An Improved Wild Horse Optimizer Incorporating Dual Weight Starvation Strategy and Randomized Convergence Factor

Xiao-Rui Zhao, Xiao-Hong Chen*, Jie-Sheng Wang

Abstract—Wild Horse Optimizer (WHO) is a swarm-based meta-heuristic algorithm inspired by animal behavior, which mainly imitates the decent behavior, grazing behavior, mating behavior and leadership dominance behavior of wild horses in nature in their lives for finding the optimal. The location update method of WHO is prone to low convergence accuracy, poor global search ability and local optimum problems. With the aim of balancing global searchability and exploitation performance, an improved WHO that incorporates a dual weight starvation strategy and a random convergence factor is proposed. In the exploration stage, the starvation strategy is inspired by the starvation characteristics of animals, so that the algorithm can continuously adjust the stallion position according to the starvation level and automatically adjust the distance between the stallion and the wild horse according to the changing adaptation value, which improves the global search performance and can jump out of the local optimum at the same time. In the exploitation stage, a convergence factor is added to help it jump out of the local optimum and continue to search for a better solution. The simulation experiment on 23 benchmark functions is to verify the effectiveness of the proposed algorithm being compared with Whale Optimization Algorithm (WOA), Moth-Flame Optimization (MFO), Rat Swarm Optimizer (RSO), Multi-Verse Optimizer (MVO), Gray Wolf Optimizer (GWO) and Artificial Bee Swarm Optimizer (ALO). Finally two real engineering design problems were solved. The simulation results show that the proposed SD3WHO has a strong seeking capability and optimization performance.

Index Terms—**wild horse optimizer, dual weight starvation strategy, convergence factor, function optimization, engineering optimization**

I. Introduction

THE growing complexity and hardness of real-world problems in latest decades has led to the need for more reliable optimization techniques to solve problems with stochastic, while they are able to estimate optimal solutions to diverse optimization problems [1]. Optimization refers to the selection of the best set of solutions from all solutions in solving a problem, and the objective function, decision variables, and constraints are the three important factors of optimization [2]. The process of optimizing a problem is to obtain the optimal value of the decision variables for the problem by solving the objective function for the optimal value. Some practical problems with high complexity and long computation time with unknown search space are a challenge for many algorithms. Usually, deterministic algorithms [3] are prone to obtain the same solution in solving a problem if the starting point and conditions are the same. This behavior tends to lead to local optimal stagnation, in which the algorithm is stuck in a local solution and cannot step out of it [4].

Meta-heuristic algorithms are becoming increasingly popular as one of the techniques designed to solve optimization problems with some flexibility and can be applied to many different types of problems. Compared with traditional optimization algorithms, the stochastic nature of meta-heuristic algorithms can effectively reduce the generation of local optimum, while allowing a large search of the space, and their search strategies are mainly exploration and exploitation strategies [5-6], and the ability to effectively balance these two strategies is one of the factors to measure the excellent performance of the algorithm. Most of their inspirations are simple, by modeling different natural concepts, such as animal behavior or evolution, physical phenomena.

Evolutionary algorithms are mainly inspired by the concept of natural evolution by. They simulate the basic laws in natural evolution, design basic models based on the laws of biological sciences and genetics, and use operators driven by biological behaviors such as natural selection, crossover and mutation [7]. Genetic algorithms (GA) [8] are one of the most representative evolutionary algorithms, which are driven by Darwinian evolutionary ideas. The optimization is based on the initial solution of the evolutionary algorithm, and iterative optimization is performed by iteratively repeating this pattern by continuously mutating and combining to renew the population generation after generation, where the better the individual is, the higher the chance of entering the new population. For example, the biogeography-based optimization algorithm (BBO) [9], which is inspired by bio-geography, investigates the effect of migration and mutation on different inhabitants or other species in bringing about the stabilization of the ecosystem under this influence.

Manuscript received March 22, 2024; revised June 18, 2024. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293), and Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137).

Xiao-Rui Zhao is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 15566281893@163.com).

Xiao-Hong Chen is a lecturer of School of Mechanical Engineering & Automation, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 086-0412-2538246; fax: 086-0412-2538244; e-mail: ascxh@126.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: wang_jiesheng@126.com).

Distinct from evolutionary algorithms are algorithms inspired by physical phenomena, who use physical rule sets to write algorithms that allow populations to be iteratively updated continuously in a specific search space. Examples are physical concepts such as gravity, inertial forces, electromagnetic forces, gravity, black holes, etc. The gravitational search algorithm (GSA) [10] is a typical example, which is designed based on the law of gravity and particle swarm interactions as a model, where in iteration after iteration, objects are mutually attracted by the gravitational force between them, and the gravitational force moves all objects toward the object with heavier mass, and the maximum mass represents the optimal solution in the true search space. There are also multiverse algorithms (MVO) [11], as well as atomic search optimization algorithms (ASO) [12], and Big Bang optimization algorithms (BBOC) [13], which are designed based on the concept of black and white holes and wormholes of multiverse theory.

The swarm intelligence optimization algorithm is mainly used to find the best by simulating the group behavior that animals have when they are in motion or hunting, and during the search process of this algorithm, information about the characteristics and locations of all animals is shared and the algorithm is able to preserve the information iterations of the search space. Compared to evolution-based algorithms, swarm intelligence optimization algorithms have very few operators, are very easy to implement, and are capable of keeping information throughout the searching space.

The most prevalent algorithm among swarm intelligence algorithms is particle swarm optimization (PSO) [14], which is on the basis of social behaviors inspired by birds, animals, etc. Each particle in this algorithm can move throughout the search space, and the particles use information about their historical best position and neighborhood best position to adjust their flight speed, and then gradually gather near the best particle and update the distance between their respective positions and the global best solution simultaneously [15-16]. There are other popular population intelligence-based techniques, such as the Whale Optimization Algorithm (WOA), which performs optimization by simulating the bubble net hunting method performed by humpback whales [17], the Moth Flame Optimizer (MFO), which performs optimization based on the path of the spiral flight of a moth's navigation mechanism [18], the Rat Swarm Optimizer (RSO), which performs optimization by simulating the behavior of a rat chasing and attacking its prey [19], Gray Wolf Optimizer (GWO) that employs hierarchical leadership behavior modeling and strategies of gray wolves during hunting [20], Ant Lion Optimizer (ALO) that performs optimization by simulating the behavior of ant lions preying on ants in nature [21], Ant Colony Optimization Algorithm (ACO) where each individual interacts indirectly through pheromone trajectories [22], and some other swarm-based algorithms. Such as Raccoon Optimization Algorithm (ROA) [23], Grasshopper Optimization Algorithm (GOA) [24], Crow Search Algorithm (CSA) [25], Tune Swarm Algorithm (TSA) [26], and Marine Predator Algorithm (MPA) [27].

Wild Horse Optimizer (WHO) is a new intelligent optimization algorithm proposed in 2021, which is inspired by the social life behavior of wild horses, and performs the optimization by simulating the life behavior of wild horse populations [28-30]. Wild horses usually live in herds, including a stallion and several mares and foals. Horses exhibit many behaviors such as grazing behavior, chasing behavior, dominance behavior, leadership behavior and mating behavior. One fascinating behavior that distinguishes horses from other animals is equine decency. Equine decency behavior allows the horse's foals to separate from the group and join other groups before they reach puberty. Such leaving is done to prevent the father from mating with his daughter or siblings. It is inspired by the decent behavior of horses, with a strong merit-seeking ability and a fast convergence rate. However, in the exploration stage, the wild horse optimizer is prone to make the global search range of the algorithm too large, resulting in a part of wild horses being too far away from the leader to get the optimal value.

To improve this situation, this paper proposes a starvation strategy incorporating double weights to balance exploration and exploitation, which to some extent can continuously change the position of the leader according to the hunger level of the horse, while continuously and dynamically adjusting the distance between the group members and the leader. The distance between the group members and the leader is constantly and dynamically adjusted, so that the group members can be stable near the leader. In the development stage, the search for the optimal position is very fine leading the algorithm to fall into local optimum easily.

In this paper, a random convergence factor is added to help the algorithm jump out of the local optimum. For the WHO (SD3WHO) that incorporates the dual weight starvation strategy and the random convergence factor, 23 benchmark test functions are used to verify the convergence effect by using the improved algorithm SD3WHO and seven intelligent optimization algorithms, WHO[31], WOA[17], MFO[18], RSO[19], MVO[11], GWO[20], and ALO[10]. In addition, two real engineering problems are solved and the performance of the original WHO is compared with the improved algorithm and other algorithms to verify the effectiveness of the SD3WHO. Specifically, the paper includes the following contends. Section 2 introduces the WHO; Section 3 introduces the improved WHO; Section 4 conducts simulation experiments and engineering optimization and Section 5 draws conclusions.

II. Wild Horse Optimizer

A. Population Initialization Stage

Wild horses leave their parents before puberty, with adult males joining a single group to seek mating opportunities and females choosing to join another family to prevent mating among relatives. Competition among groups of wild horses is required to gain access to water sources.

By dividing the initial population into groups and setting the number of populations to N, the number of stallion groups is $G = N \times PS$. PS is the percentage of stallions in the total population. Thus we can get the leader stallion (G) and the rest of the members $(N - G)$ are equally distributed among these groups.

B. Grazing Behavior

During grazing, other members of the population graze around the stallion in order to allow the wild horses to gather more. Eq. (1) was used to analog the grazing behavior of wild horses.

$$\overline{P}_{i,G}^{j} = 2Qcos(2\pi rQ) \times \left(Stallion^{j} - P_{i,G}^{j}\right) + Stallion^{j} \quad (1)$$

where, $P_{i,G}^{j}$ is the present position of the group members, $Stallion^{j}$ is the position of the stallion, Q is calculated by Eq. (5), r is a random number in the range [-2,2] , and the new location of the members of the population at the time of grazing is $\overline{P}_{i,G}^{j}$.

$$TDR = 1 - iter \times (\frac{1}{maxiter}) \quad (2)$$

$$C = \vec{r}_1 < TDR \quad (3)$$

$$IDX = (C == 0) \quad (4)$$

$$Q = r_2 \ominus IDX + \vec{r}_3 \ominus \ (\sim IDX) \quad (5)$$

where, $iter$ is the current iteration and $maxiter$ is the maximum number of iterations of the algorithm. TDR is the adaptive parameter whose value starts from 1 and decreases during the execution of Eq. (1). C in Eq. (2) is a vector consisting of 0, 1 equal to the problem dimension, $\vec{r}_1$ and $\vec{r}_3$ in Eq. (3) are random vectors uniformly distributed in the range $[0,1]$, while r_2 in Eq. (3) is a random number in the distribution. In Eq. (3), the random vector $\vec{r}_1$ of the *IDX* index needs to satisfy the condition$(C == 0)$ to return. Eq. (5) results in 0 at the end of the algorithm execution.

C. Mating Behavior

For horses, before reaching puberty, the foal will choose to leave the herd. And after reaching puberty, the foal will begin to search for a mate. During this process, males will usually choose to join a single herd, while females will choose to join another family herd. This behavior is a decent one for the horse, mainly to prevent the father from mating with his daughters or siblings. The cyclic pattern of horses leaving, mating, and reproducing would be repeated over and over again, which would also require simulating the mating behavior of horses. During the simulation, Eq. (6) and Eq. (7) uses the same cross operator as the mean cross operator.

$$P_{G,k}^{z} = Crossover(P_{G,i}^{m}, P_{G,j}^{n}) \quad (6)$$

$$rossover{=}Mean \quad (7)$$

where, $P_{G,k}^{z}$ is the foal from the *k*-th group horse position *z*. The *i*-th group $P_{G,i}^{m}$ foal m leaves the group and its mating with the horse n in position $P_{G,j}^{n}$ after puberty, and the mated horse joins the group k.

