Hindawi
Complexity
Volume 2021, Article ID 6676934, 15 pages
https://doi.org/10.1155/2021/6676934

Research Article

A Chaotic Disturbance Wolf Pack Algorithm for Solving Ultrahigh-Dimensional Complex Functions

Qiming Zhu, Husheng Wu, Na Li, and Jinqiang Hu

College of Equipment Support and Management, Engineering University of PAP, Xi'an 710086, China

Correspondence should be addressed to Husheng Wu; wuhusheng0421@163.com

Received 7 November 2020; Revised 11 January 2021; Accepted 2 February 2021; Published 12 February 2021

Academic Editor: Rui Wang

The optimization of high-dimensional functions is an important problem in both science and engineering. Wolf pack algorithm is a technique often used for computing the global optimum of a multivariable function. In this paper, we develop a new wolf pack algorithm that can accurately compute the optimal value of a high-dimensional function. First, chaotic opposite initialization is designed to improve the quality of initial solution. Second, the disturbance factor is added in the scouting process to enhance the searching ability of wolves, and an adaptive step length is designed to enhance the global searching ability to prevent wolves from falling into the local optimum effectively. A set of standard test functions are selected to test the performance of the proposed algorithm, and the test results are compared with other algorithms. The high-dimensional and ultrahigh-dimensional functions (500 and 1000) are tested. The experimental results show that the proposed algorithm features in good global convergence, high accuracy calculation, strong robustness, and excellent performance in high-dimensional functions.

1. Introduction

The rapid development of science and technology enriches human life. Meanwhile, the problems encountered in various fields, especially in engineering and technology related fields, become increasingly complex and diversified, for example, large-scale dispatching problems [1–3] and large-scale power system optimization problems [4–6]. The complexity of the problem increases exponentially with the increase of dimension, which reduces the performance of traditional optimization algorithms and thus the ability to solve this kind of problems well. Inspired by the living mode of creatures in nature, swarm intelligence algorithm contains the wisdom of nature for thousands of years. Compared with the traditional optimization algorithm, its structure is simple and easy to implement, so it is widely used. Swarm intelligence algorithm is applied to high-dimensional function optimization problems by many scholars. The study in [7] used neighborhood factors to improve the NFO algorithm, which is applied to complex optimization problems and achieves good results. Compared with the traditional particle swarm optimization (PSO) algorithm, the improved particle swarm optimization algorithm in [8] is applied to high-dimensional function optimization because its convergence accuracy in high-dimensional function is improved. The study in [9] applied the improved dolphin colony algorithm to solve the high-dimensional function optimization problem and achieved good results. Although the performance of the improved method outweighs the original algorithm in high-dimensional function optimization, the improved one still has low convergence precision and low solving efficiency, making it difficult to meet the requirements of more and more refinement in the field of engineering science and technology.

In [10], wolf pack algorithm (WPA) was proposed based on a detailed analysis of wolves' scouting behavior and prey allocation methods. WPA is abstracted as three kinds of artificial wolves (the lead wolf, the scout wolf, and the ferocious wolf). In the process of hunting, there are three kinds of intelligent behaviors (the calling of the lead wolf, the wandering of the scout wolf, and the siege of the ferocious wolf), in addition to the generation rule of the "winner is the king" and the population updating rule of "survival of the fittest". As a novel swarm intelligence algorithm, WPA has good performance in global optimization and local exploration [11]. Its optimization method is different from the previous bionic intelligent algorithms such as particle swarm optimization (PSO) [12], ant colony optimization (ACO) algorithm [13], shuffled frog leaping algorithm (SFLA) [14], genetic algorithm (GA) [15], and artificial bee colony (ABC) algorithm [16]. It is widely adopted in PID parameter tuning [17], path planning [18], knapsack problem [19], combinational optimization problem [20], etc. The study in [10] indicates that the superiority of WPA is obvious in solving complex and high-dimensional problems. However, some shortcomings still exist in the algorithm: for example, the initial population produced randomly makes the algorithm easily fall into local optimum, and low quality initial solution increases the calculation amount of the algorithm; the greedy search method and rigid wandering direction of wolf detection stand great chances of falling into local optimum and missing the optimal solution; the step size of three linear relationships leads to the inflexible movement of the whole wolf pack, increasing the computational complexity of the algorithm.

The study in [21] makes use of tent chaotic mapping method to improve the quality of initial solution to the wolf pack algorithm, which endows the WPA with faster convergence speed and higher solution accuracy. In [22], a chaos optimization method based on logistic map was used to initialize the population, which improved the optimization accuracy and convergence rate of WPA. In [23], the d-dimensional chaotic variables were mapped to the solution space to obtain the initial wolf group. With the aid of opposition based learning (OBL), the study in [24] adopts an opposite wolf pack initialization method and puts forward an oppositional wolf pack algorithm (OWPA), which improves the quality of the initial solution and the convergence speed of the algorithm.

However, considering only the initial population quality or initial population distribution, the above method is still prone to fall into local optimum and the convergence accuracy is not high in some functions. Meanwhile, the above improved method is not applied to high-dimensional function optimization, so it is difficult to show that it has advantages in high-dimensional function optimization.

Compared with the existing research work, the four main innovative aspects of this paper are as follows:

(1) Chaotic self-logical sequences are used and mapped to the solution space, and then the initial population is further optimized by the reverse formula

(2) Disturbance factor is used to disturb the wolf 's scout direction to increase the randomness of the scout

(3) The scouting step length of the scout wolf and the ferocious wolf is set as the adaptive step size, so as to adjust the moving speed according to their own position during the movement process

(4) A lot of experimental comparisons are made on hyperdimensional function showing the effectiveness of the improved algorithm

The remainder of this paper is organized as follows: In Section 2, the chaotic disturbance wolf pack algorithm (CDWPA) is described in detail. In Section 3, the theoretical analysis of the algorithm is provided. In Section 4, a variety of benchmark functions with different mathematical characteristics are tested. The simulation turns out that the CDWPA possesses higher accuracy in convergence and computational robustness and can effectively solve the problem of high-dimensional function optimization.

2. Chaotic Disturbance Wolf Pack Algorithm

The wolves are located in the Euclidean space $N \times D$, in which N is the total number of artificial wolves in the pack and $D$is the number of variables to be optimized. The position of the artificial wolf X_i can be expressed as $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, where X_{id} is the position of the artificial wolf X_i in the d-dimensional variable space$d = (1, 2, \ldots, D)$ to be optimized; the odor concentration of prey perceived by the artificial wolf X_i is $Y_i = f(X_i)$, where Y_i is the value of the target function. The improved wolf pack algorithm is composed of chaos reverse initialization population, wolf generation, adaptive and disturbance scout, fierce wolf siege, and population update rules. The algorithm flowchart is shown in Figure 1.

2.1. Chaotic Opposite Initialization of the Population. Chaotic state exists widely in nature and society. It has the characteristics of randomness, ergodicity, and regularity. Chaotic movement can traverse all the states according to its own law in a certain range. Therefore, in order to ensure the diversity and randomness of the individual population of wolves X_i, this paper adopts logical self-mapping (LSM) function to generate the wolf pack sequence. First, random wolves X_i were created, and then wolves X_i were mapped to chaotic space $(-1, 1)$ by

$$Z_i = 2 \times \frac{X_i - \mathrm{lb}}{\mathrm{ub} - \mathrm{lb}} - 1, \tag{1}$$

$$Z_{i+1} = 1 - 2 \times Z_i^2. \tag{2}$$

To intuitively display the characteristics of the above chaotic maps, the probability distribution characteristics of the logical self-mapping on the (0, 1) interval are compared with those of Gauss map [25], sinusoidal iterator [26], and logistic map [27]. The calculation of probability distribution is estimated as follows: 50000 chaotic numerical points are generated by 50000 iterations of chaotic map, the interval (0, 1) is equally divided into 100 intervals, and the probability of these 50000 points falling into 100 equal intervals is counted; the probability distribution diagram is shown in Figures 2–5.

From the analysis of Figures 2–5, the distribution of sinusoidal iterator is low in the front end but high in the back end, and most of the initial solutions are distributed in the interval $(0.5, 1)$. If the optimal value of the function problem does not exist in this interval, it will easily miss the optimal solution by the sinusoid iterator. The Gauss map is just the opposite; its probability distribution shows a decreasing trend, and the distribution trend of the logistic map is very large in middle and small at both ends. If the global optimal solution of the function problem is distributed in the middle,

