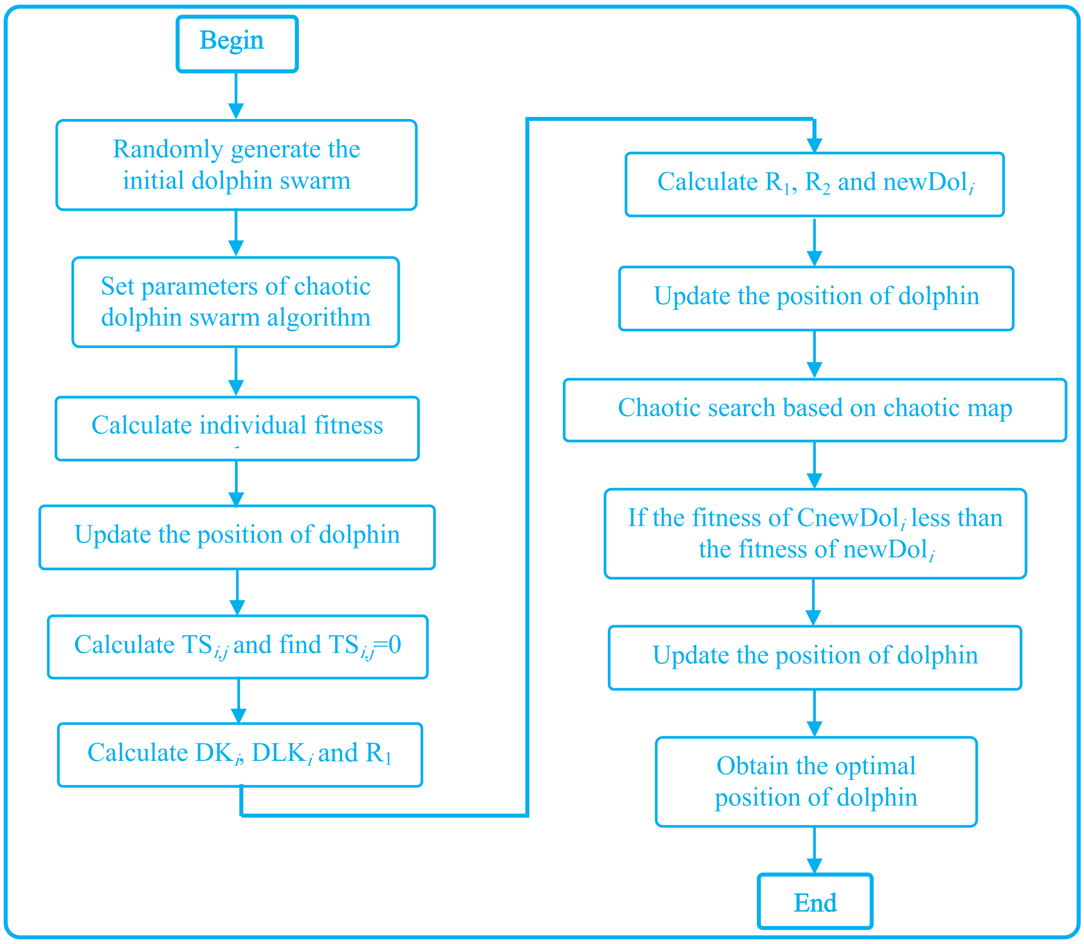
Begin
Randomly generate the initial dolphin swarm
Set parameters of chaotic dolphin swarm algorithm
Calculate individual fitness
Update the position of dolphin
Calculate TS$_{i,j}$ and find TS$_{i,j}$=0
Calculate DK$_i$, DLK$_i$ and R$_1$
Calculate R$_1$, R$_2$ and newDol$_i$
Update the position of dolphin
Chaotic search based on chaotic map
If the fitness of CnewDol$_i$ less than the fitness of newDol$_i$
Update the position of dolphin
Obtain the optimal position of dolphin
End

FIGURE 3. The flow chart of CDSA.

V. RESULT AND DISCUSSION

In this section, to verify the performance of the proposed CDSA, two experiments are performed. Experiment I compare the performance of the combination of different chaotic maps and DSA by high-dimensional function test; Experiment II compare CDSA with advanced algorithms according to performance indexes by high-dimensional function test.

A. EXPERIMENTAL PLTFORM

For different computers, the results of Experiment I and II are greatly affected. Therefore, to compare fairly, Experiment I and experiment II are carried out on the same experimental computer. Computer configuration consist of the hardware configuration (i.e. CPU: Intel(R) Core (TM) I7-8550U; Frequency: 1.99 GHz; RAM: 16.0GB (15.9 GB Available); Hard drive: 1TB) and software configuration (i.e. Operating system: Windows 10; Language edition: MATLAB R2018a).

B. HIGH-DIMENSIONAL TEST FUNCTIONS

In this study, Rastrigin function is used to test the performance of different chaotic maps combined with DSA. Furthermore, we use Levy function, Rotated hyper-ellipsoid function, and Sum squares function to test the performance of CDSA and advanced meta-heuristic algorithms (e.g., whale optimization algorithm (WOA)). The definition of Rastrigin function, Levy function, Rotated hyper-ellipsoid function, and Sum squares function is shown in (25), (26), (28), and (29). Also, The Rastrigin function whose large-scale search interval and many local minimum results in the difficulty of finding a global minimum is a typical nonlinear multimodal function. The Rotated Hyper-Ellipsoid function, which is an extension of the axis parallel Hyper-Ellipsoid function, also referred to as the Sum Squares function is continuous, convex, and unimodal. The Sum Squares function, which is continuous, convex, and unimodal, also referred to as the Axis Parallel Hyper-Ellipsoid function, has no local minimum except the global one. Rastrigin function, Levy function, Rotated hyper-ellipsoid function and Sum squares function Rastrigin function, Levy function, Rotated hyper-ellipsoid function, and Sum squares function are shown in FIGURE 4 in their two-dimensional form.

The definition of Rastrigin function is as follow:

$$f_1(\boldsymbol{x}) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right] \tag{25}$$

where x_i and i belong to [−5, 5] and [1, D], the minimum value of $f_1(\boldsymbol{x})$ is 0.

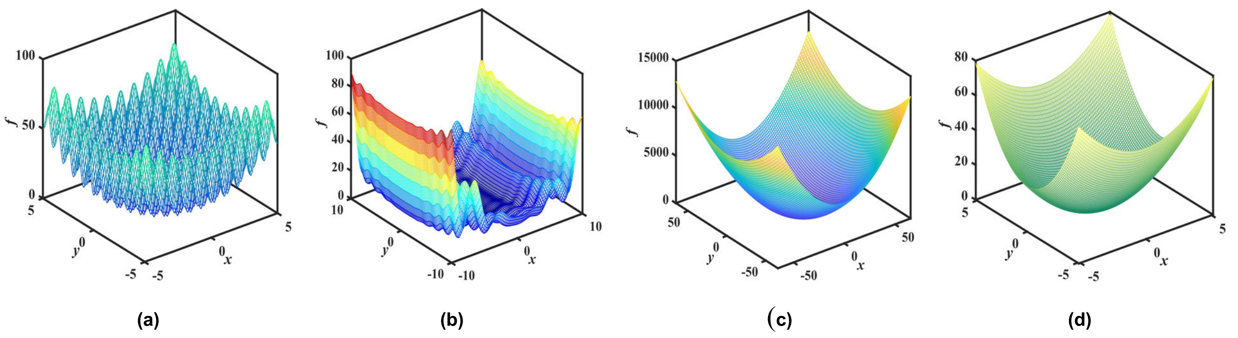

FIGURE 4. Four functions are used to test in their two-dimensional form: (a) Rastrigin function; (b) Levy function; (c) Rotated hyper-ellipsoid function; (d) Sum squares function.