D. Leadership Team

Each team in the wild horse population competes for a limited water source, and after the leader of the team succeeds in grabbing it, no other team is allowed to use it, and the losing leader and his group members can only wait until the last successful team leaves before they can use the water source, or they can choose to move in the direction of other water sources that are more likely to be occupied. Based on this behavior of wild horses, Eq. (8)-(9) propose this model of leading teams.

$$\overline{Stallion_{G,i}} = \begin{cases} 2Qcos(2\pi rQ) \times (WH - Stallion_{G,i}) + WH & if \ \ r3 > 0.5 \\ 2Qcos(2\pi rQ) \times (WH - Stallion_{G,i}) - WH & if \ \ r3 \leq 0.5 \end{cases} \quad (8)$$

$$Stallion_{G,i} = \begin{cases} P_{G,i} & if\ cost(P_{G,j}) < cost(Stallion_{G,i}) \\ Stallion_{G,i} & if\ cost(P_{G,i}) < cost(Stallion_{G,i}) \end{cases} \quad (9)$$

where $\overline{Stallion_{G,i}}$ is the next position of the leader of the first group, $Stallion_{G,i}$ is the current position of the leader of the ith group, WH is the position of the water source, and Q is the computed adaptive mechanism proposed by Eq. (6). r is a uniform random number in the range [−2，2].

III. WHO with Dual Weight Starvation Strategy and Randomized Convergence Factor

A. Dual Weight Starvation Strategy

In nature, starvation is one of the most critical reasons for determining the behavior of animals in their lives. When food is abundant, a variety of external stimuli and competing demands constantly affect the quality of life of animals. However, when food is scarce, animals choose to search for food rather than choose to cope with stimuli and mutual competition unrelated to food, when the pursuit of food sources becomes their top priority [28]. To prevent food shortages, animals regularly search for food, they constantly hover around it, and they snatch it through competitive and defensive behaviors [29]. Smart and powerful animals have more advantage in finding food, while the survival space of weak animals is decreasing time and time again, and every wrong decision may lead to the death of an individual or even the extinction of the whole species[30]. Therefore, when food sources are limited, starving animals adopt various strategies to cope with and win the situation, and animals with different levels of starvation and different thirst for food will adopt different coping strategies.

Wild horses are a population with social behavior in their search activities and, like other animals, their behavior is often influenced by hunger in their daily lives and constantly changes. Hunger is one of the most critical influences in the life of wild horses, and their search activity is directly proportional to the level of hunger. Based on the effects of hunger on animals, the mathematical model $W(i)$ simulates the hunger characteristics of individuals in the searching wild horse population, representing the effect of hunger on the activity range of each individual. One of the equations related to hunger $Stave(i)$ is defined as follows:

$$Stave(i) = \begin{cases} 0, & F(i) == BF \\ stave(i) + S, & F(i) != BF \end{cases} \quad (10)$$

$$S = \begin{cases} SL \times (1 + r_4), & ST < SL \\ ST, & ST \geq SL \end{cases} \quad (11)$$

where, $Stave(i)$ represents the hunger of the wild horses and $SStave$ represents the total hunger of the whole wild horse population, which is short for sum(stave). $F(i)$ is the fitness value of each individual in the current iteration, BF is the best fitness value in the current iteration, and the hunger of the best individual in the whole population is set to 0. For the other individuals, a new hunger value (S) is added to the initial hunger is increased, from which we can be sure that for different individuals, his hunger value (S) is different. In order to make the performance of the starvation strategy superior, we require the starvation value (S) to take the maximum of ST and SL, when the lower bound of S is the value around SL, which is set to 100 in this paper. r_4 is a random number in the range $[0,1]$.

$$ST = \frac{F(i) - BF}{WF - BF} \times r_5 \times 2 \times (UB - LB) \quad (12)$$

where, WF denotes the worst fitness value obtained in the current iteration, UB denotes the upper limit in the feature space, and LB denotes the lower limit in the feature space. $F(i) - BF$ represents the amount of food required for an individual not to be hungry, and in each iteration, the individual's hunger changes and the amount of food required changes. $WF - BF$ represents the total foraging capacity of the individual, and $\frac{F(i)-BF}{WF-BF}$ represents the hunger rate. r_5 is a random number in the range [0,1], and $r_5 \times 2$ represents the positive or negative effect of environmental factors on hunger.

$$W(i) = \left(1 - e^{-|stave(i) - SStave|}\right) \times r_6 \times 2 \quad (13)$$

where, r_6 is a random number in the range [0,1].

$$P_{i,G}^{j} = W(i) \times \left(Stallion^{j} - P_{i,G}^{j}\right) + ST. * Stallion^{j} \quad (14)$$

Taking the possible effects of starvation on wild horses as inspiration, the design proposes a starvation value S and a starvation weight $W(i)$. The upper limit of the starvation value depends on the magnitude of the starvation rate $\frac{F(i)-BF}{WF-BF}$ in ST, which means that the greater the starvation of horses in the whole population, the greater the effect on the population.

$W(i)$ is an exponential mathematical model, which performs a strong global search in the early stage of the algorithm, and adjusts the position of the stallion according to the effect of the hunger value ST on the stallion as iterations continue, and automatically adjusts the distance between the group members and the stallion depending on the change of the fitness value. The search step length of the algorithm is reduced compared with the original algorithm, which can effectively avoid the situation that the wild horses are far away from the stallion due to the distance between them. It can quickly locate the approximate position of the global optimal solution in the search space and ensure that the members of the group are not far away from the leader. Review the fitness value of the group members after each adjustment, and continuously adjust the position of the stallion according to the changes. On this basis, the algorithm can effectively balance exploration performance and exploitation performance to ensure that the algorithm can perform sophisticated search near the optimal solution while improving the global search capability.

B. Randomized Convergence Factor

To further enhance the exploration capability and convergence precision of the algorithm, a convergence factor D is introduced in the development phase. In the early stage of the search, the overall convergence factor is larger, and the algorithm focuses more on global optimization, but due to the presence of random perturbations, the convergence factor has the opportunity to get smaller values in some of the optimization-seeking iterations, which allows the algorithm to perform local optimization, and this strategy can improve the convergence speed. At the later stage of the search, the convergence factor is smaller overall and the algorithm focuses on local optimization. Again, due to the random perturbation, the convergence factor takes larger values in some of the optimization-seeking iterations, and the algorithm can search for optimization in a larger space, which helps to jump out of the local optimum. In this paper, the stochastic convergence factor from the five manifestations of cosine function, sine function, tangent function, power function and exponential function are proposed as shown in Eqs. (15)-(19), and the improved position update formula is shown in Eq. (20).

The random convergence factor in cosine functional form is as follows:

$$D_1 = 9 * (1 - \cos\left(\frac{1}{maxiter} - 0.35 * \pi\right)) * (rand - 0.5) \quad (15)$$

The random convergence factor in the form of a sine function is as follows:

$$D_2 = 4 * (1 + \sin\left(\frac{iter}{maxiter} + \pi\right)) * (rand - 0.5) \quad (16)$$

The stochastic convergence factor in the form of a tangent function is as follows:

$$D_3 = 2 * (2 - \tan(iter/maxiter)) * (rand - 0.5) \quad (17)$$

The stochastic convergence factor in power function form is as follows:

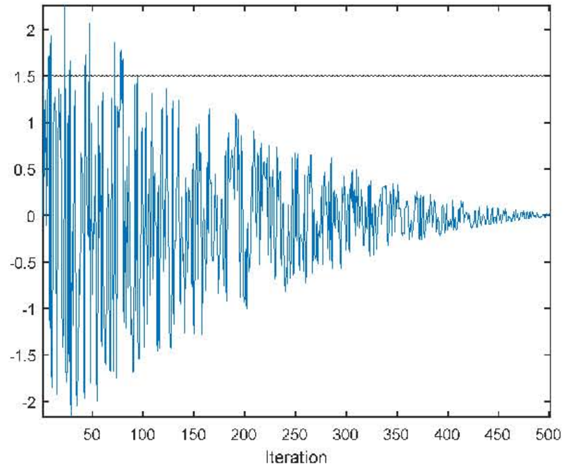
$$D_4 = 4 * (1 - (\frac{1}{maxiter})^3) * (rand - 0.5) \quad (18)$$

The stochastic convergence factor in exponential functional form is as follows:

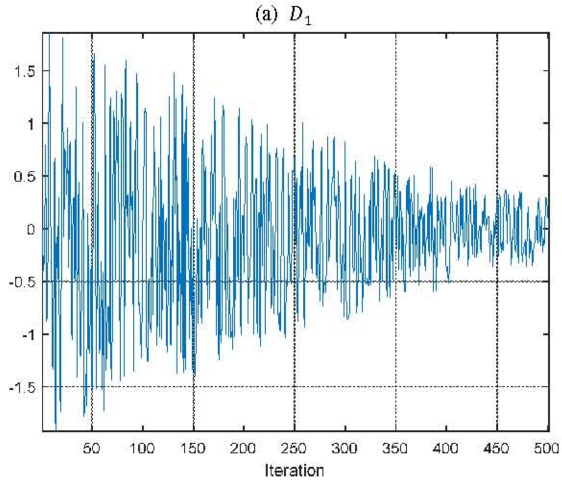
$$D_5 = 4 * (2 - e^{\left(\frac{0.7}{maxiter}\right)}) * (rand - 0.5) \quad (19)$$

$$\overline{Stallion_{G,i}} = \begin{cases} D * (2Qcos(2\pi rQ) \times (WH - Stallion_{G,i}) + WH) & if \ \ r3 > 0.5 \\ D * (2Qcos(2\pi rQ) \times (WH - Stallion_{G,i}) - WH) & if \ \ r3 \leq 0.5 \end{cases} \quad (20)$$

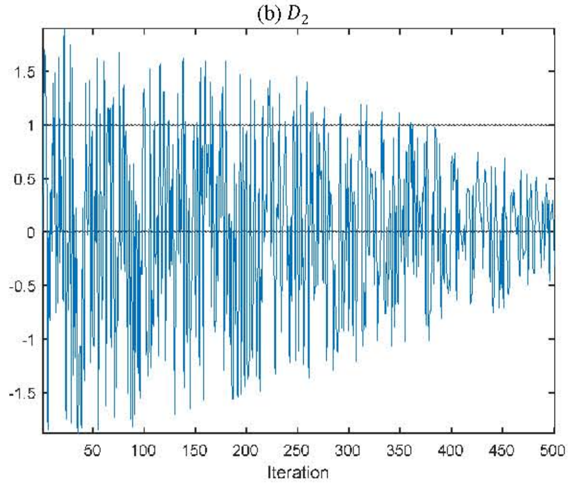
where, dim is the dimension of the current algorithm, $iter$ is the current iteration value, and $maxiter$ is the maximum iteration value. The movement trend of the stochastic convergence factor with increasing number of iterations is shown in Fig. 1.



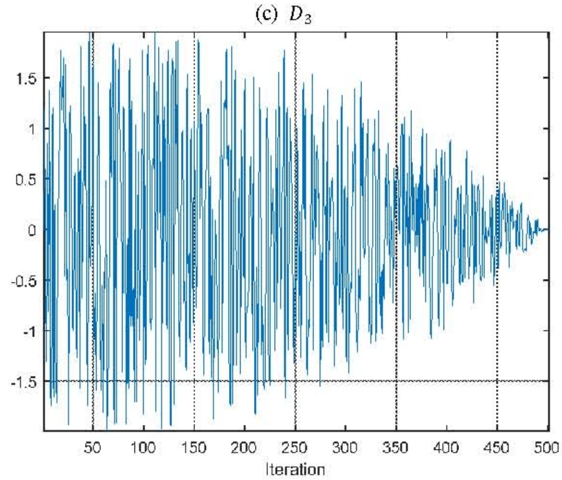
(a) D_1



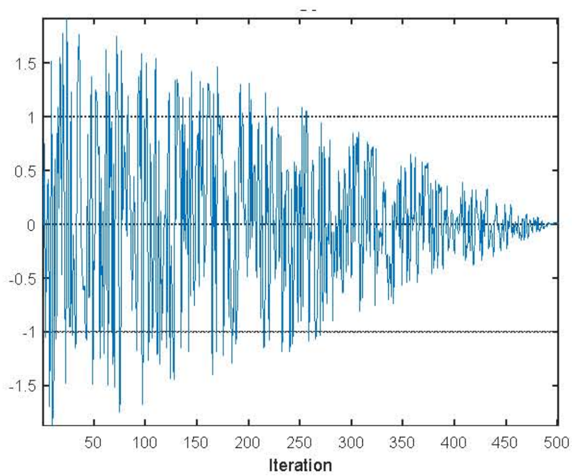
(b) D_2



(c) D_3



(d) D_4



(e) D_5

Fig. 1 Perturbation trend of random convergence factor D.