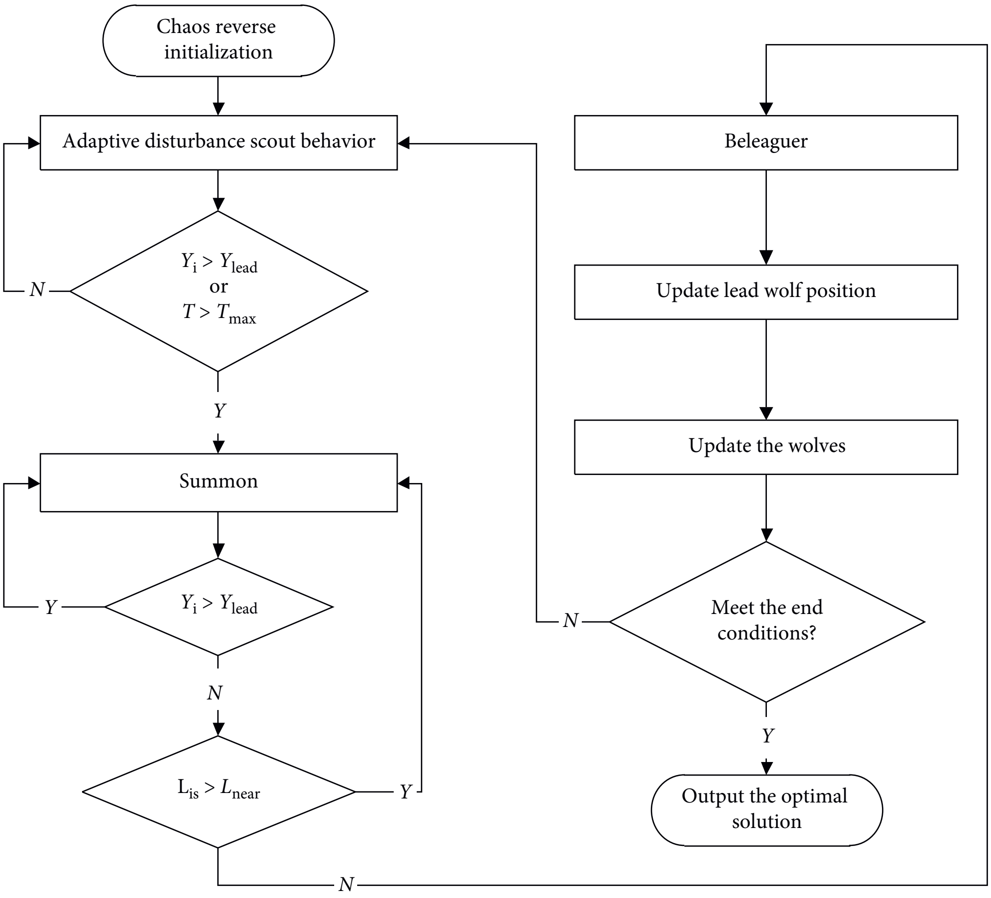

Chaos reverse initialization
Adaptive disturbance scout behavior
$Y_i > Y_{lead}$ or $T > T_{max}$
N
Y
Summon
$Y_i > Y_{lead}$
Y
N
$L_{is} > L_{near}$
Y
N
Beleaguer
Update lead wolf position
Update the wolves
Meet the end conditions?
N
Y
Output the optimal solution

Figure 1: Flowchart of CDWPA.

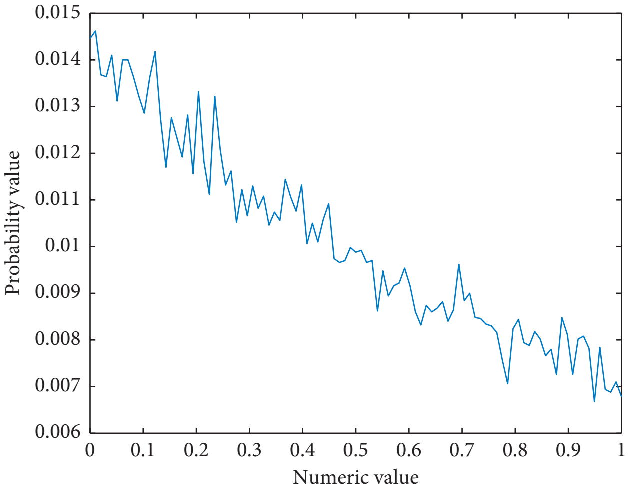


Figure 2: Gauss map.

a large number of invalid searches will appear, which will be unfavorable to global optimization. LSM chaotic sequences are uniformly distributed in the interval (0, 1). This distribution can avoid oversearching in some local areas (for example, a large number of searches of logistic maps are concentrated at both ends of the interval), thus reducing the adverse effects on the optimization algorithm due to the mismatch between the distribution characteristics of chaotic sequences and the position of the global optimal solution of the optimization problem.

If the definition domain of logical self-mapping function is $(-1, 1)$ except for 0 and 0.5, chaos will take effect in logic definition. After the chaotic variable sequence is obtained by chaotic search, the chaotic ergodic sequence should be

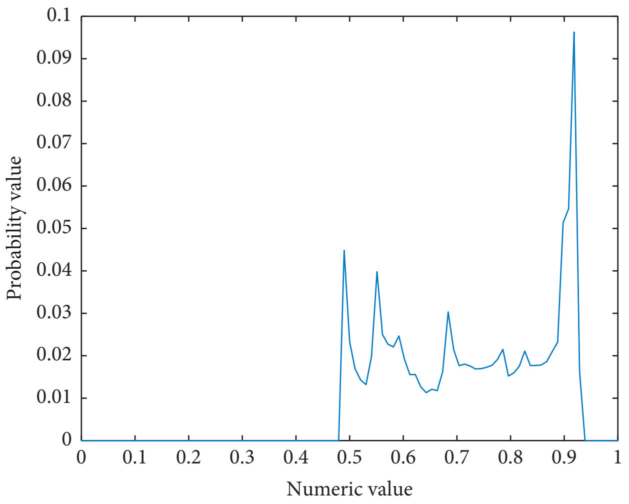


Figure 3: Sinusoidal iterator.

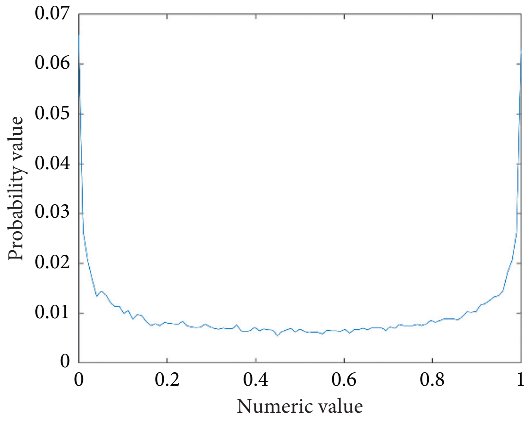


Figure 4: Logistic map.

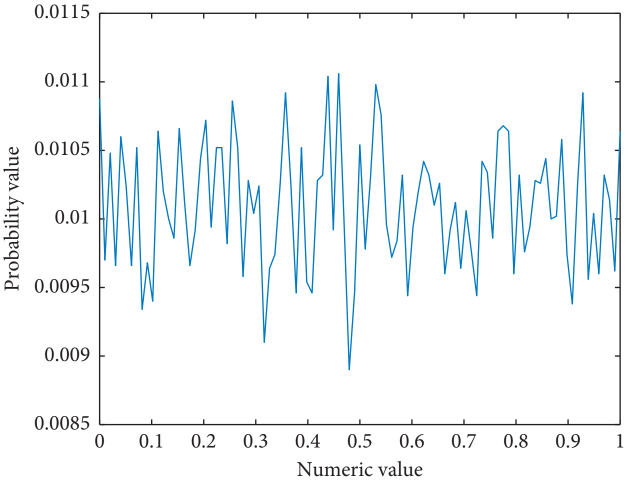


Figure 5: LSM.

transformed into the original solution space according to the following formula to evaluate the fitness value.

$$X_{i+1} = 0.5 \times (\mathrm{ub} - \mathrm{lb}) * Z_{i+1} + 0.5 \times (\mathrm{ub} + \mathrm{lb}). \quad (3)$$

Among them, uband lbrepresent the upper and lower boundaries of the artificial wolf X_i, respectively. Besides, the objective function values of the system random wolf pack X_i and chaotic mapping wolf pack X_{i+1} are calculated severally. Once the better solution is found in this process, the better wolf pack will replace the original wolf pack. Then the optimized wolf pack is operated reversely to calculate the reverse wolf pack.

$$OX_i' = L_d + U_d - x_{i+1}. \quad (4)$$

Many artificial wolves with better target function values are selected from the $\{X_{i+1} \cup OX'\}$ to form the initial wolves and complete the initialization of the wolves.

2.2. *Selecting Lead Wolf.*

The lead wolf, responsible for directing the whole to cooperate in hunting, is the artificial wolf with the largest odor concentration of prey in each generation. In the algorithm, the artificial wolf with the largest objective function value is selected as the lead wolf in the current wolf pack. There is only one lead wolf. If there are multiple wolves with the same objective function value, one of them will be randomly selected as the leader. Meanwhile, the lead wolf is not a life-long system. In the process of operation, if the objective function value of the artificial wolf X_i is better than that of the current one, the artificial wolf X_i will replace the former one and become the current leader. The lead wolf does not perform the behavior of wandering, besieging, and so on.

2.3. *Adaptive Disturbance Scout.*

After the lead wolf is determined, the scout wolf proceeds with optimization according to its own position, and the basic calculation formula is the following one:

$$x_{id} = x_{id} + \mathrm{step} \times \sin\left(2\pi \times \frac{p}{h}\right). \quad (5)$$

In this formula, x_{id} refers to the place of the current searching wolf, stepis the walk length, h is the walk direction, $p \in (1, h)$. After h is given in the formula, the searching wolf can only walk in a fixed direction, which is easy to fall into local optimum. Therefore, a random disturbance factor $\delta = \mathrm{rand}(\mathrm{round}(p/p))$ is designed to change the walking formula to

$$x_{id} = x_{id} + \mathrm{step} * \sin\left(2\pi \times \frac{p}{h} + \delta\right). \quad (6)$$

At the same time, considering the distance between the detection wolf and the lead wolf, the scouting step length is improved to

$$\mathrm{step} = \mathrm{rand} \times \mathrm{norm}\left\| x_i - x_{best} \right\|, \quad (7)$$

where x_i stands for the current location of the lead wolf and x_{best} is the optimal location of the lead wolf, so that the detection wolf can not only cover the fixed search range previously, but also search other directions randomly, which increases the optimal searching ability by breaking through the encirclement. The walking location set of the detection wolf is $x_{id} = \{x_{id}^1, x_{id}^2, \ldots, x_{id}^h\}$, and the value of the target function and that of the lead wolf are analyzed. If the fitness value is greater than the current lead wolf, then the current wolf detector becomes the lead wolf.

By analyzing (5), after the search direction h is fixed, the migration of different wolf detectors is equivalent to the movement in fixed directions of h. As shown in Figure 6, if the optimal prey is located at the parallel lines of the search direction, the detection wolf will be easily trapped into the local optimum and escape will be difficult. When the disturbance factor δ and adaptive step size are added, the search direction of the wolf detector is more random and possesses strong ability to break through; thus, it is not easy to fall into local optimum.