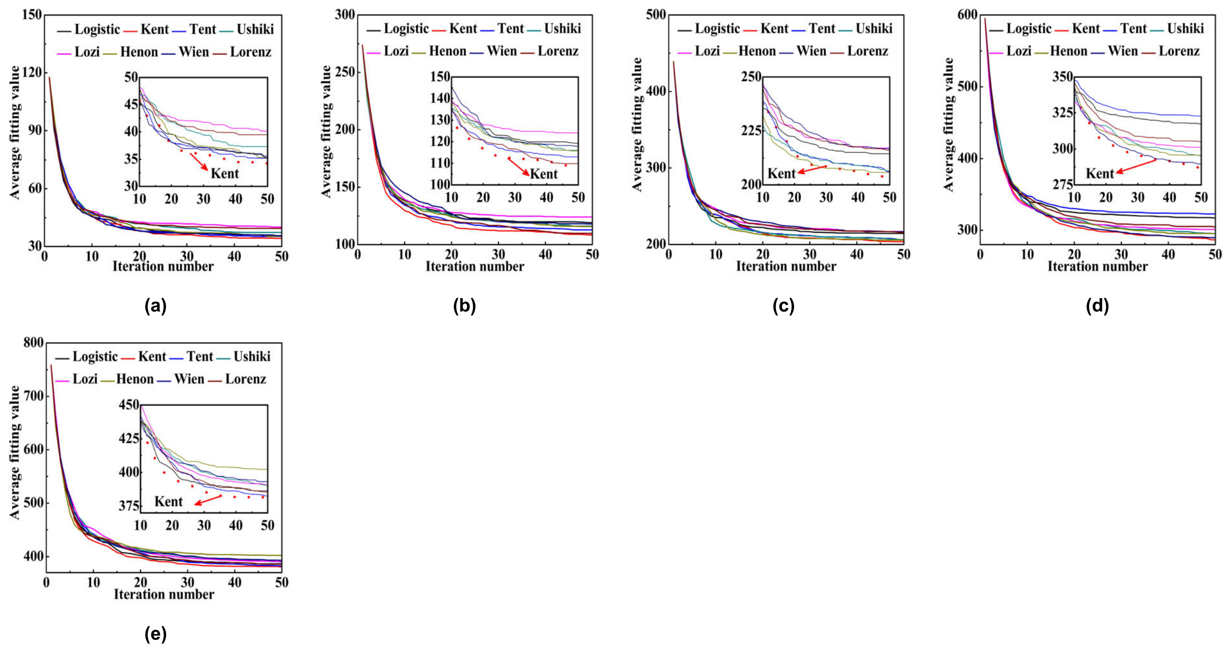

FIGURE 5. Average fitting value curve of the different dimension of the Rastrigin function: (a) 10; (b) 20; (c) 30; (d) 40; (e) 50.

The definition of Levy function is as follow:

$$f_2(\boldsymbol{x}) = \sin^2(\pi\omega_1) + \sum_{i=1}^{D-1}(\omega_i - 1)^2 \ldots$$
$$\ldots\left[1 + 10\sin^2(\pi\omega_i + 1)\right]$$
$$+ (\omega_D - 1)^2\left[1 + \sin^2(2\pi\omega_D)\right] \quad (26)$$

where i belong to [1, D] and the minimum value of $f_1(\boldsymbol{x})$ is 0. ω_i is defined as follow:

$$\omega_i = 1 + \frac{x_i - 1}{4} \quad (27)$$

The definition of Rotated hyper-ellipsoid function is as follow:

$$f_3(\boldsymbol{x}) = \sum_{i=1}^{D}\sum_{j=1}^{i} x_j^2 \quad (28)$$

where x_i and i belong to [−65.536, 65.536] and [1, D] and the minimum value of $f_1(\boldsymbol{x})$ is 0.

The definition of Sum squares function is as follow:

$$f_4(\boldsymbol{x}) = \sum_{i=1}^{D} i x_i^2 \quad (29)$$

where x_i and i belong to [−5.12, 5.12] and [1, D] and the minimum value of $f_1(\boldsymbol{x})$ is 0.

C. COMPARING CDSA WITH LITERATURE

To verify the effectiveness and performance of the proposed CDSA, some advanced evolutionary algorithms including WOA [32], DSA ([27]–[29]), AGA [33], APSO [5], WPA [26], CS [34] and CSO [35] are used to compare in this paper.

D. EXPERIMENT I: COMPARISON OF DIFFERENT CHAOTIC MAPS

In this experiment, in order to determine the optimal chaotic map, Rastrigin function whose dimensions are set to 10, 20, 30, 40 and 50 respectively are used as the test function.

TABLE 4. Statistical results of the different chaotic maps based on the different dimensional rastrigin function.

Dimensional number	Indexes	Logistic	Kent	Tent	Ushiki	Lozi	Henon	Wien	Lorenz
10	Best	21.370	**16.471**	21.169	16.901	21.785	23.295	19.492	20.341
	Worst	56.741	**51.556**	52.336	64.112	73.965	56.535	60.952	66.124
	Mean	35.327	**34.223**	35.212	37.210	40.071	35.137	35.589	39.451
	A. G	31.050	**25.700**	26.350	25.700	27.550	35.000	28.000	24.850
20	Best	64.516	**62.697**	87.249	79.439	77.952	83.134	71.972	81.112
	Worst	177.376	**136.132**	143.592	162.790	180.876	154.323	152.121	144.625
	Mean	119.100	**108.096**	112.954	116.018	124.078	115.709	117.401	109.809
	A. G	27.950	**22.600**	24.000	28.950	23.300	30.850	28.200	31.750
30	Best	149.771	**138.021**	156.211	147.331	161.163	149.676	170.583	154.984
	Worst	279.462	**270.781**	299.457	284.894	265.984	254.016	257.535	272.307
	Mean	214.430	**202.748**	205.834	206.318	216.625	205.291	215.974	215.943
	A. G	26.100	**22.550**	30.350	29.400	24.650	32.350	32.250	30.250
40	Best	239.520	**211.599**	271.526	215.969	228.050	228.750	235.316	231.631
	Worst	380.276	**349.725**	411.209	365.320	348.587	368.242	403.087	393.663
	Mean	316.709	**295.179**	322.767	286.745	300.269	295.642	289.390	304.439
	A. G	31.850	**25.700**	27.200	37.100	27.350	29.550	32.800	33.500
50	Best	311.354	**268.763**	294.777	336.414	296.818	326.537	303.674	324.927
	Worst	502.021	**432.184**	452.259	448.962	505.976	521.778	470.726	467.211
	Mean	386.032	**382.186**	380.098	389.484	391.008	402.243	393.178	385.089
	A. G	32.350	**29.200**	33.200	39.800	30.350	30.200	30.900	32.900

Note: The values in bold indicate a minimum value in each evaluation index, including St. d, Mean, Worst, Best, A. G.