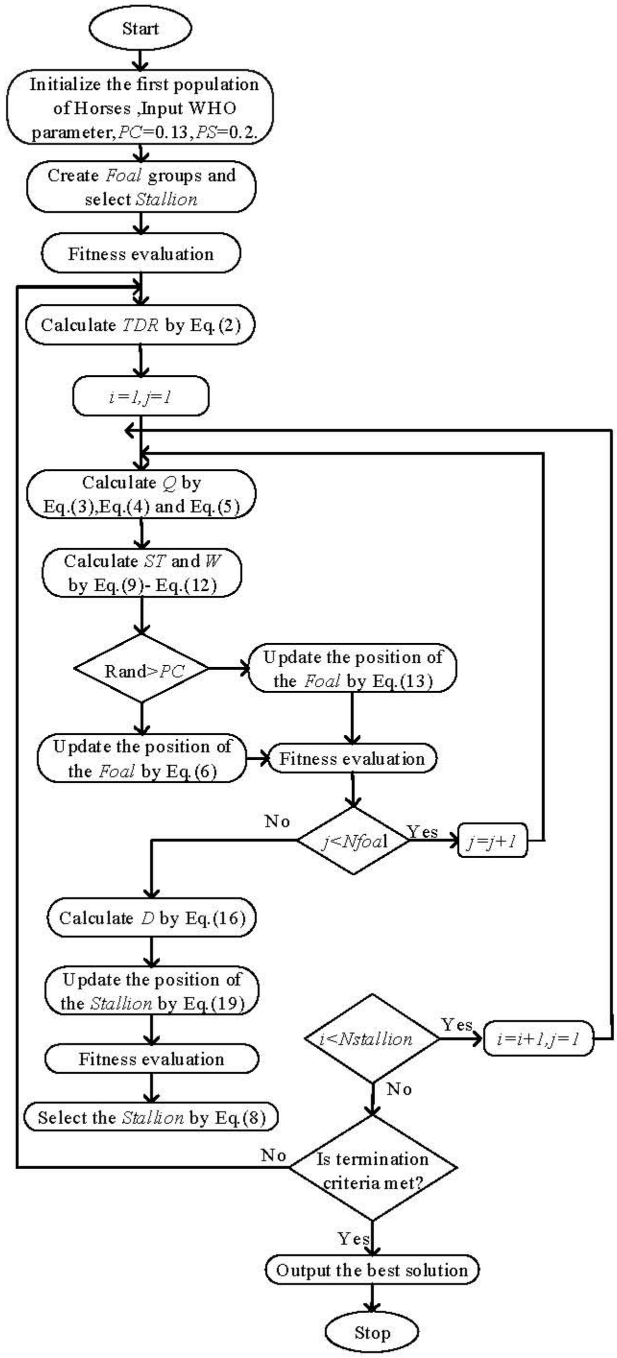


Fig. 2 Flowchart of WHO with dual weight starvation strategy and random convergence factor.

IV. Simulation Experiment and Result Analysis

Due to evaluate the optimization ability of WHO with double-weighted starvation strategy and random convergence factor and explore the optimization space of the algorithm, it is tested and compared with seven algorithms, namely, ALO[10], MVO [11],WOA [17], MFO [18], RSO [19]，GWO [20]and WHO [31], .

Due to the randomness and instability of the algorithms, for the sake of realistic and fair experiments, each group of experiments was chosen to run 30 times independently, and the average value was obtained on this basis, and the maximum number of iterations was set to 500 and the number of populations was set to 30. To further verify the effectiveness of the algorithms, two practical engineering optimization problems were used to verify the optimization performance of the algorithms again in this experiment.

A. Test Functions

In order to test the proposed SD3WHO algorithm, 23 standard test functions were selected for this experiment, including single-peak function F_1~F_7, multi-peak function F_8~F_{13}, and fixed-dimension multi-peak functionF_{14}~F_{23}. The 23 test functions are shown in Table I. No. denotes the function serial number, Function represents the function expression, dim means dimension, Range signifies the upper and lower limits, and $f_{\min}$ means the minimum value.

B. Performance Comparison and Result Analysis of Improved WHO

The performance of the algorithm was tested using 23 test functions for the six improvement schemes proposed by WHO. Due to the randomness of the algorithms, in order to make the experimental results more accurate, the algorithms were run 30 times each time and then the average of all the results was taken, and the maximum number of iterations of all the algorithms was set to 500, and 30 populations were set at the same time. The convergence curves of the six improved algorithms are compared with the convergence curves of the original WHO as shown in Fig.3, and the calculation results in the form of statistics of mean and variance are shown in Table II.

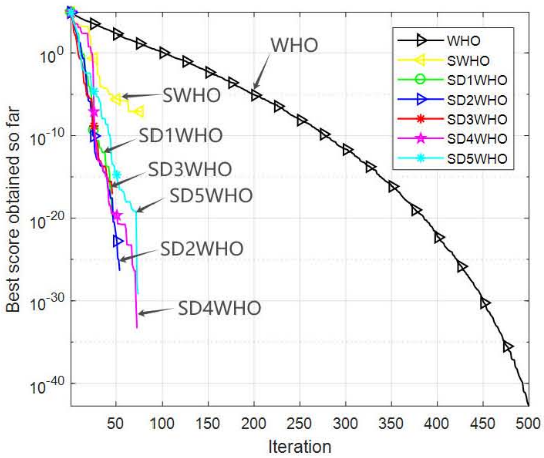
It is clear from Fig. 3 and Table II that the convergence effect and the average best-fit value of the improved algorithms are better than WHO for most of the tested functions, especially SD3WHO, which incorporates the dual weight starvation strategy and the random convergence factor, has a better finding ability than WHO for most of the tested functions, both in terms of convergence speed and mean, variance and most optimum. For the single-peaked test functions F1-F7, the mean, variance and most values of all improved algorithms in F1-F4 functions (including SWHO, SD1WHO, SD2WHO, SD3WHO, SD4WHO, and SD5WHO) are optimal, the mean and variance of F5-F7 are optimal, and the F5 and F7 have the best most values.

For the multi-peak test functions F8-F13, the best values were found for F8 and F9, and the best means and variances were found for the F9, F10 and F12, while the best mean value was found for F13. For the fixed-dimensional multi-peak test functions F14 to F23, the mean and variance of F15 and F22 are the best. From Fig. 3, it can be seen that SD3WHO has the fastest convergence speed. Most of the improved algorithms are more competitive than the original WHO, and the improved algorithms are more advantageous in terms of convergence speed and convergence accuracy, and basically find the optimal solution faster. From the comprehensive view of all the above algorithms, the effect of SD3WHO is the most obvious, the improved algorithm can improve the algorithm's global search ability, and can reduce the possibility of its falling into the local optimum, in improving the algorithm's local search ability as well as to improve is the exploration accuracy is also very advantageous.

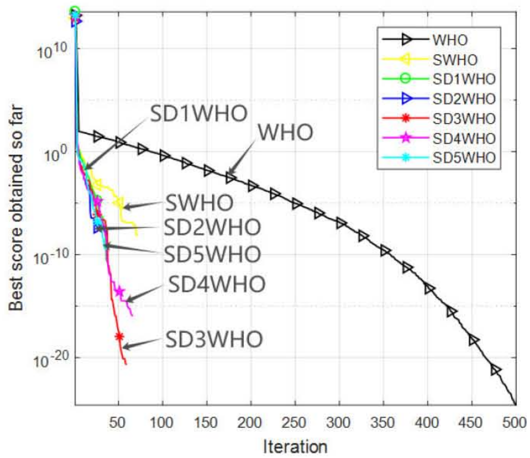
Table I. Description of the 23 benchmark functions

No.	Function	dim	Range	$f_{\min}$
F_1	$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
F_2	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
F_3	$f_3(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	30	[-100,100]	0
F_4	$f_4(x)=max_i\{\lvert x_i\rvert, 1 \ll i \ll n\}$	30	[-100,100]	0
F_5	$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30,30]	0
F_6	$f_6(x)=\sum_{i=1}^{n}([x_i+0.5]^2)$	30	[-100,100]	0
F_7	$f_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	30	[-1.28,1.28]	0
F_8	$f_8(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.982
F_9	$f_9(x)=[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12,5.12	0

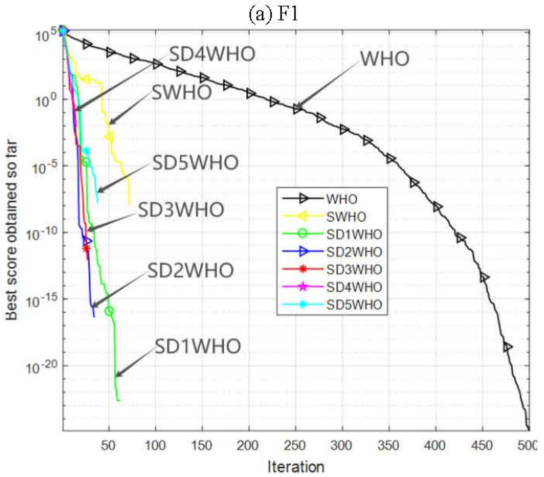
F_{10}	$f_{10}(x)=-20e^{-0.2\sqrt{\frac{\sum_{i=1}^{n}x_i}{n}}}-e^{\frac{\sum_{i=1}^{n}\cos(2\pi x_i)}{n}}+20+e$	30	[-32,32]	0
F_{11}	$f_{11}(x)=\frac{\sum_{i=1}^{n}x_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
F_{12}	$f_{12}(x)=\frac{\pi}{n}\{10\sin(ay_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})]+(y_n-1)^2+\sum_{i=1}^{n}\mu(x_i,10,100,4)\}$	30	[-50,50]	0
F_{13}	$f_{13}(x)=0.1\{10\sin^2(3\pi x_i)+\sum_{i=1}^{n}(x_i-1)^2[1+sin^2(3\pi x_i+1)]+(x_n-1)^2[1+sin^2(2\pi x_n)]+\sum_{i=1}^{n}\mu(x_i,5,100,4)\}$	30	[-50,50]	0
F_{14}	$f_{14}(x)=(\frac{1}{5000}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i+a_{ij})^6})^{-1}$	2	[-65,65]	1
F_{15}	$f_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	4	[-5,5]	0.00031
F_{16}	$f_{16}(x)=4x_i^2-2.1x_1^4+\frac{x_1^6}{3}+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
F_{17}	$f_{17}(x)=(x_2-\frac{5.1x_1^2}{4\pi^2}+\frac{5x_1}{\pi}-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5,10]	0.398
F_{18}	$f_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2])$	2	[-2,2]	3
F_{19}	$f_{19}(x)=-\sum_{i=1}^{4}c_ie^{-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2}$	3	[1,3]	-3.86
F_{20}	$f_{20}(x)=-\sum_{i=1}^{4}c_ie^{-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2}$	6	[0,1]	-3.32
F_{21}	$f_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
F_{22}	$f_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
F_{23}	$f_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	10	[0,10]	-10.5363