2.4. The Besiege of Ferocious Wolf.

2.4. The Besiege of Ferocious Wolf. When the scouting behavior is over, the current artificial lead wolf sends out a calling behavior to inform the other artificial wolves of their own location and fitness value. The rest of the artificial wolves move closer to the lead wolf in a rush step according to their distance from the lead wolf.

$$x_{id}^{k+1} = x_{id}^k + \text{step}_b^d \times \frac{g_d^k - x_{id}^k}{\left|g_d^k - x_{id}^k\right|}. \tag{8}$$

In the formula, g_d^k stands for the current location of the lead wolf. step_b^d is the besieging size of ferocious wolf. When the artificial wolf is far away from the lead wolf, it approaches the lead wolf with a larger step length; when the artificial wolf is close to the lead wolf, it approaches the lead wolf with a smaller step length. In addition, if the odor concentration of prey perceived by the *ferocious* wolf is greater than that of the current lead wolf, the strong one will become the leader, guiding the wolf pack to hunt until it is replaced by a better wolf or its operations is over.

After the iteration of the $k + 1$, if the odor concentration of prey perceived by wolf W_i is $Y_i^{k+1} > Y_{\text{lead}}^k$, then the ferocious wolf W_i replaces the former wolf leader and initiates the calling behavior. If the odor concentration of prey perceived by wolf W_i is $Y_i^{k+1} < Y_{\text{lead}}^k$, then the ferocious wolf W_i will continue to attack until the distance L_{is} between wolf W_i and the lead wolf is less than the determined distance L_{near}; it changes into besiege behavior. Given that the value range of the d-variable to be optimized is $[\min_d, \max_d]$, w is the determinant factor in distance, and the determinant distance is

$$L_{\text{near}} = \frac{1}{Dw} \sum_{d=1}^{D} \left|\max_d - \min_d\right|. \tag{9}$$

The Act of Siege. After the attack, the wolves besiege the prey to capture it. At this time, the position of the wolf is regarded as the location of the prey. If the prey's position in the d-dimensional space is G_d^k, the siege behavior of wolves can be expressed by the following equation:

$$x_{id}^{k+1} = x_{id}^k + \lambda \cdot \text{step}_c^d \cdot \left|G_d^k - x_{id}^k\right|, \tag{10}$$

where λ is the random integer evenly distributed in the interval $[-1, 1]$ and step_c is the siege step length. If the odor concentration of prey perceived by artificial wolf is greater than that of its original position after the siege, the location of the artificial wolf will be updated; otherwise, the location of the artificial wolf will remain unchanged.

2.5. Updating the Wolves.

2.5. Updating the Wolves. To maintain the competitiveness and diversity of the wolf pack, after the end of the hunting behavior, the wolf pack will distribute the prey according to their contributions and eliminate the artificial wolf R with the worst fitness value. At the same time, the artificial wolf R will be randomly generated to complete the population updating. R is the random integer in $[(N/2\beta), (N/\beta)]$, and β will be taken as the updating proportion factor.

The steps of chaotic disturbance wolf pack algorithm are as follows:

Step 1. Chaos reverse wolf group initialization: initialize the spatial coordinates of wolf pack randomly in the solution space according to (1)–(4).

Step 2. Adaptive disturbance walk: the walk strategy is carried out on (6)–(7).

Step 3. Calling behavior: the lead wolf sends out signals based on its position; the besieging wolf takes adaptive attack strategy to approach the lead wolf after receiving the signal.

Step 4. The siege of ferocious wolf: after reaching the siege distance, the ferocious wolf starts to encircle the prey.

Step 5. Update the location of the wolf pack: sort on the basis of the fitness value of the wolf pack, and select the best one.

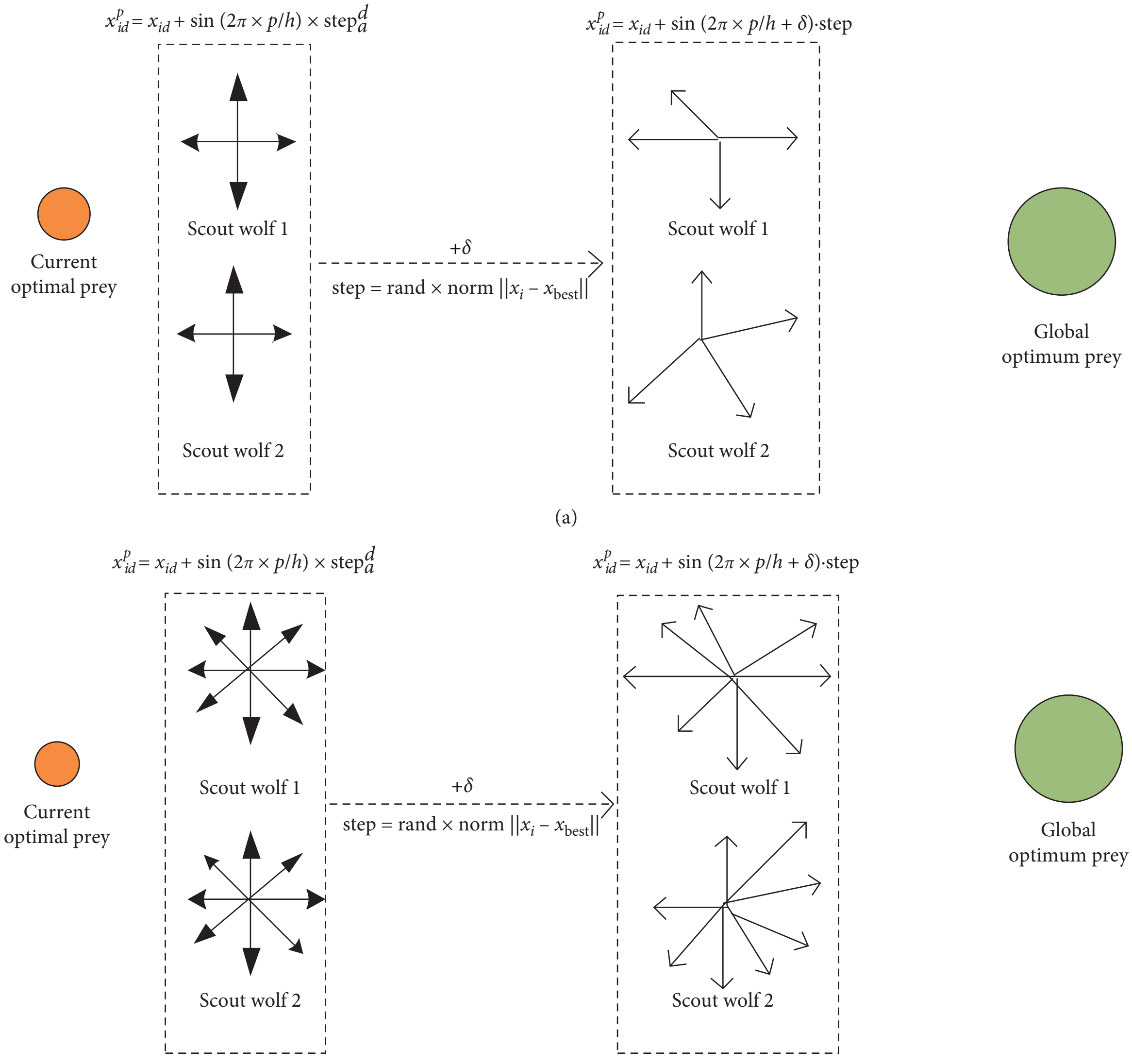
Step 6. Population regeneration: individuals with poor fitness in the population will be eliminated, and an equal number of new individuals will be generated.

Step 7. Repeat the process of 2–6 until the maximum number of iterations is reached or the algorithm accuracy meets the threshold.

3. Theoretical Analysis of the CDWPA Algorithm

3.1. Convergence Analysis of the Algorithm. Markov chain is a stochastic process with no aftereffect, which is often used to prove the convergence of the algorithm [28]. CDWPA is a process that repeatedly produces chaotic sequence, reverse scouting, summoning, rush, besieging, and wolf pack renewal behavior. Each behavior of the population is only related to the current state of the population but has nothing to do with the previous state. Therefore, the CDWPA population sequence conforms to Markov chain.

Reference [29] has proved that if an intelligent algorithm satisfies the following two conditions, the intelligent algorithm converges to the global optimal solution with probability 1.



$x_{id}^{p} = x_{id} + \sin(2\pi \times p/h) \times \mathrm{step}_a^d$
$x_{id}^{p} = x_{id} + \sin(2\pi \times p/h + \delta)\cdot\mathrm{step}$
Current optimal prey
Scout wolf 1
Scout wolf 2
$+\delta$
step = rand × norm $||x_i - x_{\mathrm{best}}||$
Global optimum prey

(a)

(b)

FIGURE 6: Disturbed scout and fixed scout: (a) when $h = 4$, $p = 1:4$; (b) when $h = 8$, $p = 1:8$.

Condition one: the sum of any two solutions x_1 and x_2 in the solution space x_2 can be obtained by various operators in x_1 of the algorithm

Condition two: if the population sequence $Q_1, Q_2, \ldots, Q_N$ is monotone

According to relevant theories, the problem of proof is transformed into the following two points:

Point one: the CDWPA population sequence is ergodic

Point two: CDWPA is a finite homogeneous Markov chain

If the above two conditions are satisfied, CDWPA will converge to the global optimal solution with probability 1.