TABLE 5. The efficiency of different chaotic maps.

Dimensional number	Indexes	Logistic	Kent	Tent	Ushiki	Lozi	Henon	Wien	Lorenz
10	Runtime	0.484	0.469	0.534	0.501	0.514	0.487	0.522	0.497
20		0.484	0.469	0.534	0.501	0.514	0.487	0.522	0.497
30		0.772	0.761	0.806	0.791	0.783	0.776	0.799	0.776
40		0.772	0.761	0.806	0.791	0.783	0.776	0.799	0.776
50		0.924	0.803	0.963	0.875	0.908	0.888	0.913	0.916

Note: The values in bold indicate minimum solution time.

Parameter settings of CDSA are P = 50, Speed = 1, T = 10, A = 1, e = 3, k = 10, u = 1. Average fitting value curve of the different dimension of the Rastrigin function are shown in FIGURE 5. TABLE 4 gives the index values of various CDSA in detail.

As can be seen from FIGURE 5 and TABLE 4, Kent map is the closest to the actual optimal solution of Rastrigin function, and the average fitness value of Kent map is lower than that of other maps. This shows that the combination of Kent map and DSA is better than that of different maps and DSA. Detailed comparisons are as follows:

(1) FIGURE 5 (a) - (e) shows that the average fitting value curve of Kent map is lower than that of other maps, which shows that the convergence speed of the combination of Kent mapping and Dolphin swarm algorithm is faster than that of different maps and DSA.

(2) From TABLE 4, we can see that the four indexes of Kent map are lower than those of different maps, and the Best index of Kent map is lower than that of different maps. It shows that the gap between the optimal solution obtained by Kent map and the actual optimal solution of Rastrigin function is small, and the Worst index of Kent map is lower than that of other maps. It shows that the worst solution obtained by Kent map is the worst one compared with the actual best solution of Rastrigin function. The result shows that the difference between the average solution obtained by Kent map and the actual optimal solution of Rastrigin function is small. The three indexes above show that the solution accuracy of the Kent map is higher than that of other maps. The A. G index of Kent map is lower than that of other maps, which indicates that the solving speed of Kent map is faster than that of different maps.

From TABLE 5, we can see that the runtime of the combination of Kent map and DSA is lower than that of other maps

TABLE 6. Parameter setting.

Algorithm	Parameter
CDSA	P=50, Speed=2, T=20, A=2, e=4, k=20, u=2
WOA	P=50, c1=2, c2=2, wmax=1.4, wmin=0.1
DSA	P=50, Speed=2, T=20, A=2, e=4
AGA	P=50, p1_max=0.8, p1_min=0.2, p2_max=0.2, p2_min=0.02
APSOA	P=50, k1=2, k2=2, w1x=1.4, w2=0.1
WPA	P=50, p=0.04, t=1, step=4, d=4, p_c=0.4;
CS	P=50, a=2, σ=0.7066, β=3, w=4
CSO	P=50, R_c=0.2, R_h =0.7, R_{ch}=0.1;

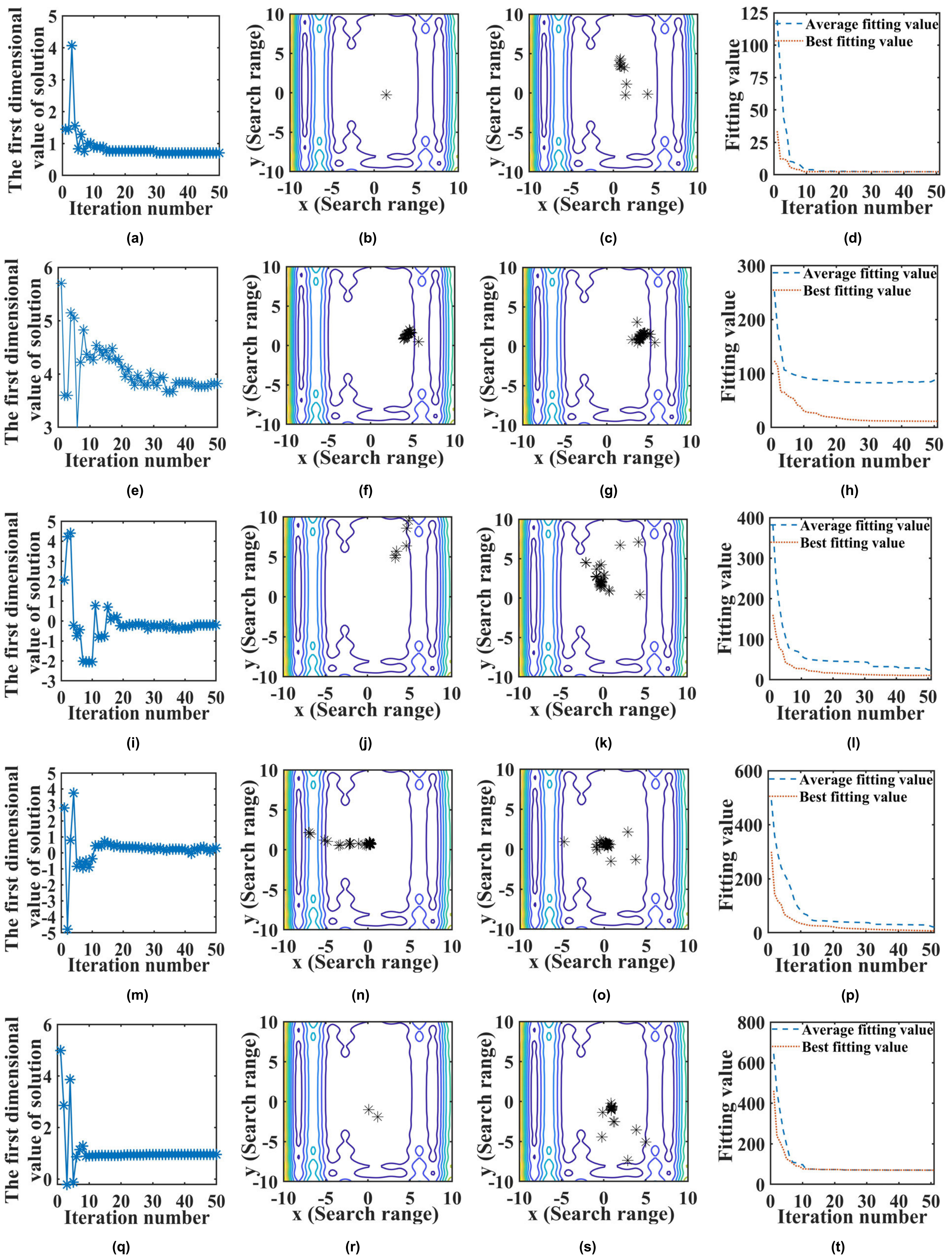

FIGURE 6. Convergence behavior based on Levy function with different dimensions by using CDSA: the first dimensional value of solution of 10, 20, 30, 40 and 50-dimension including (a), (e), (i), (m) and (q); The first individual search path in population of 10, 20, 30, 40 and 50-dimension including (b), (f), (j), (m) and (r); The optimal individual search path in population of 10, 20, 30, 40 and 50-dimension including (c), (g), (k), (n) and (s); The fitness curve of 10, 20, 30, 40 and 50-dimension including (d), (h), (p), (o) and (t).