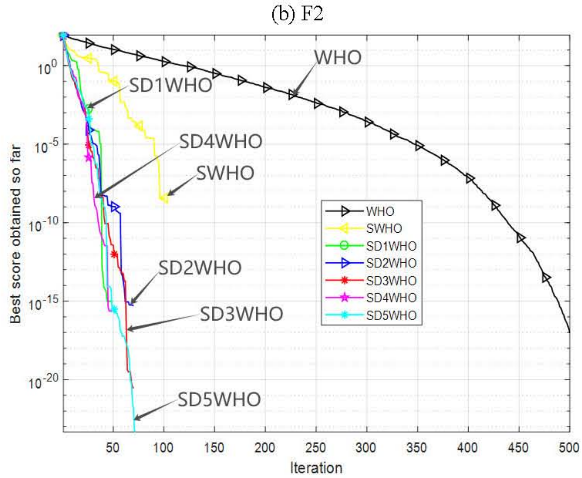
(a) F1



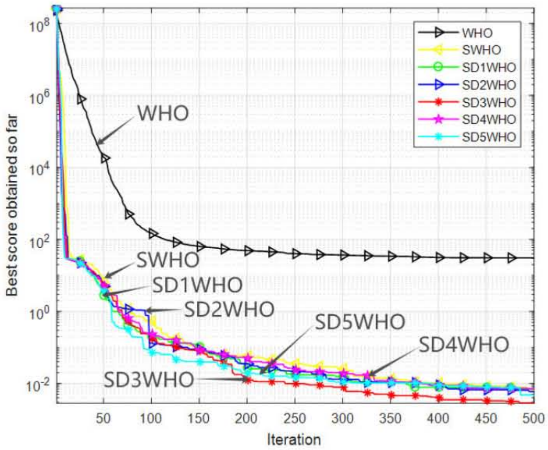
(b) F2



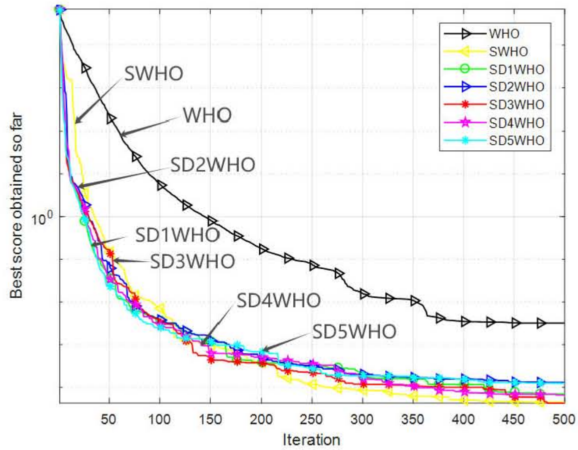
(c) F3



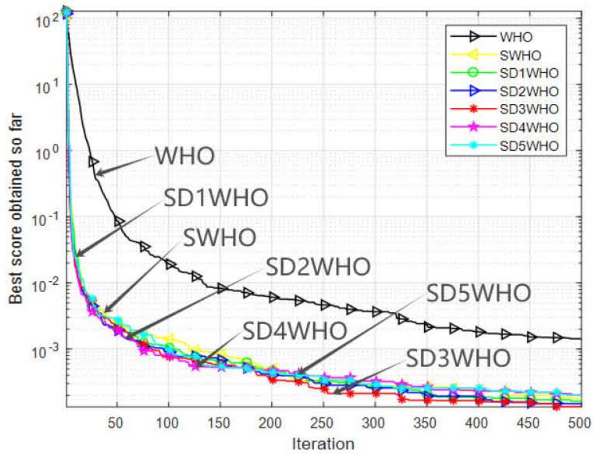
(d) F4



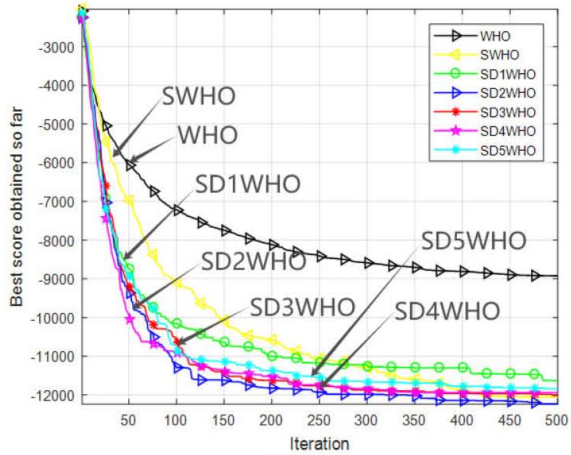
(e) F5



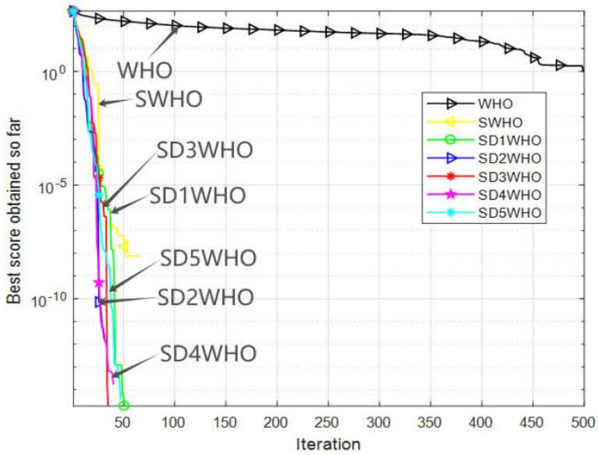
(f) F6



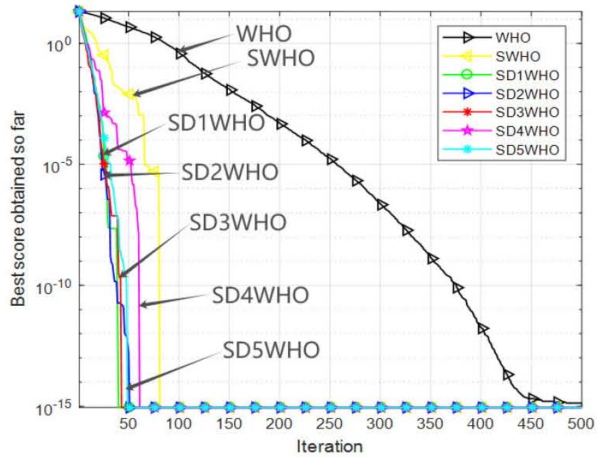
(g) F7



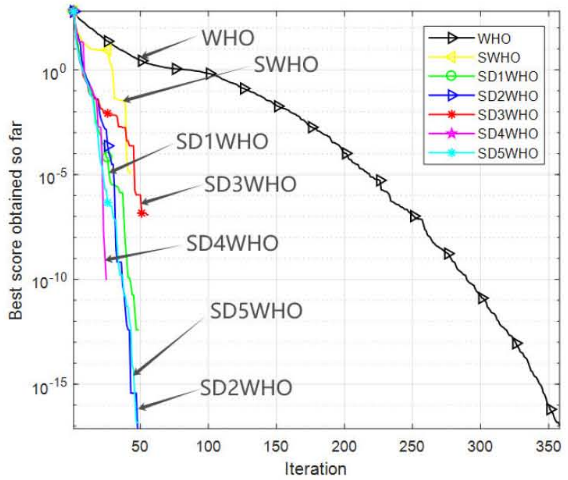
(h) F8



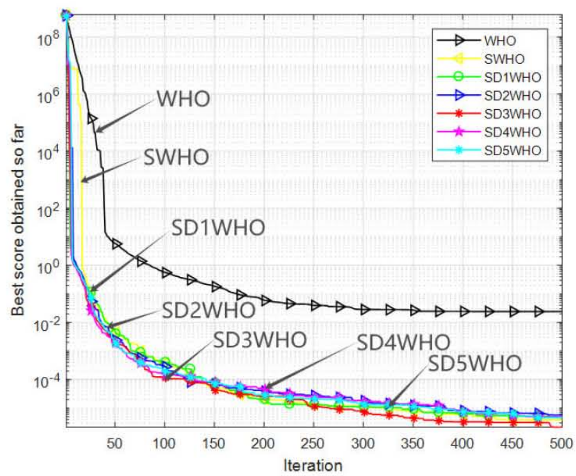
(i) F9



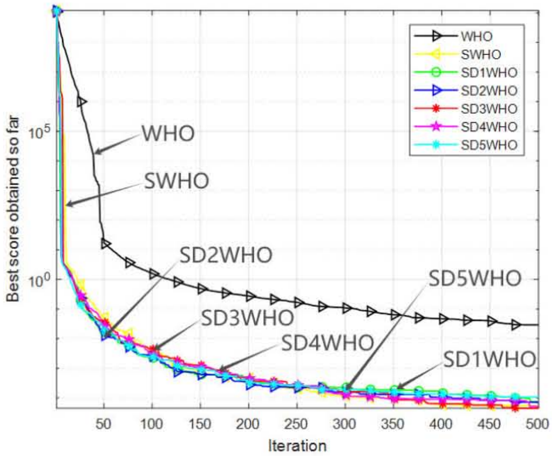
(j) F10



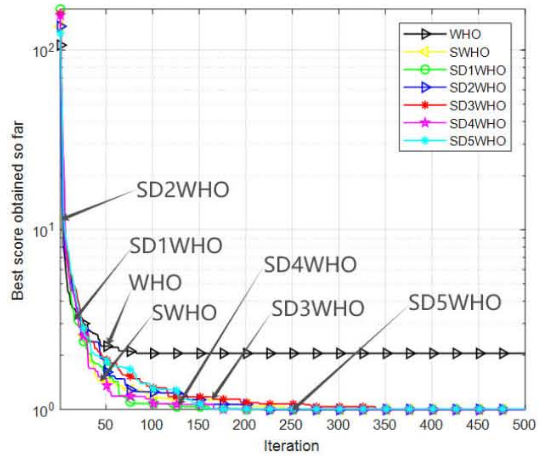
(k) F11



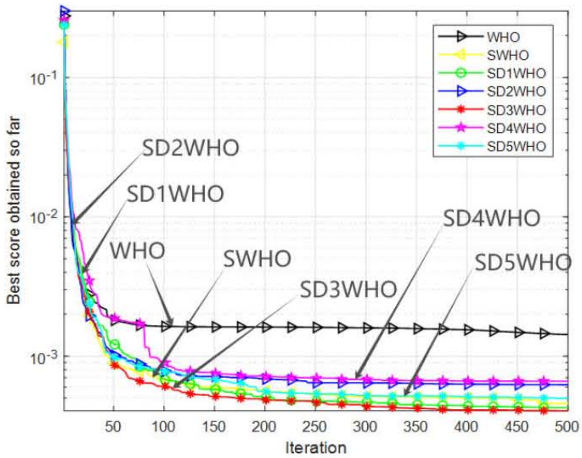
(l) F12



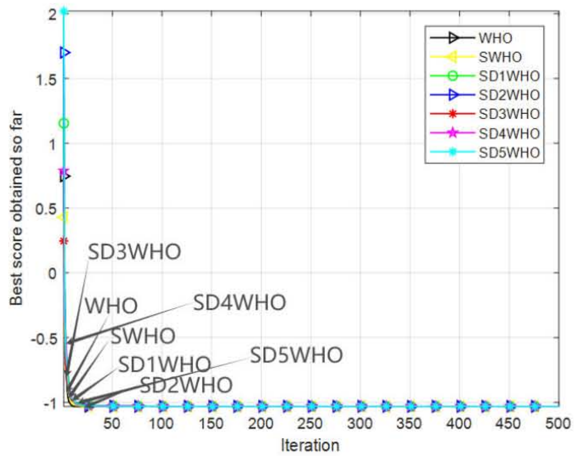
(m) F13



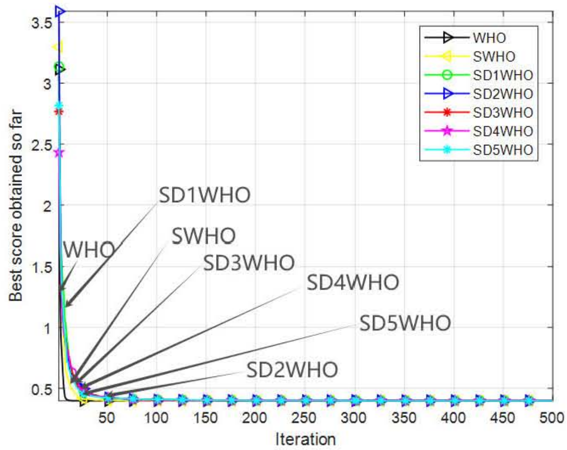
(n) F14



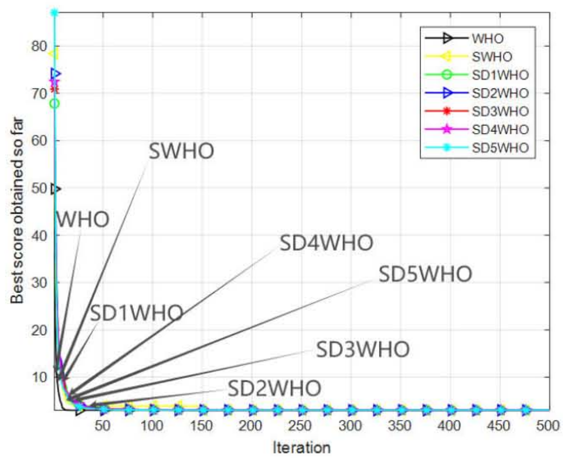
(o) F15



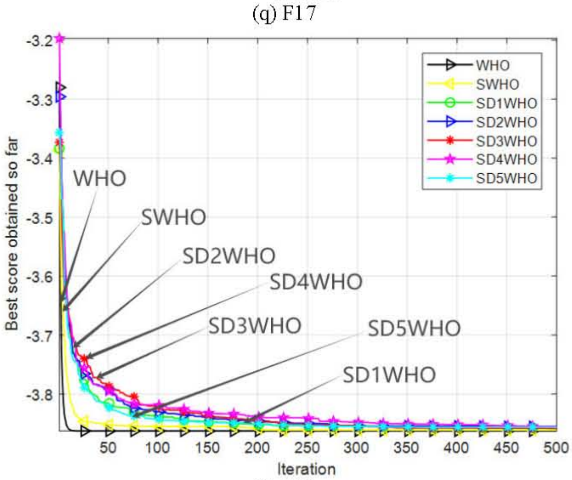
(p) F16



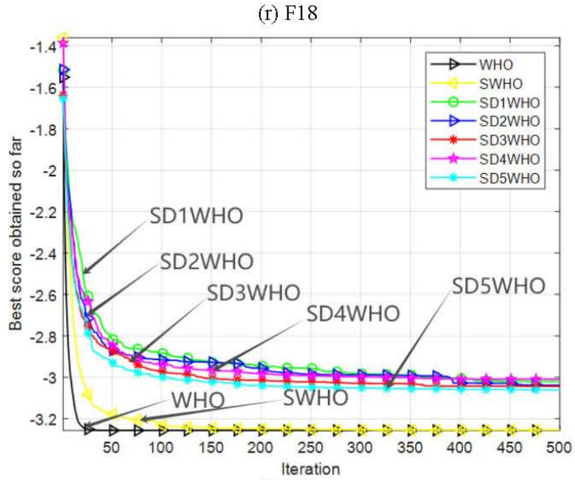
(q) F17



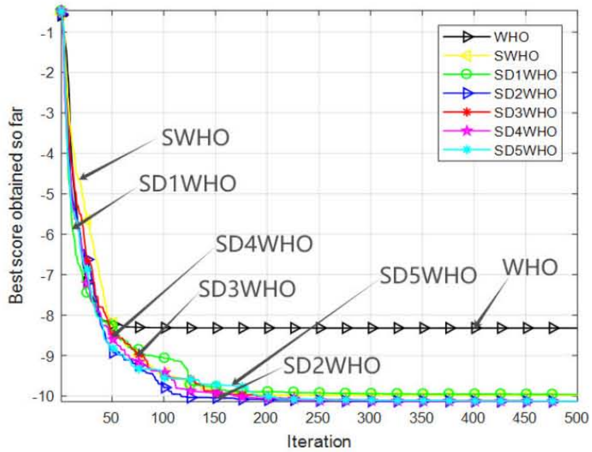
(r) F18



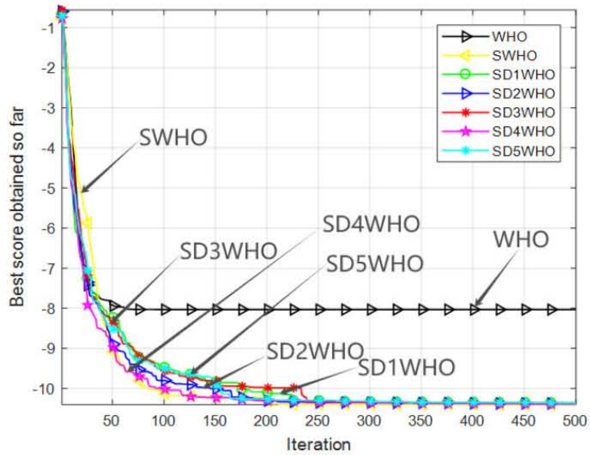
(s) F19



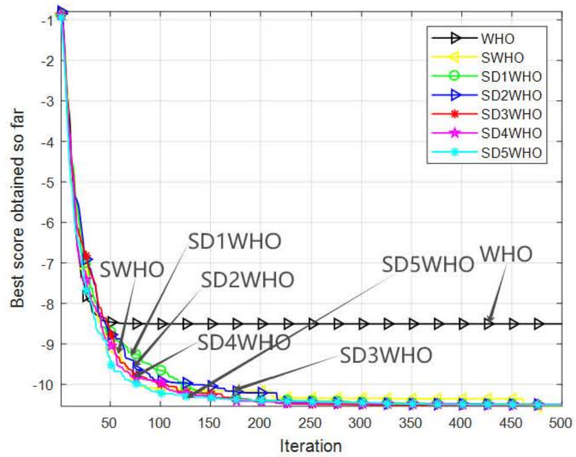
(t) F20



(u) F21



(v) F22



(w) F23

Fig. 3 Comparison of convergence plots on optimization functions.

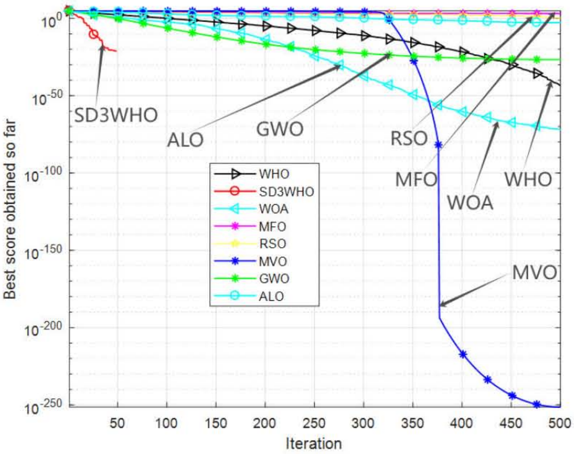
TABLE II. PERFORMANCE COMPARISON RESULTS OF THE 23 BENCHMARK FUNCTIONS OPTIMIZATION

F		WHO	SWHO	SD1WHO	SD2WHO	SD3WHO	SD4WHO	SD5WHO
	Ave	1.7205e-43	**0**	**0**	**0**	**0**	**0**	**0**
F_1	Std	7.1576e-43	**0**	**0**	**0**	**0**	**0**	**0**
	Best	1.9328-50	**0**	**0**	**0**	**0**	**0**	**0**
	Ave	2.4586e-25	**0**	**0**	**0**	**0**	**0**	**0**
F_2	Std	5.0028e-25	**0**	**0**	**0**	**0**	**0**	**0**
	Best	5.5889e-29	**0**	**0**	**0**	**0**	**0**	**0**
	Ave	9.4851e-26	**0**	**0**	**0**	**0**	**0**	**0**
F_3	Std	4.8173e-25	**0**	**0**	**0**	**0**	**0**	**0**
	Best	1.2255e-32	**0**	**0**	**0**	**0**	**0**	**0**
	Ave	1.1527e-16	**0**	**0**	**0**	**0**	**0**	**0**
F_4	Std	2.2502e-17	**0**	**0**	**0**	**0**	**0**	**0**
	Best	5.0634e-20	**0**	**0**	**0**	**0**	**0**	**0**
	Ave	30.8733	0.0064	0.0069	0.0060	**0.0028**	0.0070	0.0048
F_5	Std	78.1933	0.0128	0.0142	0.0060	**0.0036**	0.0120	0.0107
	Best	25.8206	8.7292e-06	1.8654e-04	1.1060e-05	**1.2294e-06**	1.7366e-04	1.4398e-05
	Ave	0.0032	4.4407e-05	6.9086e-05	1.2967e-04	**4.2400e-05**	6.5602e-05	1.2332e-04
F_6	Std	0.0083	6.8081e-05	6.7833e-05	4.0731e-04	**5.7715e-05**	6.9954e-05	1.9354e-04
	Best	1.3207e-07	5.8474e-07	9.9241e-07	1.3323e-07	9.5478e-07	**7.2406e-09**	3.6920e-07
	Ave	0.0014	1.7878e-04	1.5819e-04	1.4922e-04	**1.3345e-04**	2.0084e-04	1.9903e-04
F_7	Std	0.0012	1.4150e-04	1.5868e-04	1.3073e-04	**1.1622e-04**	2.032e-04	2.0031e-04
	Best	1.2855e-04	4.3637e-06	**5.5619e-07**	5.7083e-06	**3.2798e-06**	1.4588e-05	5.2697e-06
F_8	Ave	-8.9255e+03	-1.2061e+04	-1.1634e+04	**-1.2236e+04**	-1.2017e+04	-1.1947e+04	-1.1838e+04

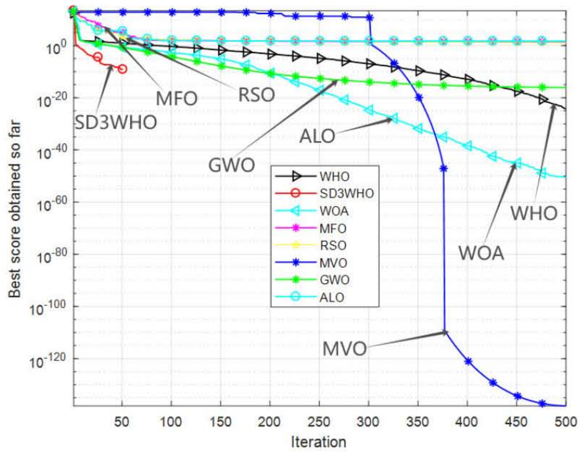
	Std	575.7809	848.2062	1.2892e+03	**571.228**	743.6616	1.0479e+03	1.0025e+03
	Best	-1.0140e+04	**-1.2569e+04**	**-1.2569e+04**	**-1.2569e+04**	**-1.2569e+04**	**-1.2569e+04**	**-1.2569e+04**