Assuming that the search space of CDWPA is Q and the state changes of state space caused by chaotic sequence, reverse population, scouting, rushing, besieging, and population updating are, respectively, expressed by transition probability H, F, Y, B, W, Z, then the transition probability matrix of Markov chain of CDWPA is obtained as

$$P = H \times F \times Y \times B \times W \times Z. \tag{11}$$

The definitions are given as follows:

Definition 1. Assume the P_{ij} to be the transition probability matrix of a Markov chain. If there is such a matrix for $i, j \in Q \ni k \geq 1$, resulting in $P_{ij}^{k} > 0$, then the Markov chain is said to be irreducible.

Definition 2. Assume that there is a nonempty set $I = \{k \mid k \geq 1, P_{ij}^{k} > 0, \forall i, j \in Q\}$. If the greatest common divisor of the set I is 1, then the Markov chain is aperiodic.

Definition 3. Assume that $u_i = \sum_{k}^{\infty} k \cdot P_{ij}^{k}$ for a recurrent state i. If $u_i < +\infty$, u_i is called normal return. In particular, if i is normal recurrent and aperiodic, then the Markov chain is ergodic.

To prove the convergence of CDWPA, we have the following two points.

Point one: the Markov chain of CDWPA population sequence is ergodic.

The following is proved: (1) Markov chain of CDWPA population sequence is irreducible. Suppose the generation population $k$of the algorithm is $Q_k = \{X_1, X_2, \ldots, X_N\}$, $X_i$is the state of the first artificial wolf. Because the transition probability matrix $P_{ij} = P\{Q_{k+1} = j \mid Q_k = i, k \geq 1\}$ of the population Markov chain is only related to the initial and terminal states i, j and $Q_k > 0$ is always true, the transition probability matrix P of a population is positive definite matrix. According to Definition 1, the Markov chain of CDWPA population sequence is irreducible. (2) The Markov chain of CDWPA population is aperiodic and irreducible. For a given $k > 0$, the Markov chain obtained from condition one is irreducible, and $\ni j \in Q$ is inevitable, so that it holds that $P_{ij} > 0$, and combined with Definition $2, k = 1$. Therefore, the maximum common divisor of $Q$is 1, so the Markov chain of CDWPA population is aperiodic. From (1) The Markov chain of CDWPA population is aperiodic and irreducible.

The Markov chain of CDWPA population is ergodic.

Since the values of the transition matrix H, F, Y, B, W, and Z are all within [0, 1] and P_{ij} is the probability of the state$i$being transformed into a state$j$ through various behaviors, there must be $0 \leq P_{ij} \leq 1$; let $\sigma = \max\{P_{ij} \mid i, j \in \Omega\}$. The *Cauchy–Riemann equation* and Definition 3 prove $u_i = \sum_{k}^{\infty} k \cdot P_{ij}^{k} \leq \sum_{k}^{\infty} k\varepsilon^{k} \leq \infty$. In conclusion, the Markov chain of CDWPA population sequence is ergodic, and point one is proved.

Point two: CDWPA is a finite homogeneous Markov chain.

It is proved that each generation of population$Q_k$is limited, and so is Markov chain; secondly, the algorithm repeatedly generates chaotic sequence, reverse scouting, summoning, rush, siege, and wolf pack update behavior to find better prey, and the individual renewal has the characteristics of high-quality selection. The generation of population is only related to current generation. After repeated iterative optimization, CDWPA can obtain a set of sequence solutions, and the sequence is a finite homogeneous Markov chain. To sum up, reasoning 1 and 2 show that the Markov chain of CDWPA population sequence is ergodic, and its sequence solution is a finite homogeneous Markov chain. Therefore, it is proved that CDWPA converges to the global optimal solution of the problem with probability 1.

3.2. Time Complexity Analysis of the Algorithm.

3.2. Time Complexity Analysis of the Algorithm. The time complexity reflects the operation efficiency of the algorithm. The time complexity of cuckoo algorithm is analyzed in [30]. In this paper, the same method is adopted to analyze the time complexity of CDWPA.

In the WPA, we set the size of the wolf pack as N and the individual dimension as n. If the time of step size step, update scale factor β, search direction of wolf $T_{\max}$, and judging distance d_{near} is η_0, the time of generating random number is η_1, and the time of solving fitness function value is $f(n)$, then the time complexity of initialization phase is shown in

$$O(\eta_0 + N(n \cdot \eta_1 + f(n))) = O(n). \tag{12}$$

The wolf pack was sorted in line with the fitness value, the time of the lead wolf was selected as η_2, and the time for the artificial wolf to execute the walking strategy was calculated as η_3. In the summoning step, the distance between the artificial wolf X_i and the lead wolf X_{lead} is η_4, the time needed to move the position of a single wolf in each dimension is η_5, and the time to judge whether the attack range is reached is η_6; then, the time complexity of this stage is given as

$$O(N(\eta_2 + \eta_3 + \eta_4 + n \cdot \eta_5 + \eta_6)) = O(n + f(n)). \tag{13}$$

From the above formula, we can get the time complexity of the (WPA) to obtain the optimal solution of each generation.

$$T_n = O(n) + O(n + f(n)) = O(n + f(n)). \tag{14}$$

The process of CDWPA is analyzed. In the initialization stage, the generation time of logical self-mapping sequence is t_1, and the time of reverse generation population is t_2. The other generation parameters, dimensions, and solution fitness function values are the same as those of WPA. Then, the time complexity of CDWPA initialization phase is given as

$$O(\eta_0 + N(n \cdot \eta_1 + f(n) + t_1 + t_2)) = O(n + f(n)). \tag{15}$$

The time of calculating the distance between the detection wolf X_i and the lead wolf added by the adaptive step length is η_7, and the other wolves are selected to perform the same steps as WPA, such as scouting, summoning, and besieging; then, the time complexity of this stage is

$$O(N(\eta_2 + \eta_3 + \eta_4 + n \cdot \eta_5 + \eta_6 + \eta_7)) = O(n + f(n)). \tag{16}$$

To sum up, the total time complexity of CDWPA to find the optimal value of each generation is shown as

$$O(n) + O(n + f(n)) = O(n + f(n)). \tag{17}$$

From the above analysis, compared with WPA, the time complexity of CDWPA does not change, and the efficiency of CDWPA does not decrease.

4. Simulation Experiment and Algorithm Validity Test

4.1. Test Function. In order to test the performance of the CDWPA proposed in this paper, we select 10 standard test functions in [31] (dimensions from 2 to 200) to conduct the first set of experiments, and compare the optimization performance with wolf pack algorithm (WPA), oppositional wolf pack algorithm (OWPA) in [24], chaotic wolf pack algorithm (CWPA) in [22], PSO, ABC, and ASFA [32]. In order to further verify the ability of the improved algorithm to solve high-dimensional complex functions, the second set of experiments was carried out. Four high-dimensional (500 and 1000 dimensions) functions covering different types were selected, and the test results of seven algorithms were compared. The specific characteristics of the 10 standard test functions are shown in Table 1. “U” is unimodal function, “M” is multimodal function, “S” is separable function, and “N” is nonseparable function. Unimodal function is used to test the mining ability of the algorithm, multimodal function tests the exploration ability of the algorithm, and high-dimensional function tests the ability of the algorithm to solve complex problems.

To express the optimization effect more clearly, BEST, WORST, MEAN, and STD are selected to represent the best objective function value, the worst value, the mean value of target function, and the standard deviation of target function at the end of the optimization result. SR (successful rate) indicates the success rate of optimization, and the ideal optimal value of the function is set as R. when the relationship of them satisfies (18), the optimization is successful.

$$|\mathrm{BEST} - R| \leq 1\mathrm{OE} - 6. \tag{18}$$

4.2. Simulation Experiment and Result Analysis. Because different authors adopt various parameters, this paper selects the same general parameters; for example, the population size is 100 (wolves, particle swarm, fish swarm, bee colony, etc.), the maximum number of iterations in the first set of experiments is 200, and the maximum number of iterations in the second set is 2000. The other parameter settings are shown in Table 2.

The above experimental environment is as follows: HP Shadow Genie 4, Windows 10, inter® Core™ i7-8750H; the program is implemented by MATLAB R2017b, M language.

In the above table, $T_{\max}$ means the maximum scouting times. S means step length factor. $\omega\omega$ means determination distance factor. β means renewal scaling factor. ω means inertia weight. c_1 and c_2 mean learning factor. limit_v means individual speed limit. N means number of bees. limit means controls parameter. trynum means maximum number of temptations. v means sense of distance. de means crowding factor. step means step size.

4.2.1. Analysis of the First Set of Experimental Results. Table 3 shows the optimal value, mean value, worst value, standard deviation, and success rate of the seven algorithms for ten standard test functions.

Table 3 shows the comparison of the first set of experimental results of the seven algorithms. The analysis shows that the success rate of CDWPA for ten test functions is 100%, and the results are better than WPA, OWPA, and CWPA, except that the performance of Booth optimization of 2D function is worse than ABC, the performance of other function optimization is better than other algorithms, and the convergence accuracy of CDWPA in $f_2 f_3 f_5 f_6 f_7 f_9, f_{10}$ is improved. In particular, for functions f_1 and f_6, their convergence results achieved the theoretical optimal value 0. Simulation results show the effectiveness of the improved CDWPA.

In the case of the same population size and iteration times, on the solution of the low peak one-dimensional separable function sphere, the success rate of PSO, ABC, and ASFA was 0. The success rate of WPA, OWPA, CWPA, and CDWPA was 100%. Moreover, CDWPA is obviously superior to other algorithms in terms of optimal value, worst value, standard deviation, and average value, which indicates that CDWPA owns better calculation accuracy and robustness.

Except for ABC, all algorithms can successfully search for the solution of the unimodal low-dimensional indivisible functions Easom and Matyas. OWPA has the best optimal value, and CDWPA follows closely with slight difference. Moreover, it is better than OWPA and other algorithms in mean value, standard deviation, and STD, which shows that CDWPA algorithm has stronger stability and balance ability.