TABLE 7. Quantitative performance comparison based on the high-dimensional Levy function.

Dimensional number	Indexes	CDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
10	Best	**0.069**	0.338	0.400	0.680	3.511	6.454	3.896	0.213
	Worst	**1.663**	3.335	12.279	19.306	20.358	15.105	9.166	11.769
	Mean	**0.454**	1.402	2.757	7.263	8.171	10.286	5.518	2.894
	A. G	**30.800**	44.900	43.150	38.550	39.900	46.750	31.350	38.300
20	Best	**1.457**	1.989	2.609	15.466	36.533	13.860	4.119	2.539
	Worst	**6.701**	39.906	44.329	56.853	106.873	39.113	18.713	16.713
	Mean	**2.961**	10.011	16.390	31.594	70.650	28.487	9.489	8.489
	A. G	**32.850**	44.700	40.500	41.650	36.850	48.650	44.900	50.850
30	Best	**2.300**	10.012	6.113	36.734	103.335	33.464	40.781	4.406
	Worst	**10.142**	39.789	53.533	109.153	233.593	66.139	121.926	24.647
	Mean	**5.179**	20.831	17.515	62.969	156.069	54.481	91.114	13.024
	A. G	**28.650**	44.650	44.750	44.950	30.600	50.150	31.100	46.750
40	Best	**6.167**	13.394	9.434	93.097	212.287	84.509	118.458	7.131
	Worst	**20.822**	70.882	47.995	190.888	332.367	110.623	219.866	75.474
	Mean	**11.772**	41.520	24.378	125.668	279.602	99.045	156.498	27.658
	A. G	**23.900**	44.900	46.650	45.700	37.000	50.500	26.350	49.250
50	Best	**15.387**	25.756	16.157	137.046	305.993	113.017	191.819	18.347
	Worst	**42.109**	106.201	76.085	251.568	456.791	167.453	297.483	70.024
	Mean	**26.180**	56.548	35.983	188.505	379.372	140.831	238.354	35.696
	A. G	**26.900**	43.750	49.350	44.650	30.050	50.550	30.850	48.050
Average	Best	**5.076**	10.298	6.943	56.605	132.332	50.261	71.815	6.527
	Worst	**16.287**	52.023	46.844	125.554	229.996	79.687	133.431	39.725
	Mean	**9.309**	26.062	19.405	83.200	178.773	66.626	100.195	17.552
	A. G	**28.620**	44.580	44.880	43.100	34.880	49.320	32.910	46.640

Note: The values in bold indicate an average minimum value in each evaluation index, including St. d, Mean, Worst, Best, A. G.

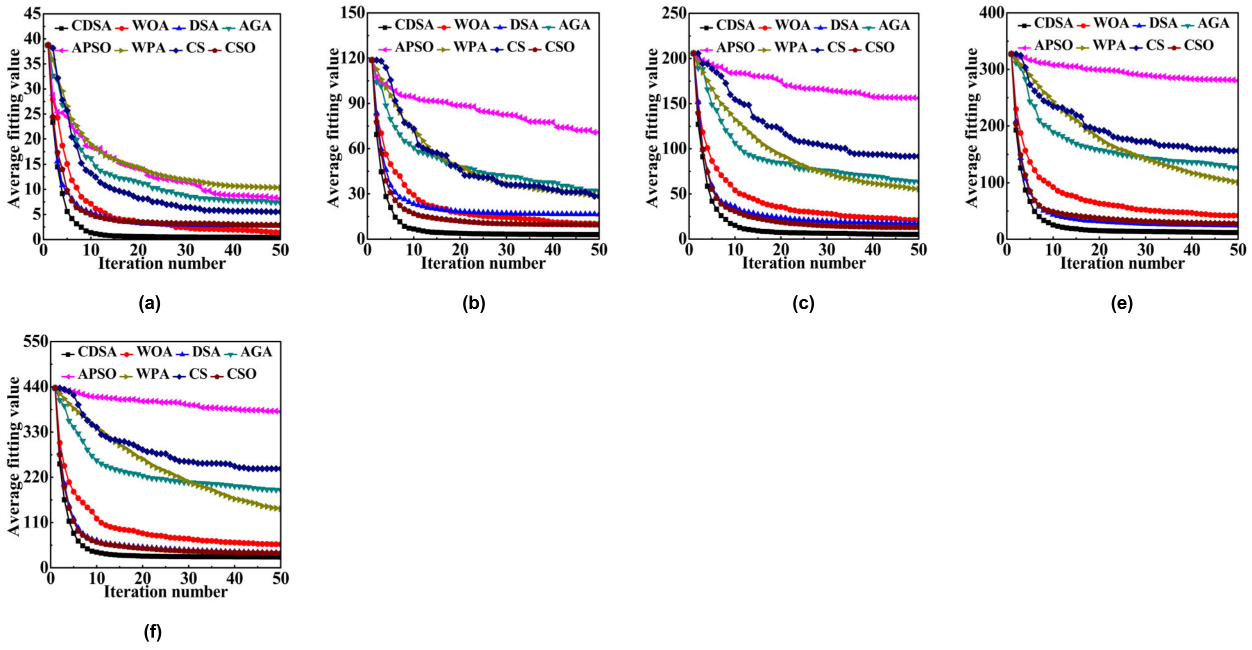

FIGURE 7. Average fitting curve based on Levy function with different dimensions: (a) 10; (b) 20; (c) 30; (d) 40; (e) 50.

and DSA, which shows that the combination of Kent map and DSA is the most efficient.

E. EXPERIMENT II: COMPARISON WITH STATE-OF-THE ART ALGORITHMS

To compare the performance of the proposed CDSA, based on the optimal chaotic map (Kent map) determined in Experiment I, CDSA is compared with the state-of-the-art algorithms (shown in Section V) in this study.