	Ave	1.0566	**0**	**0**	**0**	**0**	**0**	**0**
F_9	Std	5.7874	**0**	**0**	**0**	**0**	**0**	**0**
	Best	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Ave	1.3619e-15	**8.8818e-16**	**8.8818e-16**	**8.8818e-16**	**8.8818e-16**	**8.8818e-16**	**8.8818e-16**
F_{10}	Std	1.2283e-15	**0**	**0**	**0**	**0**	**0**	**0**
	Best	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16
	Ave	0	0	0	0	0	0	0
F_{11}	Std	0	0	0	0	0	0	0
	Best	0	0	0	0	0	0	0
	Ave	0.0242	3.8615e-06	4.8298e-06	5.6592e-06	**2.1479e-06**	4.8935e-06	4.5792e-06
F_{12}	Std	0.0931	5.7022e-06	6.4331e-06	8.1597e-06	**1.8592e-06**	9.1400e-06	6.8396e-06
	Best	7.6710e-08	**3.5374e-09**	5.2198e-08	1.5079e-07	5.9868e-07	7.9483e-09	**2.6485e-10**
	Ave	0.0291	5.4634e-05	7.4137e-05	7.1158e-05	**4.4459e-05**	5.1368e-05	1.0685e-04
F_{13}	Std	0.0335	9.0676e-05	7.9329e-05	9.5369e-05	7.3507e-05	**6.5796e-05**	2.3083e-04
	Best	5.2320e-05	**1.8767e-09**	1.1913e-07	3.8234e-07	1.1640e-07	2.8144e-07	1.1076e-06
	Ave	2.0504	**0.9980**	**0.9980**	**0.9980**	1.0010	0.9981	1.0055
F_{14}	Std	1.9912	**3.7201e-12**	2.2702e-05	2.0177e-08	0.0114	3.9357e-04	0.0411
	Best	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980
	Ave	0.0014	4.5769e-04	4.2689e-04	6.2398e-04	**4.0526e-04**	6.6108e-04	4.9634e-04
F_{15}	Std	0.0036	4.0066e-04	1.9866e-04	5.6060e-04	**1.0624e-04**	**6.2424e-04**	**3.5840e-04**
	Best	3.0768e-04	**3.0749e-04**	3.1972e-04	3.1426e-04	3.1807e-04	3.1232e-04	3.1144e-04
	Ave	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
F_{16}	Std	4.7908e-16	2.5561e-06	4.5395e-05	9.8244e-05	4.2831e-05	2.9795e-05	6.6193e-05
	Best	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Ave	0.3979	**0.3979**	0.3988	0.3986	0.3985	0.3986	0.3984
F_{17}	Std	3.2434e-16	4.0698e-07	8.3714e-04	8.6380e-04	5.3665e-04	7.6764e-04	7.3503e-04
	Best	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
	Ave	3.0000	**3.0000**	3.0022	3.0031	3.0026	3.0045	3.0026
F_{18}	Std	1.2775e-15	1.7772e-08	0.0026	0.0038	0.0036	0.0116	0.0043
	Best	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	Ave	-3.8628	-3.8612	-3.8572	-3.8554	-3.8584	-3.8551	-3.8564
F_{19}	Std	2.6823e-15	0.0030	0.0065	0.0117	0.0257	0.0201	0.0226
	Best	-3.8628	**-3.8628**	-3.8625	-3.8625	-3.8627	-3.8623	-3.8627
	Ave	-3.2564	-3.2550	-3.0212	-3.0383	-3.0437	-3.0098	-3.0611
F_{20}	Std	0.0635	0.0898	0.2121	0.1446	0.1686	0.2069	0.1400
	Best	-3.3220	-3.3220	-3.3125	-3.2337	-3.2884	-3.2402	-3.2695
	Ave	-8.1390	-9.9833	-9.9621	-10.1367	**-10.1394**	-10.1346	-10.1325
F_{21}	Std	3.1416	0.9308	0.9458	0.0280	**0.0193**	0.0390	0.0425
	Best	-10.1532	-10.1532	-10.1532	-10.1532	-10.1532	-10.1532	-10.1532
	Ave	-8.2366	**-10.4029**	-10.3700	-10.3764	-10.3883	-10.3883	-10.3624
F_{22}	Std	3.4468	8.1571e-05	0.0679	0.0558	0.0270	0.0322	0.0742
	Best	-10.4029	-10.4029	-10.4028	-10.4028	-10.4028	-10.4028	-10.4028
	Ave	-8.5025	**-10.5363**	-10.5100	-10.4013	-10.5189	-10.5029	-10.5078
F_{23}	Std	3.1994	**8.3390e-04**	0.0486	0.1010	0.0432	0.0670	0.0786
	Best	-10.5364	-10.5364	-10.5363	-10.5363	-10.5363	-10.5363	-10.5363