When solving the low-dimensional, multipeak, and separable function Booth, the success rate of all algorithms is 100% except ASFA which is 95%. CWPA has the best optimal value. Compared with WPA and the two improved wolf pack algorithms OWPA and CWPA, CWPA is better in terms of mean, variance, and worst value. However, CDWPA has no obvious advantage compared to ABC.

In solving the step of middle dimension, unimodal and separable functions, WPA and CDWPA can find the theoretical optimal value, and the success rate is 100%. CDWPA has fewer iterations and higher optimization efficiency. However, the OWPA with the same reverse strategy holds a success rate of only 50%, and its convergence accuracy is lower than that of CDWPA. In CDWPA, chaotic sequences are used to optimize the initial population.

For solving unimodal and multidimensional inseparable function on Eggcrate, the success rate of PSO, ABC, WPA, and CDWPA is 100%, and that of OWPA and CWPA is 95% and 40%, respectively, yet the success rate of ASFA is only 5%. CDWPA has the best performance in respect of optimal value optimization and is many orders of magnitude more accurate than other algorithms. At the same time, it performs better in mean value, standard deviation, and worst value.

For 200-dimensional functions, PSO, ABC, and ASFA cannot be well optimized when the number of iterations is set to 200, making it easy to fall into local optimization and

Table 1: Benchmark test function.

No.	Name	Expression	Trait	Search range	Min
1	Easom	$f(x) = -\cos(x_1)\cos(x_2) \times e^{(-x_1-\pi)2-(x_2-\pi)^2}$	UN	$[-100, 100]^2$	−1
2	Sphere	$f(x) = \sum_{i=1} x_i^2$	US	$[-1.5, 1.5]^2$	0
3	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	UN	$[-10, 10]^2$	0
4	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	MS	$[-32, 32]^2$	0
5	Eggcrate	$f(x) = x_1^2 + x_2^2 + 25(\sin x_1)^2 + (\sin x_2)^2$	MN	$[-2\pi, 2\pi]^2$	0
6	Step	$f(x) = \sum_{i=1}^{D} \lceil (x_i + 0.5)^2 \rceil$	US	$[100, 100]^{30}$	0
7	Sumsquares	$f(x) = \sum_{i=1}^{D} (x_i)^2$	US	$[-10, 10]^{100}$	0
8	Quadric	$f(x) = \sum_{i=1}^{D} (\sum_{k=1}^{D} x_k)^2$	MS	$[-30, 30]^{100}$	0
9	Griewank	$f(x) = (\sum_{i=1}^{D} (x_i)^2/4000) - (\prod_{i=1}^{D} (x_i^2/\sqrt{x_i}) + 1)$	MN	$[-600, 600]^{100}$	0
10	Ackley	$f(x) = -20e^{-0.2\sqrt{\sum_{i=1}^{D} (x_i^2/D)}} - e^{\sum_{i=1}^{D} \cos((2\pi x_1)/D)} + 20 + e$	MN	$[-32, 32]^{100}$	0

Table 2: Parameter setting of methods.

Methods	Parameter setting
WPA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
OWPA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
CWPA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
CDPWA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
PSO	$\omega = 0.7298, c_1 = c_2 = 1.4946, \text{limit}_v \in (-0.5, 0.5)$
ABC	$N = 2, \text{limit} = 100$
ASFA	trynum = 100, $v = 1$, de = 0.618, step = 0.1

difficult to escape. WPA, OWPA, CWPA, and CDWPA proposed in this paper have good performance on high-dimensional functions. By analyzing and comparing the characteristics of these algorithms in solving high-dimensional functions, the characteristics of CDWPA are illustrated.

First, the function Sumsquares belongs to the high-dimensional, unimodal, separable function, and Quadric is a peak multidimensional separable function. The success rate of WPA, OWPA, and CDWPA is 100%, and the success rate of CWPA is 90%. Compared with function f_5, the only difference from function f_7 and function f_8 is that both of them are high-dimensional function. This shows that CDWPA has its own advantages in high-dimensional function optimization, and CDWPA has improved a lot compared with the optimal value, average value, variance, and other indicators.

For Griewank and Ackley with 200-dimensional, multimodal, separable function, the success rate of WPA is 95% and 96%, and that of CWPA is 100% and 0, respectively, which are easy to fall into local extremum. For OWPA and CDWPA, the success rate is 100%, while CDWPA can get global extremum. At the same time, by comparing OWPA with CDWPA further, it turns out that CDWPA is better than OWPA in the precision of optimal value, and CDWPA is less than OWPA in variance, which indicates that CDWPA is better than OWPA in robustness.

For low-dimensional functions such as f_1, f_3, f_4, and f_5, the classical intelligent optimization algorithms of PSO, ABC, and ASFA have a good solving effect. Although CDWPA effect is improved, it is not very obvious, and the solving effect is even slightly lower than that of ABC on function f_4. However, as the dimensions rise to 100, the CDWPA advantage becomes apparent. The analysis shows that, for most heuristic optimization algorithms, the generation of random variables is usually based on a certain standard probability distribution, such as uniform distribution and Gaussian distribution. As chaos not only has randomness but also has better spatial ergodicity and unrepeatability, the algorithm is more diverse after adding chaos to random search based on certain probability distribution, and the possibility of jumping out of local extremum point is greater, which enables the algorithm to search at a relatively faster speed. When the dimension is low, the solution space is relatively simple, the distribution of the initial population can meet the needs of the solution space, and the advantages of chaos are not obvious. However, when the solution space becomes more complex, the quality of the solution is required to be higher, and the advantages of chaos and disturbance are fully embodied. This is the reason why the optimization effect of the algorithm is improved significantly in the case of high-dimensional complex functions and is not obvious in the case of low-dimensional functions.

4.2.2. Analysis of the Second Set of Experimental Results. Tables 4 and 5 show the mean value, variance, and optimal value of the four test functions calculated 20 times by seven algorithms on 500 and 1000 dimensions, respectively. Tables 6 and 7 show the P values for the rank sum test. Figures 7 and 8 show the optimization curves of seven algorithms in 500 and 1000 dimensions.

The advantages of CDWPA are not very obvious on low-dimensional functions. However, when the function dimension increases to 500 or even 1000 dimensions, the results of CDWPA are better than other algorithms in optimal value, mean value, and standard deviation. In the case of 500 dimensions, CDWPA achieves the theoretical optimal value of 0 towards Griewank function. With the increase of function dimension, the convergence accuracy of several algorithms decreases, but the optimization effect of CDWPA is obviously better than that of other algorithms

Table 3: The results of the first experiment were compared.

Dim	Fun	Index	PSO	ABC	AFSA	WPA	OWPA	CWPA	CDWPA
2	**Easom** f_1	BEST	**−1.00**	**−1.00**	**−1**	**−1**	**−1.00**	**−1.00**	**−1.00**
		MEAN	**−1.00**	**−1**	−0.7254	**−1.00**	**−1.00**	**−1.00**	**−1.00**
		WORST	**−1.00**	**−1**	−8.08e – 05	**−1.00**	−0.99	−0.99	**−1.00**
		STD	5.57e – 07	**0**	0.43	6.25e – 07	2.58e – 05	2.71e – 05	1.80e – 07
		SR/%	**100%**	**100%**	0	**100%**	35%	65%	**100%**
2	**Sphere** f_2	BEST	7.88	50.20	444	6.41e – 17	4.10e – 17	2.21e – 10	**1.67e – 18**
		MEAN	10.30	58.06	468.78	**7.17e – 16**	8.24e – 16	2.14e – 09	7.19e – 15
		WORST	12.24	66.75	488.93	6.53e – 15	2.79e – 15	1.74e – 08	**4.53e – 16**
		STD	1.36	4.47	12.99	1.43e – 15	8.24e – 14	3.78e – 09	**1.27e – 15**
		SR/%	0	0	0	**100%**	**100%**	**100%**	**100%**
2	**Matyas** f_3	BEST	2.53e – 10	1.47e – 06	7.29e – 10	1.50e – 17	7.67e – 26	1.22e – 15	9.78e – 25
		MEAN	1.16e – 08	9.91e – 05	4.08e – 7	2.46e – 07	8.38e – 20	1.66e – 7	**3.46e – 20**
		WORST	5.38e – 08	4.16e – 04	7.07e06	9.86e – 07	5.55e – 19	1.66e – 6	**2.54e – 19**
		STD	1.56e – 10	1.06e – 04	1.57e – 06	2.51e – 07	1.52e – 19	4.17e – 07	**7.50e – 20**
		SR/%	100%	15%	100%	100%	100%	100%	**100%**
2	**Booth** f_4	BEST	7.12e – 09	**1.50 e – 13**	1.50e – 08	1.61e – 08	1.48e – 07	1.22e – 07	1.46e – 09
		MEAN	3.67e – 07	**4.82 e – 11**	3.39e – 06	3.07e – 07	2.13e – 06	2.71e – 07	1.44e – 07
		WORST	1.53e – 06	**2.19 e – 10**	2.06e – 05	2.53e – 06	5.80e – 06	1.30e – 06	6.25e – 07
		STD	3.57e – 07	**5.66 e – 11**	4.65e – 06	5.92e – 07	1.75e – 06	3.74e – 07	1.58e – 07
		SR/%	**100%**	**100%**	95%	**100%**	**100%**	**100%**	**100%**
2	**Eggcrate** f_5	BEST	4.56e – 08	7.33e – 21	7.66e – 06	3.53e – 21	1.77e – 21	6.83e – 15	**1.28e – 22**
		MEAN	4.61e – 06	2.55e – 18	1.90	9.25e – 19	9.31e – 17	1.37e – 04	**4.21e – 20**
		WORST	2.08e – 05	1.20e – 17	9.48	6.71e – 18	1.05e – 15	0.001	**3.36e – 20**
		STD	5.31e – 06	2.89e – 18	3.89	1.64e – 18	2.48e – 16	2.48e – 04	**8.49e – 20**
		SR/%	**100%**	**100%**	5%	**100%**	**100%**	40%	**100%**
30	**Step** f_6	BEST	0	0	199592	**0**	0	7	**0**
		MEAN	0.43	4.25	2.39e + 05	**0**	0	34.45	**0**
		WORST	2	16	279999	**0**	4	138	**0**
		STD	0.55	4.01	1.90e + 04	**0**	1.01	29.22	**0**
		SR/%	60%	5%	0	**100%**	50%	0	**100%**
100	**Sumsquares** f_7	BEST	334.29	6.35e + 04	1.51e + 04	2.69e – 13	1.09e – 12	8.38e – 08	**1.83e – 14**
		MEAN	402.34	1.02e + 05	1.61e + 04	3.53e – 12	2.68e – 12	8.48e – 07	**1.24e – 13**
		WORST	518.25	1.17e + 05	1.73e + 04	1.47e – 11	2.49e – 11	2.27e – 06	**5.40e – 13**
		STD	44.88	1.21e + 04	588.52	4.12e – 12	5.44e – 12	6.76e – 07	**1.29e – 13**
		SR/%	0	0	0	100%	100%	100%	**100%**
100	**Quadric** f_8	BEST	11.82	1.66e + 04	4.99e – 05	1.40e – 12	1.02e – 12	5.33e – 07	**1.81e – 13**
		MEAN	19.44	2.42e + 04	0.02	2.90e – 11	1.90e – 11	3.92e – 06	**1.98e – 12**
		WORST	30.10	3.03e + 04	0.18	1.38e – 10	9.85e – 11	2.04e – 05	**1.82e – 11**
		STD	4.37	3.70e + 03	0.05	3.88e – 11	2.80e – 11	4.59e – 06	**4.03e – 13**
		SR/%	0	0	0	100%	100%	90%	**100%**
100	**Griewank** f_9	BEST	0.04	298	8.59e + 03	1.84e – 13	1.31e – 13	5.52e – 09	**8.50e – 14**
		MEAN	0.06	426.21	9.32e + 03	0.04	1.59e – 12	2.34e – 08	**1.72e – 13**
		WORST	0.09	513.55	1.00e + 04	0.82	7.11e – 12	7.30e – 08	**1.00e – 11**
		STD	0.01	60.07	381	0.18	2.37e – 12	1.90e – 08	**2.64e – 13**
		SR/%	0	0	0	95%	100%	100%	**100%**
100	**Ackley** f_{10}	BEST	2.23	19.70	1.59e + 16	6.94e – 08	6.45e – 08	3.62e – 05	**3.68e – 09**
		MEAN	2.50	19.91	8.01e + 14	2.12e – 07	1.43e – 07	1.51	**1.61e – 09**
		WORST	2.73	20.04	8.88e + 08	6.28e – 07	6.20e – 07	3.59	**3.76e – 09**
		STD	0.11	0.07	3.56e + 15	1.51e – 07	1.46e – 07	1.44	**9.36e – 09**
		SR/%	0	0	0	96%	100%	0	**100%**