1) COMPARISON OF DIFFERENT ALGORITHMS BASED ON HIGH-DIMENSIONAL LEVY FUNCTION

In this subsection, Levy function whose dimensions are set to 10, 20, 30, 40, and 50 respectively are used as the

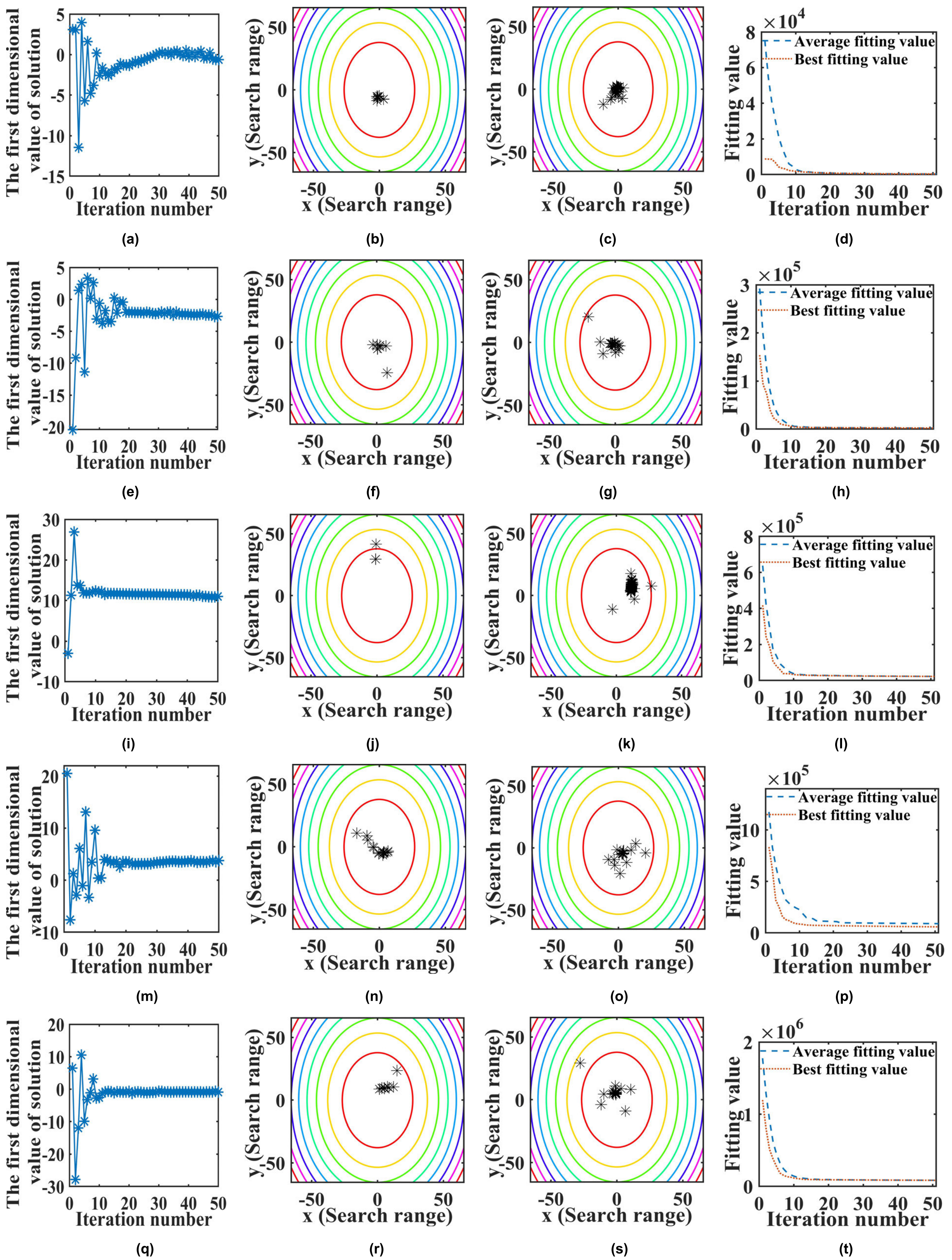

FIGURE 8. Convergence behavior based on rotated hyper-ellipsoid function with different dimensions by using CDSA: the first dimensional value of solution of 10, 20, 30, 40 and 50-dimension including (a), (e), (i), (m) and (q); The first individual search path in population of 10, 20, 30, 40 and 50-dimension including (b), (f), (j), (m) and (r); The optimal individual search path in population of 10, 20, 30, 40 and 50-dimension including (c), (g), (k), (n) and (s); The fitness curve of 10, 20, 30, 40 and 50-dimension including (d), (h), (p), (o) and (t).

TABLE 8. Quantitative performance comparison based on the high-dimensional hyper-ellipsoid function.

Dimensional number	Indexes	CDSA	WOA	DSA	AGA	APSO	WPA	567.779	CSO
10	Best	**5.811**	235.777	13.179	2088.753	1583.201	4158.622	3889.536	6.975
	Worst	**308.067**	2900.175	1495.788	11612.054	10051.500	19111.249	1674.901	3276.110
	Mean	**71.059**	987.162	504.251	7120.061	5160.157	8910.939	39.600	736.497
	A. G	**36.650**	46.800	50.800	38.700	40.800	51.000	8005.596	50.450
20	Best	**484.805**	5120.736	2447.287	17071.921	39776.303	45255.113	32306.518	1558.323
	Worst	**2874.999**	16292.552	24052.520	102722.435	127384.656	139249.417	16591.695	13884.405
	Mean	**1411.862**	9945.038	7017.109	54110.207	68704.433	90595.144	39.150	7454.761
	A. G	**37.650**	46.250	50.950	41.050	39.300	51.000	33479.807	50.950
30	Best	**3073.139**	19388.884	8163.161	82584.959	167560.857	158862.128	68852.267	12363.923
	Worst	**14942.920**	52474.088	34507.126	251284.715	358709.227	309705.456	51356.510	40612.953
	Mean	**7111.707**	36974.586	23505.150	165728.166	276320.122	241034.325	35.100	24105.924
	A. G	**36.450**	48.350	51.000	44.350	38.600	51.000	64234.296	50.600
40	Best	**7948.224**	45766.667	30837.453	194643.195	507813.677	380081.574	189149.454	22376.042
	Worst	**39711.267**	94728.831	72786.316	711224.220	784064.500	715851.013	108506.429	77272.316
	Mean	**19194.401**	68948.564	44423.753	382081.682	616765.534	517281.473	34.500	49216.117
	A. G	**32.800**	47.000	51.000	43.050	31.600	51.000	150230.818	51.000
50	Best	**19579.489**	91758.714	44311.753	391889.583	851117.310	637271.212	296615.263	56312.856
	Worst	**57618.403**	220616.782	156247.215	987056.188	1239475.799	1156832.825	210287.043	171382.021
	Mean	**35364.808**	143616.440	89024.195	654738.281	1040906.735	941676.471	35.150	92048.979
	A. G	**26.650**	47.500	51.000	44.600	30.400	51.000	51303.659	51.000
Average	Best	**6218.294**	32454.156	17154.567	137655.682	313570.270	245125.730	118162.608	18523.624
	Worst	**23091.131**	77402.486	57817.793	412779.922	503937.136	468149.992	77683.316	61285.561
	Mean	**12630.767**	52094.358	32894.892	252755.679	401571.396	359899.670	36.700	34712.456
	A. G	**34.040**	47.180	50.950	42.350	36.140	51.000	567.779	50.800

Note: The values in bold indicate an average minimum value in each evaluation index, including St. d, Mean, Worst, Best, A.