C. Performance Comparison of SD3WHO with Other Algorithms

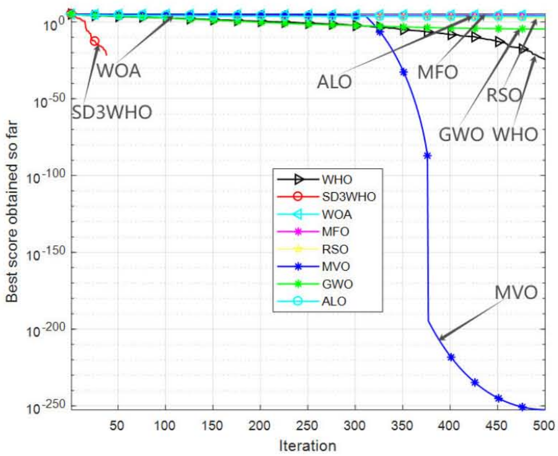
To more clearly show the advantage of the improved algorithm, the convergence plots of the SD3WHO were compared with seven algorithms, WHO[31], WOA[17], MFO[18], RSO[19], MVO[11], GWO[20], and ALO[10], as shown in Fig. 4. Table Ⅲ lists their optimal, mean and variance statistics. Due to the randomness of the algorithms, in order to make the experimental results more accurate, the algorithms were run 30 times each time and then the average of all the results was taken, and the maximum number of iterations of all the algorithms was set to 500, and 30 populations were set at the same time.From Fig. 4 and Table III, it can be seen that SD3WHO has the best performance with best fitness values compared to other algorithms. Functions F1-F7, F1-F4 and F6 have the best mean, variance and best fit values, F5 has the best mean and F7 has the best mean and variance. As can be seen in Fig. 4, the convergence speed advantage of SD3WHO is obvious. For functions F8-F13, F8-F13 have very superior rate of finding superiority, F9-F13 have the best mean, variance and optimum value, while F8 have the best mean and optimum value. Function F14-F23, F14, F15, F17, F21, F22 and F23 have the best mean, F15 has the best variance, and F21 and F23 have the best maxima, which can be seen in Fig. 4, and F14, F15, F21, F22 and F23 have the obvious convergence advantage. The results of the above experimental data show that SD3WHO, which integrates the dual-weight starvation strategy and the stochastic convergence factor, exhibits strong superiority in search performance, and is feasible and effective in improving the global search capability and improving the local search capability.