and has better advantages in standard deviation, which demonstrates that CDWPA is better in robustness. In general, CDWPA performs better than the other six algorithms in 500- and 1000-dimensional test functions, which verifies the effectiveness of CDWPA in solving high-dimensional complex functions.

For low-dimensional space optimization, the guidance information of optimization is easy to obtain and confirm. The introduction of the optimization strategy based on the idea of "certainty" can improve the algorithm's optimization precision and convergence speed. However, high-dimensional data space to present a kind of highly nonlinear and asymmetric, nonconvex, multipeak complex features, such as value orientation information, is very difficult to obtain accurately; at this point in the optimization strategy, parameter is set to introduce some "random" strategy and will

Table 4: 500-dimensional comparison of experimental results.

Fun		PSO	ABC	ASFA	WPA	OWPA	CWPA	CDWPA
Sphere	MEAN	22.86	6.61	1.36*e* + 03	4.61*e* − 43	1.09*e* − 08	5.56*e* − 89	**9.53*e* − 149**
	STD	5.52	1.28	1.38*e* + 03	1.61*e* − 42	1.23*e* − 08	1.19*e* − 88	**2.92*e* − 148**
	BEST	15.61	4.87	1.66*e* + 03	8.36*e* − 76	7.15*e* − 10	6.48*e* − 93	**6.13*e* − 165**
Sumsquares	MEAN	4.16*e* + 03	6.29*e* + 04	5.66*e* + 04	1.03*e* − 39	8.20*e* − 05	1.65*e* − 85	**1.11*e* − 144**
	STD	287.36	2.47*e* + 04	907.03	3.35*e* − 39	7.39*e* − 05	5.63*e* − 85	**3.21*e* − 144**
	BEST	3.48*e* + 03	1.75*e* + 04	5.43*e* + 04	5.75*e* − 72	1.00*e* − 05	1.53*e* − 88	**1.92*e* − 160**
Griewank	MEAN	0.11	426	5.39*e* + 04	0.05	0.0015	8.88*e* − 17	**1.92*e* − 160**
	STD	0.009	60.07	1.01*e* + 03	0.22	0.0068	1.17*e* − 16	**0**
	BEST	0.10	117.15	5.21*e* + 04	0	1.77*e* − 16	0	**0**
Ackley	MEAN	1.71	19.91	−5.43*e* + 29	1.59*e* − 15	10.32	0.94	**0**
	STD	0.16	0.078	2.42*e* + 30	1.45*e* − 15	5.59	1.33	**8.88*e* − 16**
	BEST	1.43	13.50	−1.08*e* + 31	8.88*e* − 16	3.30	7.99*e* − 15	**8.88*e* − 16**

Table 5: 1000-dimensional comparison of experimental results.

Fun		PSO	ABC	ASFA	WPA	OWPA	CWPA	CDWPA
Sphere	MEAN	74.39	134.78	2.90*e* + 03	3.87*e* − 41	5.17*e* − 06	1.54*e* − 88	**3.78*e* − 149**
	STD	10.38	6.14	39.26	1.67*e* − 40	4.63*e* − 06	3.54*e* − 88	**1.69*e* − 148**
	BEST	47.79	125.56	2.83*e* + 03	1.65*e* − 74	1.01*e* − 16	2.71*e* − 92	**6.23*e* − 164**
Sumsquares	MEAN	3.87*e* + 04	2.96*e* + 06	1.19*e* + 05	2.58*e* − 39	0.0877	3.68*e* − 83	**1.37*e* − 145**
	STD	2.16*e* + 03	1.59*e* + 05	1.69*e* + 03	7.68*e* − 39	0.0707	3.68*e* − 82	**6.08*e* − 145**
	BEST	3.32*e* + 04	2.36*e* + 06	1.16*e* + 05	1.73*e* − 71	0.0133	3.85*e* − 89	**4.00*e* − 159**
Griewank	MEAN	0.22	5.6*e* + 03	1.11*e* + 05	0	0.01	1.55*e* − 16	**0**
	STD	0.01	329.22	1.22*e* + 03	0	0.03	1.54*e* − 16	**0**
	BEST	0.19	4.89*e* + 03	1.09*e* + 05	0	1.27*e* − 05	0	**0**
Ackley	MEAN	2.78	19.07	−7.01*e* + 43	2.48*e* − 15	9.68	0.90	**8.88*e* − 16**
	STD	0.06	0.13	3.08*e* + 44	1.81*e* − 15	4.61	1.29	**8.88*e* − 16**
	BEST	2.64	18.63	−1.3*e* + 45	8.88*e* − 16	4.73	4.44*e* − 15	**0**

Table 6: *P* value of different reference functions in 500 dimensions by rank sum test.

		PSO	ABC	ASFA	WPA	OWPA	CWPA
*F*1	*P*	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08
*F*2	*P*	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	5.2269*e* − 07	3.3819*e* − 04
*F*3	*P*	1.7313*e* − 12	8.0065*e* − 09	8.0065*e* − 09	0.3421	8.0065*e* − 09	3.8459*e* − 04
*F*4	*P*	2.3837*e* − 07	1.1267*e* − 08	1.1267*e* − 08	7.10e − 03	1.1267*e* − 08	1.0324*e* − 07

Table 7: *P* value of different reference functions in 1000 dimensions by rank sum test.

		PSO	ABC	ASFA	WPA	OWPA	CWPA
*F*1	*P*	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08
*F*2	*P*	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	6.7956*e* − 08	1.2009*e* − 06	2.5960*e* − 05
*F*3	*P*	8.0065*e* − 09	8.0065*e* − 09	8.0065*e* − 09	0.3421	8.0065*e* − 09	8.7603*e* − 06
*F*4	*P*	1.1267*e* − 08	1.1267*e* − 08	1.1267*e* − 08	7.10e − 03	1.1267*e* − 08	1.0468*e* − 07

help the algorithm to maintain the diversity of population, to produce more high-quality data processing, prompting algorithm to jump out of local extremum, and balance the exploration and development ability of the algorithm. Precisely, because the strategy adopted by CDWPA is the "directed random" method, the performance of the low-dimensional solution space is improved, but the gap with other algorithms is not obvious. However, in the high-dimensional solution space, the advantage of "directed random" is clearly reflected.