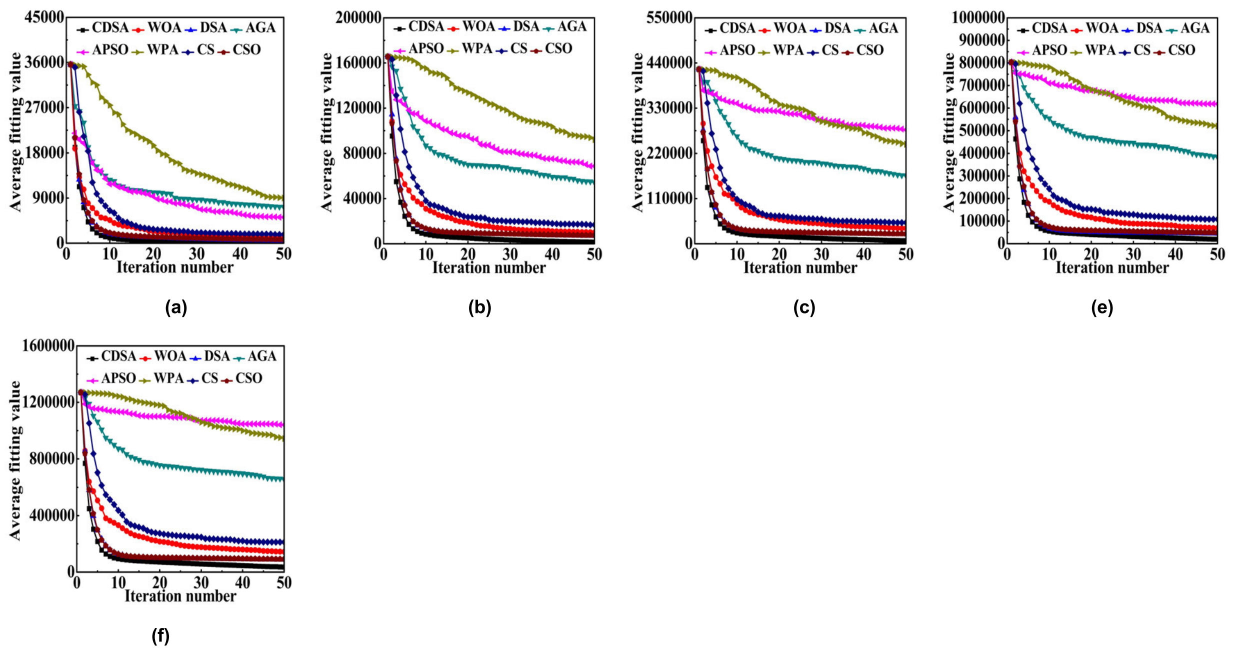

FIGURE 9. Average fitting curve based on rotated hyper-ellipsoid function with different dimensions: (a) 10; (b) 20; (c) 30; (d) 40; (e) 50.

test function. Because the results of each experiment are different, to ensure the fairness of the comparison, ten experiments are carried out, and the average results of 10 experiments are compared. Also, the parameter settings of CDSA and the metaheuristic algorithm, including WOA, DSA, AGA, APSO, WPA, CS, and CSO considered for comparison, are listed in TABLE 6 in detail. Convergence behavior based on Levy function with different dimensions by using CDSA is shown in FIGURE 6, and Average fitting curve based on Levy function with different dimensions is shown in FIGURE 7. Moreover, TABLE 7 presents quantitative performance comparison based on the high-dimensional Levy function.

To analyze the convergence behavior of CDSA and expand the search space, the first-dimensional value of the solution,

TABLE 9. **Quantitative performance comparison based on the high-dimensional Sum squares function.**

Dimensional number	Indexe s	CDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
10	Best	**0.199**	0.554	0.938	10.623	15.222	0.240	20.337	0.538
	Worst	**4.058**	13.427	5.627	136.997	58.100	19.273	92.417	4.138
	Mean	**1.325**	5.102	2.706	46.929	35.018	4.627	54.679	2.147
	A. G	**30.500**	45.700	35.850	33.800	39.300	50.250	32.800	43.350
20	Best	**9.986**	20.527	14.704	149.150	261.872	57.522	351.689	10.441
	Worst	**34.553**	97.301	63.391	501.041	615.266	264.750	699.699	65.360
	Mean	**23.629**	51.017	27.436	307.204	422.423	163.085	509.774	28.834
	A. G	**25.300**	47.200	44.800	42.450	38.600	51.000	36.950	43.400
30	Best	**35.117**	115.865	49.029	654.066	1255.730	343.077	1274.301	53.801
	Worst	**136.595**	322.200	130.519	1467.754	2181.583	889.263	2035.205	177.249
	Mean	**79.341**	200.017	87.000	1002.932	1704.137	585.714	1660.237	97.536
	A. G	**32.750**	46.300	43.700	42.800	37.800	51.000	34.500	45.300
40	Best	**102.646**	301.610	74.821	1734.372	2846.256	936.175	2169.119	114.252
	Worst	**350.677**	790.458	380.532	3142.263	4565.461	2594.261	3661.817	376.796
	Mean	**214.015**	512.153	219.202	2286.969	3857.952	1510.448	3051.329	237.316
	A. G	**28.450**	48.400	47.100	44.700	34.700	51.000	29.200	50.900
50	Best	**192.572**	415.945	205.887	2526.755	5841.006	2020.272	4872.404	254.445
	Worst	**596.518**	1125.506	694.458	5485.283	8540.137	4470.176	6531.780	605.711
	Mean	**370.978**	794.316	374.854	4114.411	7086.943	3085.481	5671.120	376.771
	A. G	**26.400**	49.100	46.250	45.150	31.600	51.000	28.900	50.750
Average	Best	**68.104**	170.900	69.076	1014.993	2044.017	671.457	1737.570	86.695
	Worst	**224.480**	469.778	254.905	2146.668	3192.109	1647.545	2604.184	245.851
	Mean	**137.858**	312.521	142.240	1551.689	2621.295	1069.871	2189.428	148.521
	A. G	**28.680**	47.340	43.540	41.780	36.400	50.850	32.470	46.740

Note: The values in bold indicate an average minimum value in each evaluation index, including St. d, Mean, Worst, Best, A.

the first individual search path in population, the optimal individual search path in population and the fitness curve of Levy function with different dimensions is observed respectively. The specific process of the convergence behavior of CDSA is shown in FIGURE 6. It can be seen from FIGURE 6 that with the gradual increase of Levy function, the first solution of an individual fluctuates first and then approaches to 1. Also, with the gradual increase of the dimension of Levy function, the range of values of average fitting value and best fitting value increases gradually. So, the results verify the performance of CDSA in solving high-dimensional Levy function.

It can be seen from FIGURE 7 and TABLE 7:

(1) As can be seen from FIGURE 7, the average fitting value curve of CDSA is lower than that of WOA, DSA, AGA, APSO, WPA, CS, and CSO. Also, the average fitting value curve of APSO algorithm is higher than that of CDSA, WOA, AGA, WPA, CS, and CSO. These comparisons show that CDSA is better than the algorithm considered for comparison.

(2) It can be concluded from TABLE 7 that for different dimensions of Levy function, the average Best, Worst, Mean and A.G of CDSA are 5.222, 1.867, 51.529, 127.256, 45.185, 66.739, 1.451, 35.736, 30.557, 109.267, 213.709, 63.400, 117.144, 23.438, 16.753, 10.096, 73.891, 169.464, 57.317, 90.886, 8.243 and 15.960, 16.260, 14.480, 6.260, 20.700, 4.290, 18.020 less than WOA, DSA, AGA, APSO, WPA, CS and CSO.