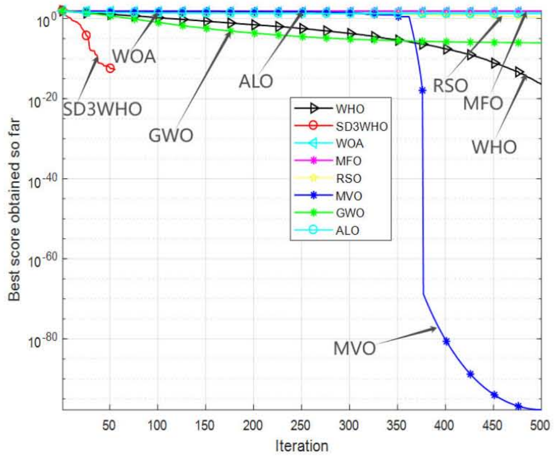
(a) F1



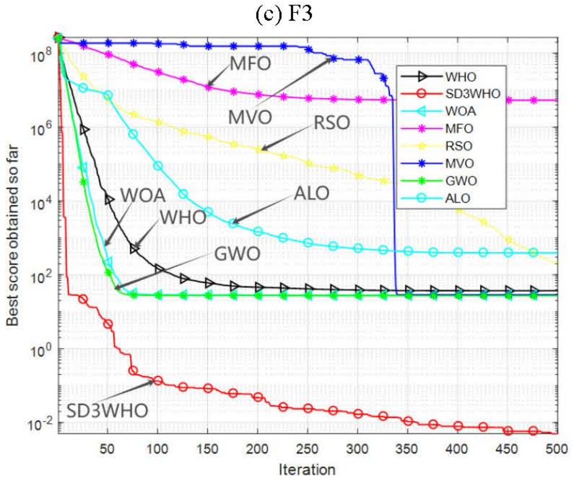
(b) F2



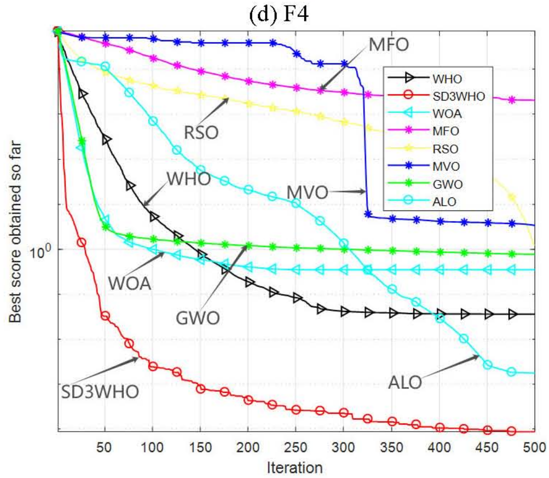
(c) F3



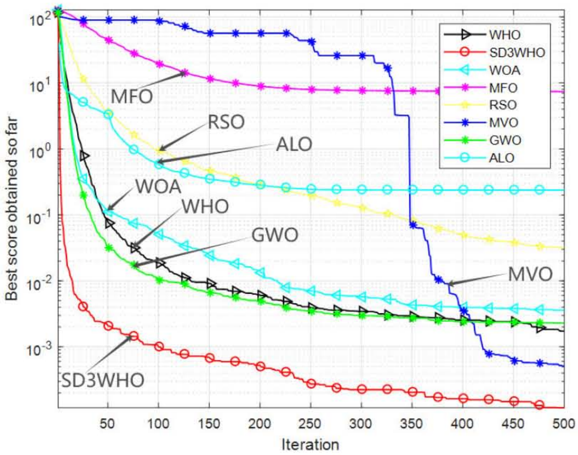
(d) F4



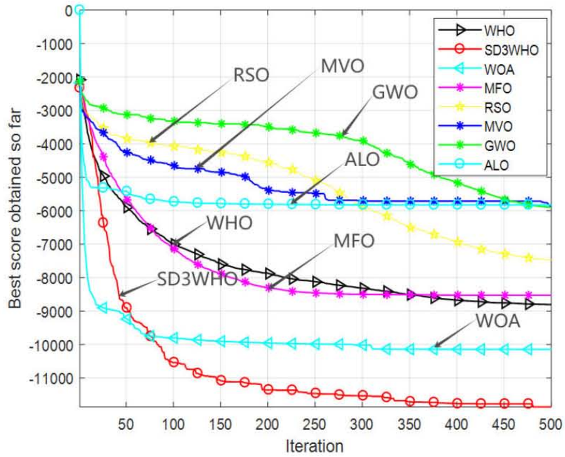
(e) F5



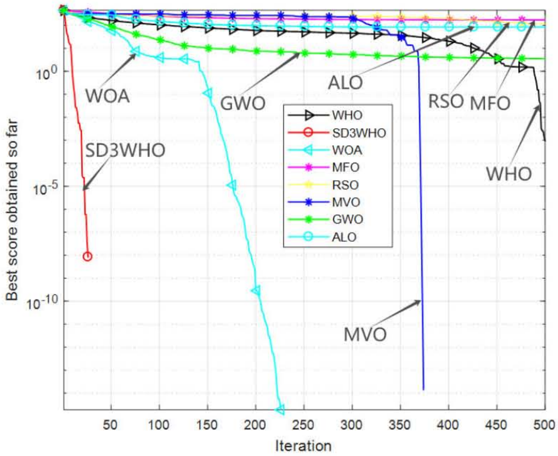
(f) F6



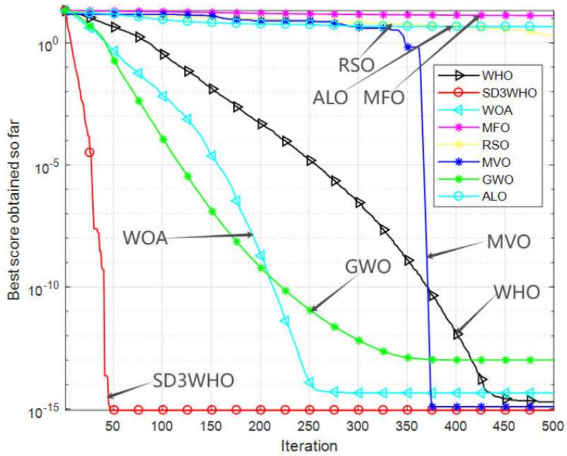
(g) F7



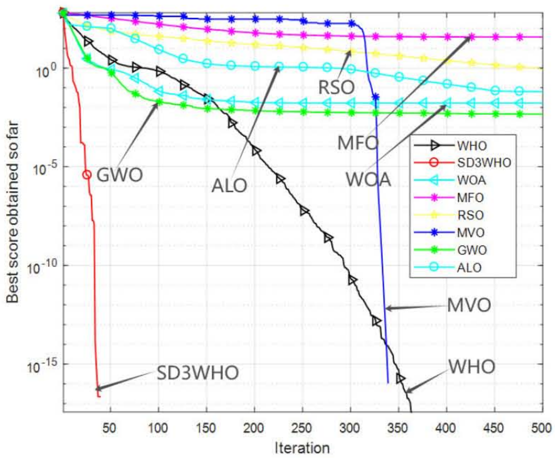
(h) F8



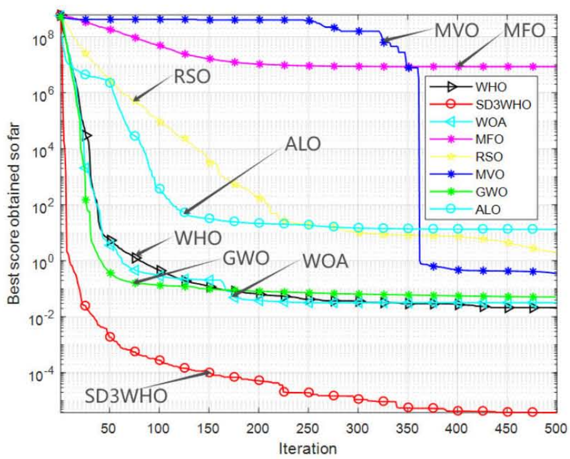
(i) F9



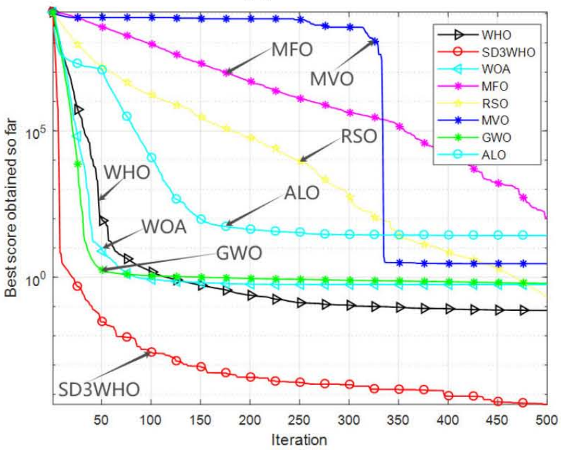
(j) F10



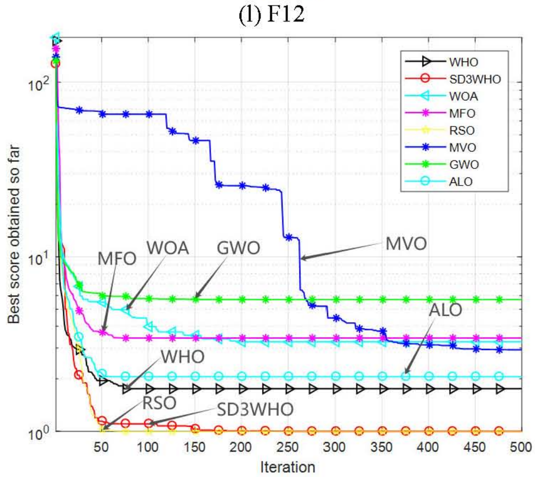
(k) F11



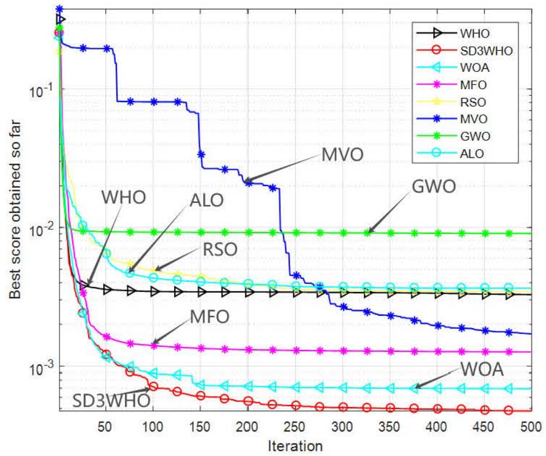
(l) F12



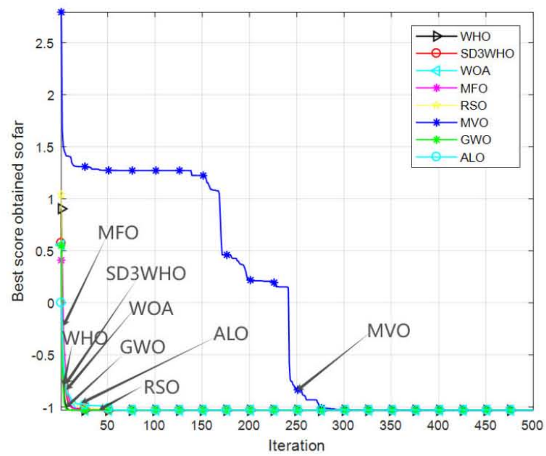
(m) F13



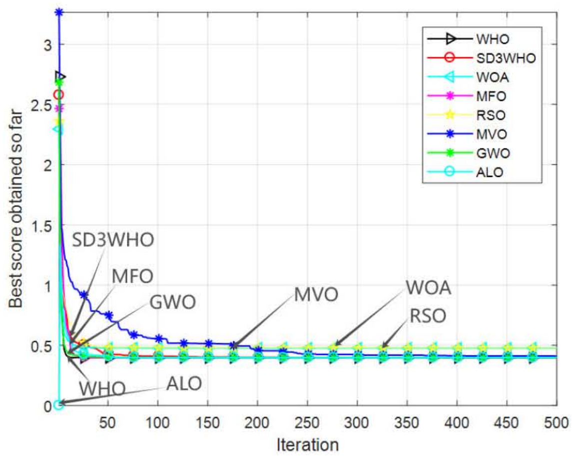
(n) F14



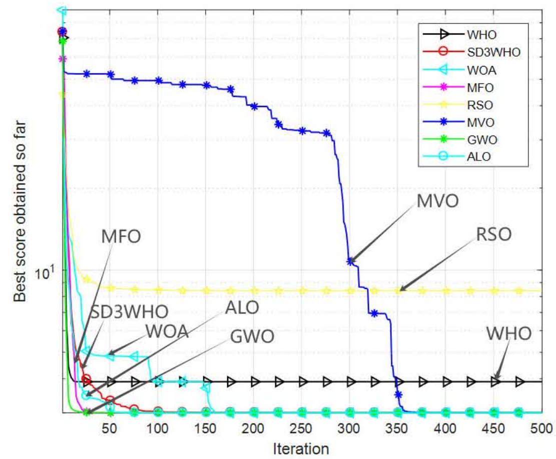
(o) F15



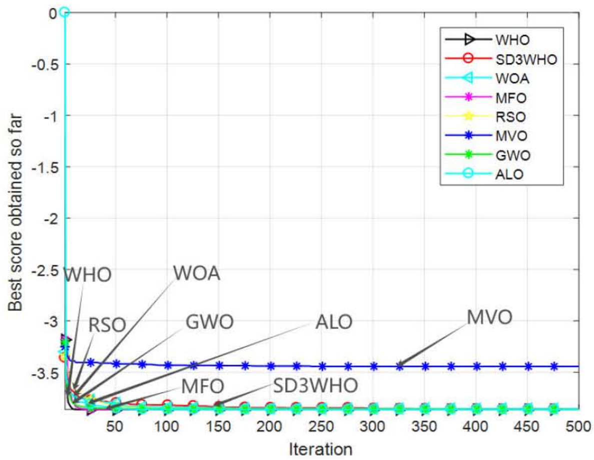
(p) F16



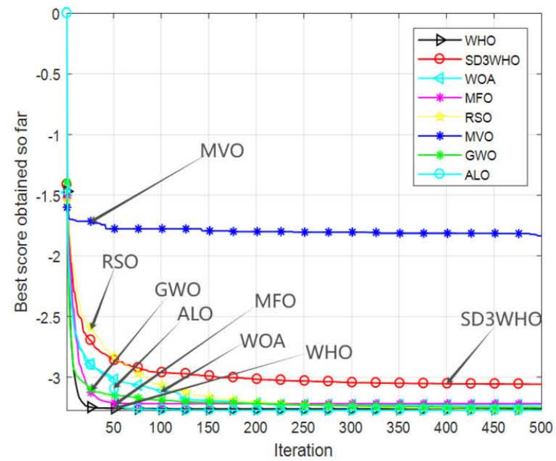
(q) F17



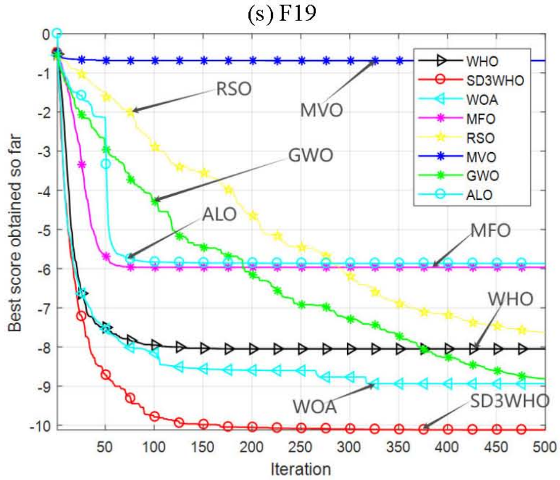
(r) F18



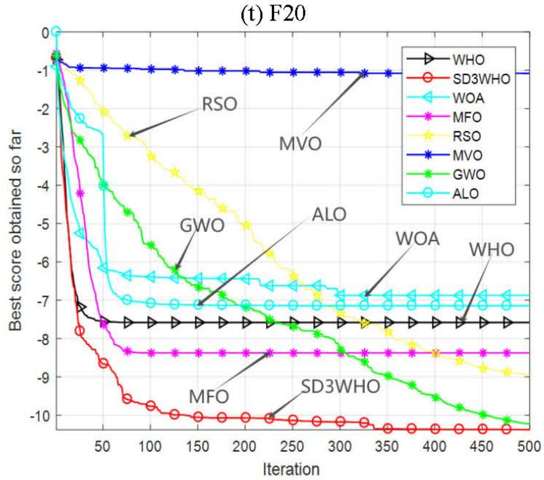
(s) F19



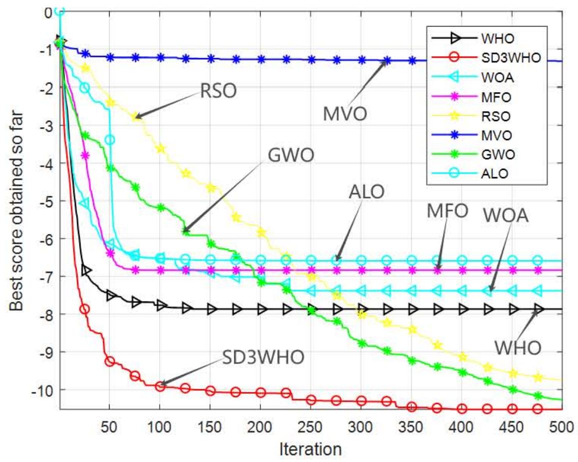
(t) F20



(u) F21



(v) F22



(w) F23

Fig. 4 Convergence comparison on optimization functions.