In order to determine whether the proposed CDWPA is significantly improved compared with other algorithms, this paper carried out nonparametric statistical test, namely, Wilcoxon's rank sum test [33]. The results of CDWPA in each benchmark function of 500 and 1000 dimensions were tested and compared with other algorithms by 5%. Tables 6 and 7 list the *P* values obtained through the tests, in which *P* values less than 0.05 indicated that the null hypothesis was rejected, so there was a significant difference at the level of 5%. Conversely, an underlined *P* value (greater than 0.05)

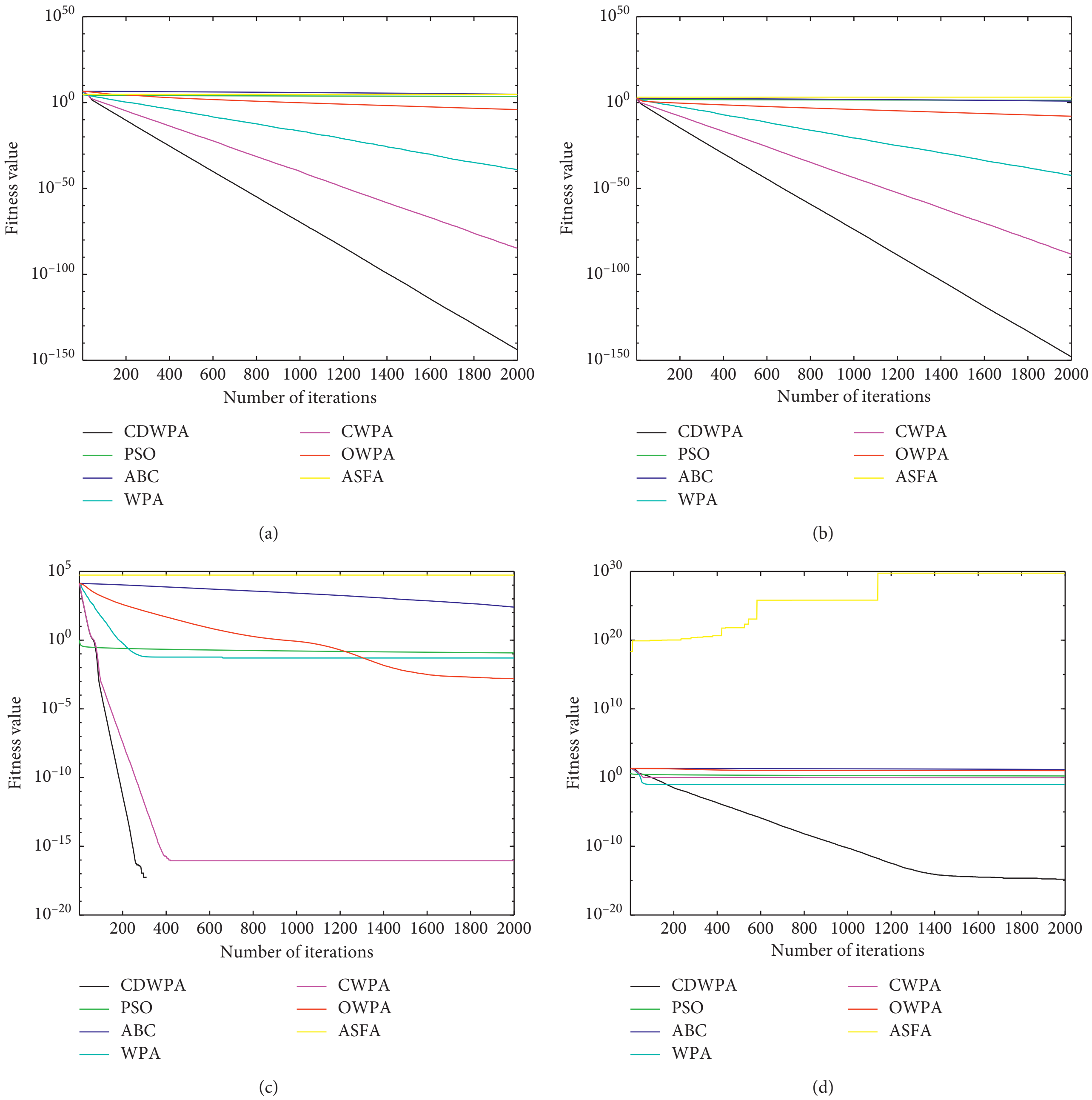
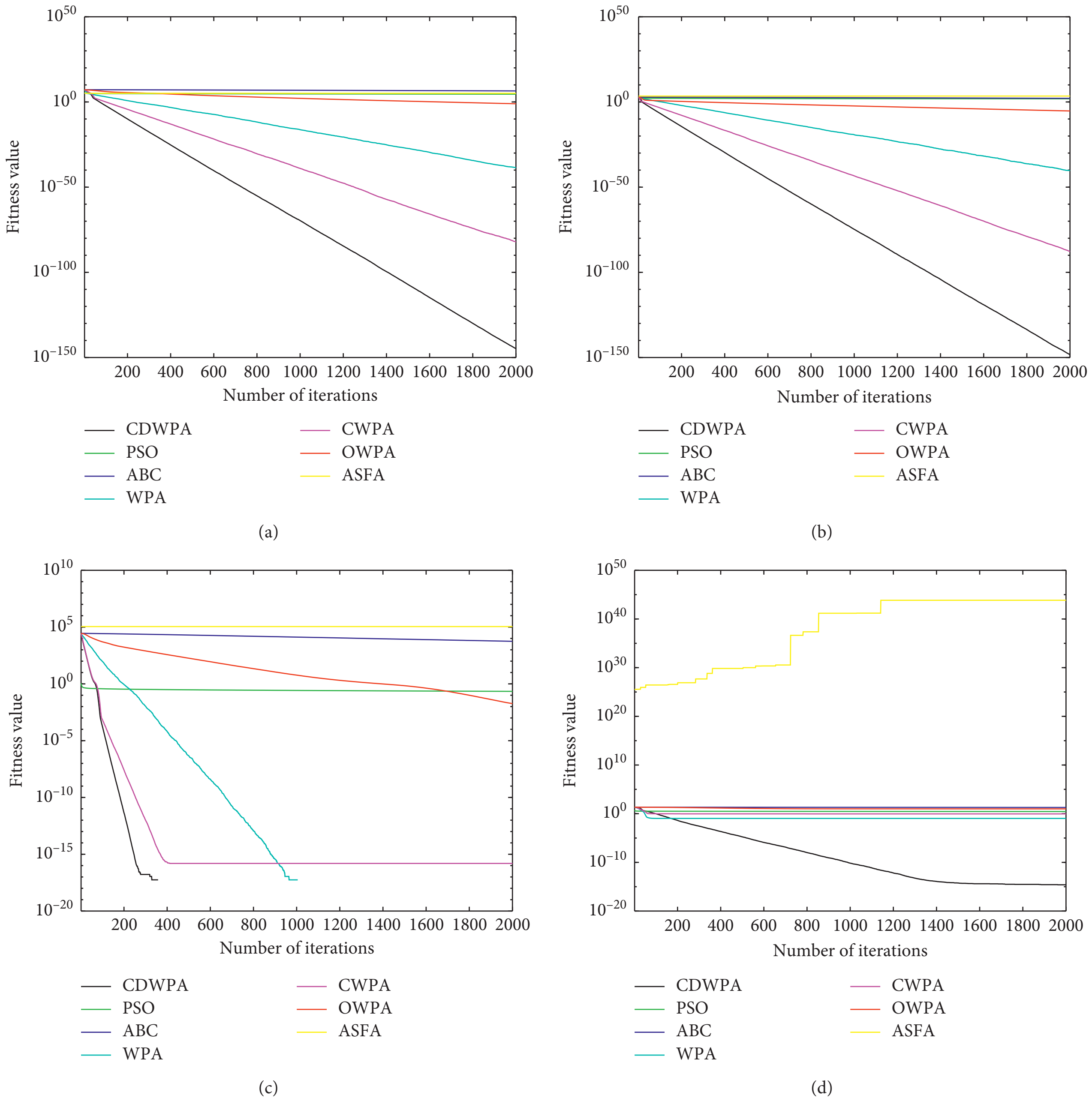


Figure 7: Four kinds of functions find the optimal curve comparison in 500 dimensions.

indicates no significant difference between the comparison values. It can be concluded from the results in Tables 6 and 7 that, in most comparisons, the P values are less than 0.05, which proves that the improvements achieved by the proposed CDWPA are statistically significant for most of the benchmark functions.

Analysis suggests that, for low-dimensional space optimization, the guidance information of optimization is easy to obtain and confirm. The introduction of the optimization strategy based on the idea of "certainty" can improve the algorithm's optimization precision and convergence speed. However, high-dimensional data space to present a kind of highly nonlinear and asymmetric, nonconvex, multipeak complex features, such as value orientation information, is very difficult to obtain accurately; at this point in the optimization strategy, parameter is set to introduce some "random" strategy and will help the algorithm to maintain the diversity of population, to produce more high-quality data processing, prompting algorithm to jump out of local extremum, and balance the exploration and development ability of the algorithm. Precisely, because the strategy adopted by CDWPA is the "directed random" method, the performance of the low-dimensional solution space is improved, but the gap with other algorithms is not obvious. However, in the high-dimensional solution space, the advantage of "directed random" is clearly reflected.



(a)

(b)

(c)

(d)

Figure 8: Four kinds of functions find the optimal curve comparison in 1000 dimensions.