In summary, CDSA is better than WOA, DSA, AGA, APSO, WPA, CS, and CSO for high-dimensional Levy function.

2) COMPARISON OF DIFFERENT ALGORITHMS BASED ON HIGH-DIMENSIONAL ROTATED HYPER-ELLIPSOID FUNCTION

In this subsection, Rotated Hyper-Ellipsoid function whose dimensions are set to 10, 20, 30, 40, and 50 respectively are used as the test function. Also, the number of experiments is 10. Also, the parameters of CDSA, WOA, DSA, AGA, APSO, WPA, CS, and CSO are specified in TABLE 6. Convergence behavior based on Rotated Hyper-Ellipsoid function with different dimensions by using CDSA is shown in FIGURE 8, and Average fitting curve based on Rotated Hyper-Ellipsoid function with different dimensions is shown in FIGURE 9. Moreover, TABLE 8 presents quantitative performance comparison based on the high-dimensional Rotated Hyper-Ellipsoid function.

It can be seen from FIGURES 8-9, and TABLE 8:

(1) As can be seen from FIGURE 8, with the gradual increase of the dimension of Rotated Hyper-Ellipsoid, the value range of average fitting value and best fitting value increases gradually. This verifies the characteristics of CDSA in solving high-dimensional Rotated Hyper-Ellipsoid function.

(2) Similar to FIGURE 7, the average convergence curve of the CDSA is lower than that of the comparison algorithm, so the CDSA algorithm is better than the comparison algorithm.

(3) It can be concluded from TABLE 8 that for different dimensions of Rotated Hyper-Ellipsoid function, the average Best, Worst, Mean and A.G of CDSA are 26235.862, 10936.273, 131437.388, 307351.976, 238907.436, 45085.365, 12305.330, 54311.355, 34726.662,

389688.791, 480846.005, 445058.861, 95071.477, 38194.430, 39463.591, 20264.125, 240124.912, 388940.629, 347268.903, 65052.549, 22081.689 and 13.140, 16.910, 8.310, 2.100, 16.960, 2.660, 16.760 less than WOA, DSA, AGA, APSO, WPA, CS and CSO.

In a word, CDSA is superior to WOA, DSA, AGA, APSO, WPA, CS, and CSO in terms of efficiency and accuracy for Rotated Hyper-Ellipsoid function with different dimensions. At the same time, CDSA has better ability to obtain globally optimal solutions than WOA, DSA, AGA, APSO, WPA, CS, and CSO.

3) COMPARISON OF DIFFERENT ALGORITHMS BASED ON HIGH-DIMENSIONAL SUM SQUARES FUNCTION

In this subsection, Sum Squares function whose dimensions are set to 10, 20, 30, 40, and 50 respectively are used as the test function. The number of experiments is the same as Section 1) and Section 2), and the parameters of different algorithms are specified in TABLE 6. Moreover, TABLE 9 presents a quantitative performance comparison based on the high-dimensional Sum Squares function.

It can be concluded from TABLE 9 that for different dimensions of Sum Squares function, the average Best, Worst, Mean and A.G of CDSA are 102.796, 0.972, 946.889, 1975.913, 603.353, 1669.466, 18.591, 245.298, 30.425, 1922.188, 2967.629, 1423.065, 2379.704, 21.371, 174.663, 4.382, 1413.831, 2483.437, 932.013, 2051.570, 10.663 and 18.660, 14.860, 13.100, 7.720, 22.170, 3.790, 18.060 less than WOA, DSA, AGA, APSO, WPA, CS and CSO.

VI. CONSLUSIONS AND FUTURE WORK

Aiming at the problem that DSA has weak global convergence ability and is easy to fall into local optimum, this paper introduces a chaotic map into DSA and successfully proposes CDSA. Based on Rastrigin function, the optimal chaotic map is determined. To verify the performance of CDSA, we compared it with WOA, DSA, AGA, APSO, WPA, CS, and CSO based on high-dimensional Levy function, Rotated Hyper-Ellipsoid function and Sum Squares function. Detailed conclusions are as follows:

(1) For Rastrigin function with different dimensions, Kent map is better than Logistic, Tent, Ushiki, Lozi, Henon, Wien, and Lorenz map according to Best, Worst, Mean, and A.G.

(2) Based on the optimal chaotic map (Kent map), CDSA is lower than WOA, DSA, AGA, APSO, WPA, CS and CSO for high-dimensional Levy function, Rotated Hyper-Ellipsoid function, Sum Squares function in terms of Best, Worst, Mean and A.G. This indicates that CDSA is superior to WOA, DSA, AGA, APSO, WPA, CSO and CSO and shows that CDSA has strong global convergence ability and does not fall into local optimum.

This paper is based on high-dimensional Rastrigin function, Levy function, Rotated Hyper-Ellipsoid function, Sum Squares function whose solving dimension is 50. To further improve the ability of CDSA to solve higher dimensions, CDSA needs to be improved. In addition, CDSA needs to be applied to more areas of hyperparametric optimization [36]–[40].

REFERENCES

[1] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Inf. Sci.*, vol. 295, pp. 407–428, Feb. 2015.

[2] Y. Wu, B. Yan, and X. Qu, "Improved chicken swarm optimization method for reentry trajectory optimization," *Math. Problems Eng.*, vol. 2018, Jan. 2018, Art. no. 8135274.

[3] P. Andras, "High-dimensional function approximation with neural networks for large volumes of data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 500–508, Feb. 2018.

[4] L. Xiaohu, Z. Jinhua, W. Sunan, L. Maolin, and L. Kunpeng, "A small world algorithm for high-dimensional function optimization," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat.*, Dec. 2009, pp. 55–59.

[5] Y.-J. Wang, "Improving particle swarm optimization performance with local search for high-dimensional function optimization," *Optim. Methods Softw.*, vol. 25, no. 5, pp. 781–795, 2010.

[6] G. E. Fang-Zhen, Z. Wei, Y.-M. Tian, and Y. Lu, "High-dimensional optimization problems via disturbance chaotic ant swarm algorithm," *J. Comput. Appl.*, vol. 31, no. 4, pp. 1084–1089, 2011.

[7] D. Jia, G. Zheng, B. Qu, and M. K. Khan, "A hybrid particle swarm optimization algorithm for high-dimensional problems," *Comput. Ind. Eng.*, vol. 61, no. 4, pp. 1117–1122, 2011.

[8] G. Yuan and M. Zhang, "A modified hestenes-stiefel conjugate gradient algorithm for large-scale optimization," *Numer. Funct. Anal. Optim.*, vol. 34, no. 8, pp. 914–937, 2013.

[9] Y. Ren and Y. Wu, "An efficient algorithm for high-dimensional function optimization," *Soft Comput.*, vol. 17, no. 6, pp. 995–1004, 2013.

[10] P. Wang, Y. Huang, C. Ren, and Y. M. Guo, "Multi-scale quantum harmonic oscillator for high-dimensional function global optimization algorithm," *Chin. J. Electron.*, vol. 41, no. 12, pp. 2468–2473, Dec. 2013.

[11] S. Peng, A. Ouyang, G. Yue, M. He, and X. Zhou, "Improved glowworm swarm optimization algorithm for high-dimensional functions," *J. Comput. Appl.*, vol. 33, no. 8, pp. 2253–2256, 2013.

[12] Y. S. Hong, J. Z. Zhou, and C. L. Liu, "Research of parallel artificial bee colony algorithm based on MPI," *Appl. Mech. Mater.*, vols. 380–384, pp. 1430–1433, Aug. 2013.

[13] C. Wang and J.-H. Gao, "A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization," *Optim. Lett.*, vol. 8, no. 2, pp. 477–492, 2014.

[14] M. Moshki, P. Kabiri, and A. Mohebalhojeh, "Scalable feature selection in high-dimensional data based on GRASP," *Appl. Artif. Intell.*, vol. 29, no. 3, pp. 283–296, 2015.

[15] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.

[16] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Inf. Sci.*, vols. 454–455, pp. 59–72, Jul. 2018.

[17] W. Long, J. Jiao, X. Liang, and M. Tang, "Inspired grey wolf optimizer for solving large-scale function optimization problems," *Appl. Math. Model.*, vol. 60, pp. 112–126, Aug. 2018.

[18] S. Özyön, C. Yaşar, and H. Temurtaş, "Incremental gravitational search algorithm for high-dimensional benchmark functions," *Neural Comput. Appl.*, vol. 33, no. 13, pp. 1–25, 2018.

[19] P. S. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Appl. Math. Comput.*, vol. 188, no. 1, pp. 129–142, 2007.

[20] S. Mirjalili, G.-G. Wang, and L. D. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, no. 6, pp. 1423–1435, 2014.

[21] J. Luo and B. Shi, "A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems," *Appl. Intell.*, vol. 49, no. 5, pp. 1982–2000, 2019.

[22] Z.-J. Teng, J.-L. Lv, and L.-W. Guo, "An improved hybrid grey wolf optimization algorithm," *Soft Comput.*, vol. 23, no. 15, pp. 6617–6631, 2019.

[23] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft Comput.*, vol. 66, pp. 232–249, May 2018.

[24] Z. Liang, J. Ouyang, and Y. Feng, ''A hybrid GA-PSO optimization algorithm for conformal antenna array pattern synthesis,'' *J. Electromagn. Waves Appl.*, vol. 32, no. 13, pp. 1601–1615, 2018.

[25] G. I. Sayed and A. E. Hassanien, ''A hybrid SA-MFO algorithm for function optimization and engineering design problems,'' *Complex Intell. Syst.*, vol. 4, pp. 195–212, Oct. 2018.

[26] R. Dong, S. Wang, G. Wang, and X. Wang, ''Hybrid optimization algorithm based on wolf pack search and local search for solving traveling salesman problem,'' *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 24, no. 1, pp. 41–47, 2019.

[27] W. Yong, W. Tao, Z. Cheng-Zhi, and H. Hua-Juan, ''A new stochastic optimization approach—Dolphin swarm optimization algorithm,'' *Int. J. Comput. Intell. Appl.*, vol. 15, no. 2, 2016, Art. no. 1650011.

[28] T.-Q. Wu, M. Yao, and J.-H. Yang, ''Dolphin swarm algorithm,'' *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 8, pp. 717–729, 2016.

[29] T. Wu, M. Yao, and J. Yang, ''Dolphin swarm extreme learning machine,'' *Cognit. Comput.*, vol. 9, no. 2, pp. 275–284, 2017.

[30] R. L. Devaney, ''An introduction to chaotic dynamical systems,'' *Acta Applicandae Math.*, vol. 40, no. 7, p. 72, 1987.

[31] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X.-S. Yang, ''Economic dispatch using chaotic bat algorithm,'' *Energy*, vol. 96, pp. 666–675, Feb. 2016.

[32] S. Mirjalili and A. Lewis, ''The whale optimization algorithm,'' *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[33] J. Santos, A. Ferreira, and G. Flintsch, ''An adaptive hybrid genetic algorithm for pavement management,'' *Int. J. Pavement Eng.*, vol. 20, no. 3, pp. 266–286, 2019.

[34] D. Chitara, K. R. Niazi, A. Swarnkar, and N. Gupta, ''Cuckoo search optimization algorithm for designing of a multimachine power system stabilizer,'' *IEEE Trans. Ind. Appl.*, vol. 54, no. 4, pp. 3056–3065, Jul./Aug. 2018.

[35] J. Liang, L. Wang, M. Ma, and J. Zhang, ''A fast SAR image segmentation method based on improved chicken swarm optimization algorithm,'' *Multimedia Tools Appl.*, vol. 77, no. 24, pp. 31787–31805, 2018.

[36] W. Qiao, K. Huang, M. Azimi, and S. Han, ''A novel hybrid prediction model for hourly gas consumption in supply side based on improved whale optimization algorithm and relevance vector machine,'' *IEEE Access*, vol. 7, pp. 88218–88230, 2019. doi: 10.1109/ACCESS.2019.2918156.

[37] H. Lu, K. Huang, M. Azimi, and L. Guo, ''Blockchain technology in the oil and gas industry: A review of applications, opportunities, challenges, and risks,'' *IEEE Access*, vol. 7, pp. 41426–41444, 2019. doi: 10.1109/ACCESS.2019.2907695.

[38] W. B. Qiao, B. D. Chen, and H. F. Lu, ''Gas load forecasting based on chaotic theory and volterra adaptive filter,'' *Scientia Sinica Technologica*, vol. 45, no. 1, p. 91, 2015.

[39] W. Qiao and H. Wang, ''Analysis of the wellhead growth in HPHT gas wells considering the multiple annuli pressure during production,'' *J. Natural Gas Sci. Eng.*, vol. 50, pp. 43–54, Feb. 2018.

[40] Q. Weibiao, L. Bingfan, and K. Zhangyang, ''Differential scanning calorimetry and electrochemical tests for the analysis of delamination of 3PE coatings,'' *Int. J. Electrochem. Sci.*, vol. 14, pp. 7389-7400, Aug. 2019. doi: 10.20964/2019.08.05.

WEIBIAO QIAO received the B.S. degree in information and computing science from Northeast Agricultural University, Harbin, Heilongjiang, in 2009, the M.S. degree in oil and gas storage and transportation engineering from Liaoning Shihua University, Fushun, Liaoning, in 2012, and the Ph.D. degree in oil and gas storage and transportation engineering from China Petroleum University, Qingdao, Shandong, in 2017.

Since 2017, he has been a Lecturer with the North China University of Water Resources and Electric Power, Zhengzhou, Henan. His research interests include the gas consumption forecasting, gas pipeline robot detection, and intelligent scheduling of gas storage group and so on.

ZHE YANG was born in Shanxi.

He received the B.S. degree in computer science from the Taiyuan University of Technology (TUT), Shanxi, China, in 2017.

He is currently pursuing the M.Eng. degree with the School of Computer Science, The University of Manchester, Manchester, U.K.

His current research interests inlcude swarm intelligence algorithm and machine learning.