Figure 9 shows the comparison of standard deviations of the seven algorithms in optimizing Ackley function in the dimensions of 300–1000. STD reflects the stability of algorithm in the optimization process, with dimension rising and solution space becoming more and more difficult. From the analysis of the STD of seven kinds of algorithms in Ackley function, we found that the standard deviation of stable algorithms such as PSO, ABC, OWPA, and CWPA optimization effect is very poor. PSO, ABC, and OWPA can even not succeed in finding the optimal value, were trapped into local optimum, and cannot get out of the local optimal trap after optimal "trap." However, because it falls into the local optimum at dimension 300 and cannot jump out of the local extreme value effectively, it still cannot find the optimization successfully. WPA and CDWPA have a good optimization effect. Interestingly, the change curves of standard deviations of the two algorithms are similar, with large pulsations at 500 and 1000 dimensions, indicating that the algorithm can effectively jump out of the local optimum and find better targets when it conducts optimization in higher dimensions. OWPA adopts the reverse strategy to optimize the initial population, but it does not consider the distribution location of the initial population, which easily leads to the aggregation of the population. The initial populations of WPA and CWPA were relatively dispersed, but due to the low quality of the initial populations and the

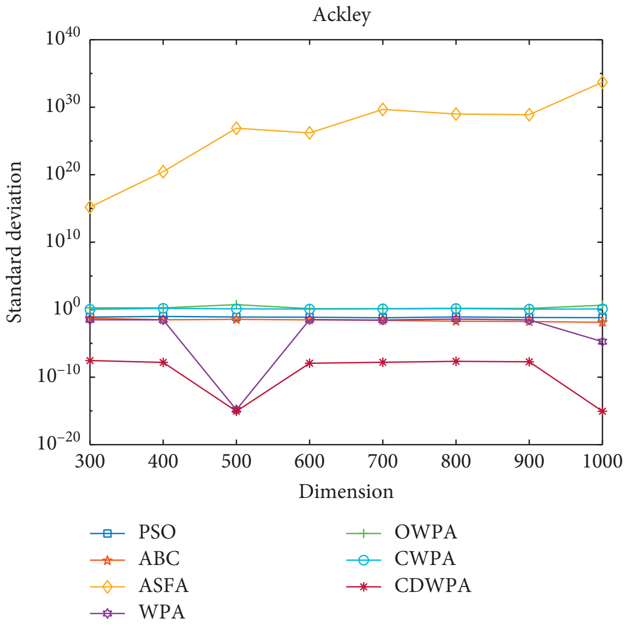


Figure 9: Comparison STD of Ackley with 7 algorithms in 300–1000 dimensions.

relatively fixed migration direction, the convergence accuracy was not high in the optimization. However, CDWPA considers both the initial population distribution position and the initial population mass and adds disturbance factors to the movement of wolves, which equips CDWPA with the most efficiency in high-dimensional function optimization.

5. Conclusion

In order to improve the convergence accuracy of wolf pack algorithm (WPA) in solving high-dimensional complex functions, it is easy to fall into local optimization. In this paper, we use logical self-mapping to generate chaotic sequence in the reinitialization stage, and optimize the initial population through the reverse wolf pack, adding disturbance factor in the process of scouting, which improves the ability of wolf detection to jump out of local optimum. Meanwhile, according to the distance away from the lead wolf, the strong wolf adopts appropriate step size to approach the lead wolf. The convergence of the algorithm is tested by Markov process with the selection of different types of functions. The simulation results show that the comprehensive performance of CDWPA is better than that of wolf pack algorithm (WPA), oppositional wolf pack algorithm (OWPA), and chaotic wolf pack algorithm (CWPA), and three widely applied swarm intelligence algorithms (PSO, ABC, and ASFA) are more suitable for solving problems of high-dimensional function optimization. The improved method of the algorithm can be applied to other intelligent optimization algorithms. This algorithm can be applied to large-scale scheduling problem, photovoltaic power generation problem, multidimensional data clustering, and other problems.

However, practical problems are often more complex and changeable, and all methods cannot be perfect. Premature convergence still restricts the development of swarm intelligence algorithm. In the next step, we should focus on solving this problem to further improve the algorithm performance and efficiency. In addition, it is still the next key research direction to expand the application scope of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by military science project of National Social Science Foundation (2019-SKJJ-C-092), National Natural Science Foundation of China (Grant no. 61502534), Natural Science Foundation of Shanxi Province (No. 2020JQ-493), Military Equipment Research Project (WJ20191C080073-17 and WJ2020A020029), and Research Foundation of Armed Police Force Engineering University (Nos. WJY201922 and JLY2020084).

References

[1] X. Wen, “A dynamic scheduling method for logistics tasks oriented to intelligent manufacturing workshop,” *Journal of Mathematical Problems in Engineering*, vol. 2019, Article ID 7237459, 18 pages, 2019.

[2] Y. Tang and Y. Huang, “Framework for artificial intelligence analysis in large-scale power grids based on digital simulation,” *CSEE Journal of Power and Energy Systems*, vol. 4, no. 4, pp. 459–468, 2018.

[3] L. Yue and M. Hang, “Studies of sub-synchronous oscillations in large-scale wind farm integrated system,” *Iop Conference*, vol. 108, Article ID 052075, 2018.

[4] R. Patriarca and S. Hosseini, “Modeling and quantification of resilience in complex engineering systems,” *Complexity*, vol. 2019, Article ID 1038908, 2 pages, 2019.

[5] Y. Zhang, “A survey on optimal control and operation of integrated energy systems,” *Complexity*, vol. 2019, Article ID 9462518, 14 pages, 2019.

[6] A. Wadood, “Application of the jaya algorithm in solving the problem of the optimal coordination of overcurrent relays in single- and multi-loop distribution systems,” *Complexity*, Article ID 5876318, 13 pages, 2019.

[7] N. Ao, M. Zhao, Q. Li, S. Qu, and Z. Wu, “Network characteristics for neighborhood field algorithms,” *Neural Computing and Applications*, vol. 32, no. 16, pp. 12061–12078, 2020.

[8] R.-L. Tang, Wu, and Y.-J. Fang, “Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems,” *Soft Computing*, vol. 21, no. 16, pp. 4735–4754, 2017.

[9] Q. Weibiao, “Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function

optimization problems," *IEEE Access*, vol. 7, no. 1104, pp. 72–86, 2019.

[10] H. S. Wu, "New swarm intellingence algorithm—wolf pack algorithm," *Systems Engineering and Electronic Technology*, vol. 35, no. 11, pp. 2430–2438, 2013.

[11] J. Hu, "Swarm intelligence-based optimisation algorithms: an overview and future research issues," *International Journal of Automation and Control*, vol. 14, no. 5-6, pp. 656–693, 2020.

[12] J. Kennedy, "Particle swarm optimization," in *Proceedings of 1995 IEEE International Conference Neural Networks*, vol. 4, no. 8, pp. 1942–1948, Perth, Australia, 2011.

[13] P. Duan and Y. Ai, "Research on an improved ant colony optimization algorithm and its application," *International Journal of Hybrid Information Technology*, vol. 9, no. 4, pp. 223–234, 2016.

[14] J. Li, Q. Pan, and S. Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9353–9371, 2012.

[15] L. Chambers, "Practical handbook of genetic algorithms," *Organization for Security and Co-operation in Europe (OSCE)*, vol. 1, no. 491, p. 346, 1995.

[16] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.

[17] C. Gao, "Optimization of hydraulic turbine governor parameters based on WPA," *IOP Conference Series: Earth and Environmental Science*, vol. 108, no. 5, Article ID 052011, 2018.

[18] L.. Hongdan, R. Sun, and Q. Liu, "The tactics of ship collision avoidance based on Quantum-behaved Wolf Pack Algorithm," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 6, 2019.

[19] H. Li, P. Bai, and H. S. Wu, "Hybrid binary Wolf pack algorithm for the 0-1 multidimensional knapsack problem," *International Journal of Wireless and Mobile Computing*, vol. 12, no. 3, pp. 291–304, 2017.

[20] J. Hu, H. Wu, R. Zhan, and N. Li, "Hybrid integer-coded wolf pack algorithm for multiple-type flatcars loading problem," *Rail Transport Planning & Management*, Article ID 100201, 2020.

[21] Z. Xiu and W. Zhen-Hua, "Improved wolf pack algorithm based on tent chaotic mapping and levy flight," in *Proceedings of the 2017 International Conference on Robots & Intelligent System (ICRIS)*, pp. 165–169, IEEE Computer Society, Huaian, China, 2017.

[22] Y. Zhu, "A chaos wolf optimization algorithm with self-adaptive variable step-size," *Aip Advances*, vol. 7, no. 10, Article ID 105024, 2017.

[23] R. H. Wang, X. Li, and N. Li, "Wolf pack algorithm based on Guassian disturbance and chaotic initialization," *Computer Engineering and Design*, vol. 40, no. 10, pp. 2879–2884, 2019.

[24] H. Li and H. Wu, "An oppositional wolf pack algorithm for Parameter identification of the chaotic systems," *Optik*, vol. 127, no. 20, pp. 9853–9864, 2016.

[25] Y. J. Chiang, "Exponentially harmonic maps, Gauss maps and Gauss sections," *Mediterranean Journal of Mathematics*, vol. 16, no. 5, 2019.

[26] H. Gan, S. Xiao, Z. Zhang, S. Shan, and Y. Gao, "Chaotic compressive sampling matrix: where sensing architecture meets sinusoidal iterator," *Circuits, Systems, and Signal Processing*, vol. 39, no. 3, pp. 1581–1602, 2020.

[27] M. Li, X. Zheng, and Y. Zhang, "Detection of subtle differences in seismic amplitude using convergence rate of the logistic map," *Geophysics*, vol. 84, no. 1, 2019.

[28] X. Jun-Jie, "A smart wolf pack algorithm and its convergence analysis," *Control & Decision*, vol. 31, no. 12, pp. 2131–2139, 2016.

[29] D. B. Fogel, "Evolutionary algorithms in theory and practice," *Complexity*, vol. 2, no. 4, pp. 26-27, 1997.

[30] Y. W. Zhang, L. Wang, and Q. D. Wu, "Dynamic adaptation cuckoo search algorithm," *Control and Decision*, vol. 29, no. 4, pp. 617–622, 2014.

[31] O. Castillo and P. Melin, "Optimization of benchmark mathematical functions using the firefly algorithm with dynamic parameters," *Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics*, vol. 574, no. 1, pp. 81–89, 2015.

[32] Y. Baba and O. C. Ugweje, "Development and analysis of a modified artificial fish swarm algorithm," in *Proceedings of the 2017 13th International Conference on Electronics*, pp. 1–6, Computer and Computation(ICECCO), Abuja, Nigeria, 2017.

[33] R.. Wang and R. C. Purshouse, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2012.