TABLE III. PERFORMANCE COMPARISON RESULTS OF THE 23 BENCHMARK FUNCTIONS OPTIMIZATION

F		WHO	FHPWHO	WOA	MFO	RSO	MVO	GWO	ALO
	Ave	5.4402e-44	**0**	1.1104e-72	1.3380e+03	3.7682e-252	1.3089	2.0827e-27	0.0017
F_1	Std	2.2657e-43	**0**	4.4417e-72	3.4562e+03	0	0.4125	4.2293e-27	0.0023
	Best	5.2641e-51	**0**	1.6372e-86	0.5309	0	0.5418	3.4540e-29	2.2307e-04
	Ave	5.6874e-25	**0**	3.0019e-51	32.1642	7.7560e-139	10.7858	9.4580e-17	46.0565
F_2	Std	1.8056e-24	**0**	1.6169e-50	20.4751	4.0371e-138	30.3807	7.2697e-17	47.5098
	Best	6.1167e-29	**0**	1.6338e-58	0.2939	0	0.3841	1.5618e-17	1.5557
	Ave	6.1876e-25	**0**	4.1117e+04	1.7287e+04	2.7970e-253	216.4454	2.8308e-05	3.5007e+03
F_3	Std	3.1534e-24	**0**	1.1079e+04	9.5112e+03	0	88.4535	8.3925e-05	1.5111e+03
	Best	2.2086e-31	**0**	9.8749e+03	3.7714e+03	0	81.7330	8.0779e-10	1.1172e+03
	Ave	3.7431e-17	**0**	42.6955	68.1003	1.9760e-98	2.1614	8.8722e-07	17.5487
F_4	Std	8.5250e-17	**0**	27.5762	9.0472	1.0823e-97	0.7296	8.3010e-07	5.1594
	Best	1.4700e-20	**0**	2.4229	40.7287	0	0.8783	8.9629e-08	6.5131
	Ave	36.9045	**0.0050**	28.0608	5.326e+06	28.8550	205.6537	27.2102	390.9225
F_5	Std	28.5278	0.0059	0.4453	2.0291e+07	0.1795	324.3178	0.6532	620.3567
	Best	25.9256	2.0596e-05	26.8996	290.0465	28.1718	36.2586	26.0830	21.8765
	Ave	0.0361	**8.7064e-05**	0.3523	2.0185e+03	3.4097	1.2413	0.7794	0.0018
F_6	Std	0.1766	**1.7049e-04**	0.1990	4.8574e+03	0.5442	0.2466	0.3912	0.0021
	Best	9.6352e-06	**1.0399e-06**	0.0927	0.3233	2.1291	0.6904	7.3690e-05	2.4439e-04
	Ave	0.0017	**1.1919e-04**	0.0036	7.3995	5.1464e-04	0.0320	0.0320	0.0023
F_7	Std	0.00122	**1.1166e-04**	0.0042	12.6061	4.0315e-04	0.0153	0.0013	0.0889
	Best	1.6366e-04	1.1415e-06	1.2002e-04	0.0672	4.5693e-05	0.0107	5.8706e-04	0.1267
	Ave	-8.8040+03	**-1.1867e+04**	-1.0145e+04	-8.5273e+03	-5.8968e+03	-7.4626e+03	-5.8945e+03	-5.8259e+03
F_8	Std	422.8624	959.4193	1.7401e+03	725.5669	911.2691	811.3362	975.44464	1.2204e+03
	Best	-9.7433e+03	**-1.2569e+04**	-1.2568e+04	-1.00097e+04	-6.9750e+03	-9.1820e+03	-7.1796e-+03	-1.1918e+04
	Ave	8.9404e-04	**0**	**0**	171.6365	**0**	118.8215	3.5400	83.5108
F_9	Std	0.0049	**0**	**0**	51.3716	0	28.0849	4.9810	21.3728
	Best	**0**	**0**	**0**	88.7238	0	60.5730	5.6843e-14	41.7888
	Ave	1.954e0-15	**8.8818e-16**	4.5593e-15	13.0138	1.2434e-15	1.8881	1.0155e-13	4.5984
F_{10}	Std	1.6559e-15	**0**	2.1847e-15	7.5234	1.0840e-15	0.6143	1.6142e-14	2.3889
	Best	8.8818e-16	**8.8818e-16**	8.8818e-16	1.1773	8.8818e-16	0.4643	7.5495e-14	2.1203
	Ave	0	**0**	0.0168	37.2234	0	0.8490	0.0046	0
F_{11}	Std	0	**0**	0.0641	55.7152	0	0.0982	0.0086	0.0332
	Best	0	**0**	0	0.6633	0	0.6422	0	0.0184
	Ave	0.0213	**3.7826e-06**	0.0320	8.5334e+06	0.3456	2.0295	0.0507	13.2602
F_{12}	Std	0.0788	**4.5764e-06**	0.0458	4.6739e+07	0.1316	0.8817	0.0253	5.4217
	Best	3.6006e-07	**2.9249e-08**	0.0062	3.1149	0.1284	0.4117	0.0071	4.2210

	Ave	0.0733	**4.5313e-05**	0.5637	95.1706	2.9015	0.2164	0.6199	26.5194
F_{13}	Std	0.1244	**5.9077e-05**	0.2870	394.7437	0.1763	0.2897	0.2249	18.5807
	Best	1.2289e-05	**6.1501e-08**	0.1543	4.4770	2.8005	0.0542	1.1385e-04	0.0037
	Ave	1.7525	**0.9980**	3.2569	3.4228	2.9337	**0.9980**	5.6873	2.0540
F_{14}	Std	2.0033	5.4074e-06	3.3078	3.1890	2.8028	3.5702e-11	4.8480	1.4655
	Best	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980
	Ave	0.0033	**4.7886e-04**	6.9431e-04	0.0013	0.0017	0.0034	0.0091	0.0036
F_{15}	Std	0.0068	**4.0240e-04**	4.0400e-04	0.0014	0.0037	0.0068	0.0101	0.0067
	Best	3.0749e-04	3.1122e-04	3.0790e-04	5.3912e-04	3.8268e-04	3.2079e-04	3.0788e-04	4.8882e-04
	Ave	-1.0316	-1.0316	-1.0316	-1.0316	-1.0315	-1.0316	-1.0316	-1.0316
F_{16}	Std	5.1334e-16	3.4614e-05	6.5053e-10	6.7752e-16	2.1259e-04	2.4876e-07	1.9974-08	1.5213e-13
	Best	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Ave	0.3979	**0.3979**	0.4777	0.3979	0.4110	04777	0.3979	0.3979
F_{17}	Std	0	6.5917e-04	0.4370	0	0.0144	0.4370	3.9391e-06	**3.8232e-14**
	Best	0.3979	0.3984	0.3979	0.3979	0.3980	0.3979	0.3979	0.3979
	Ave	3.9000	3.0023	3.0001	3.0000	3.0001	8.4000	3.0000	3.0000
F_{18}	Std	4.9295	0.0037	3.0492e-04	2.1630e-15	1.8773e-04	20.5504	4.6869e-05	9.6869e-13
	Best	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	Ave	-3.8628	-3.8580	-3.8540	-3.8625	-3.4432	-3.8628	-3.8612	-3.8628
F_{19}	Std	2.6684e-15	0.0043	0.0146	0.0014	0.3390	1.2608e-06	0.0028	1.1395e-12
	Best	-3.8628	-3.8624	-3.8628	-3.8628	-3.8551	-3.8628	-3.8628	-3.8628
	Ave	-3.2621	-3.0584	-3.2284	-3.2195	4-1.836	-3.2698	-3.2495	-3.2744
F_{20}	Std	0.0670	0.1767	0.1159	0.0695	0.4499	0.0607	0.0778	0.0593
	Best	-3.3220	-3.2986	-3.3214	-3.3220	-2.6389	-3.3220	-3.3220	-3.3220
	Ave	-8.0495	**-10.1186**	-8.9387	-5.9657	-0.6873	-7.6277	-8.8136	-5.8667
F_{21}	Std	2.8885	0.0730	2.1797	3.1763	0.2445	2.8087	2.7720	2.8044
	Best	-10.1532	**-10.1532**	-10.1531	-10.1532	-1.3926	-10.1531	-10.1528	-10.1532
	Ave	-7.5809	**-10.3783**	-6.8733	-8.3764	-1.0828	-8.9315	-10.2255	-7.1457
F_{22}	Std	3.3552	0.0494	3.2424	3.2034	0.7024	2.7705	0.9630	3.1648
	Best	-10.4029	-10.4028	-10.4001	-10.4029	-2.9915	-10.4029	-10.4028	-10.4029
	Ave	-7.8677	**-10.5240**	-7.3803	-6.8367	-1.3219	-9.7406	-10.2640	-6.5905
F_{23}	Std	3.5992	0.0207	3.3264	3.8350	0.8038	2.0966	1.4812	3.3464
	Best	-10.5364	**-10.5364**	-10.5363	-10.5364	-4.2306	-10.5363	-10.5360	-10.5364

D. Engineering Optimization

We solved the following two real-world engineering problems using the algorithm SD3WHO proposed in this paper with the aim of verifying its effectiveness, both of which are mathematical models consisting of multiple inequality-constrained single-objective functions. The results of 20 runs are averaged and the optimal values of these functions under the inequality constraints are obtained using SD3WHO, and the optimal values, means and variances are simultaneously compared with those of the improved algorithm and other algorithms to draw conclusions.

(1) Tension/compression Spring Design

The objective of the extension/compression spring design problem is to minimize the spring weight while currently satisfying the constraints of minimum deflection, shear stress, impact frequency, outside diameter limits and design variables. The design problem consists of three design variables including wire diameter $d(X_1)$, average coil diameter $D(X_2)$, and effective number of coils $P(X_3)$. Fig. 5 illustrates the specific model diagram of the problem with the following constraints and objective function specifics.

Set $\vec{X} = [X_1\ X_2\ X_3] = [d\ D\ P]$

Objective function: $f(\mathrm{X}) = (\mathrm{X}_3 + 2)X_2X_1^2$

Constraints: $g_1(\mathrm{X}) = 1 - \frac{X_2^3X_3}{71785X_1^4} \le 0$;

$$g_2(\mathrm{X}) = \frac{4X_2^2 - X_1X_2}{12566(X_1^3X_2 - X)} + \frac{1}{5108X_1^2} - 1 \le 0$$

$$g_3(\mathrm{X}) = 1 - \frac{140.45X_1}{X_2^2X_3} \le 0$$

$$g_4(\mathrm{X}) = \frac{X_1 + X_2}{1.5} - 1 \le 0$$

Boundary constraints: $0.05 \le X_1 \le 2$; $0.25 \le X_2 \le 1.3$; $2 \le X_3 \le 15$.

The WHO and the improved algorithm were experimented on the tension/compression spring design problem and their convergence curves were are shown in Fig. 6. The experimental data are presented in Table IV, where the maximum number of iterations is set to 100, and 30 experiments were performed for each algorithm. The data of optimal value, mean and variance are listed in Table

V, and the best experimental data are marked with bold. The experimental results show that the improved algorithms perform better in solving the tension/compression spring design problem, with SD2WHO performing the best and SD3WHO performing the second best. The improved WHO and other algorithms were adopted to solve the tension/compression spring design problem, and their convergence curves shown in Fig. 7 were compared with the original algorithm. The simulation data of this experiment is detailed in Table IV, in which the maximum number of iterations is set to 100, and the average value is taken in thirty runs. Table VII records the experimental data in detail. In summary, it can be seen that SD3WHO has better data in optimizing the extension/compression spring design problem. From Table VII, it can be seen that SD3WHO has better mean and variance results compared with other algorithms.

(2) Pressure Vessel Design

With regard to the design problem of pressure vessels, it centers on the search for an optimized solution aimed at meeting the production requirements while minimizing the overall cost. This design problem can be summarized as the search for four design variables (inner radius $R(X_1)$, vessel length $L(X_2)$, shell thickness $T_s(X_3)$ and head thickness $T_h(X_4)$) to ensure that the manufacturing cost of the pressure vessel is minimized while satisfying certain boundary condition constraints. The model constructed for this problem is shown in Fig. 8, and its objective function and constraints are specifically described as follows.

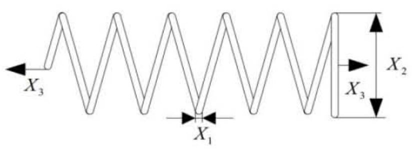


Fig. 5 Tension/compression spring design model.

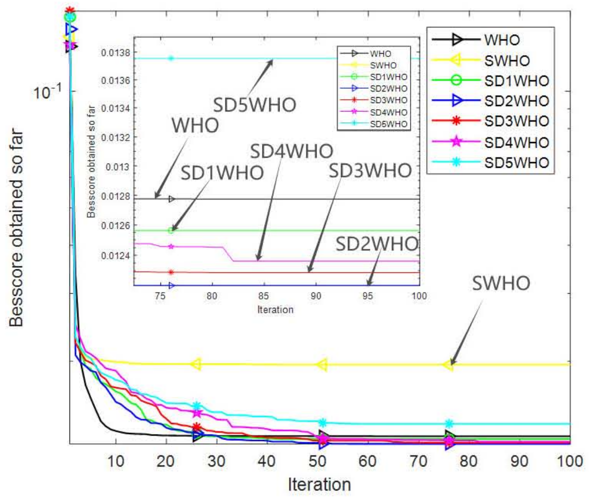


Fig. 6 Convergence diagram of the improved algorithm for optimizing extension/compression springs.

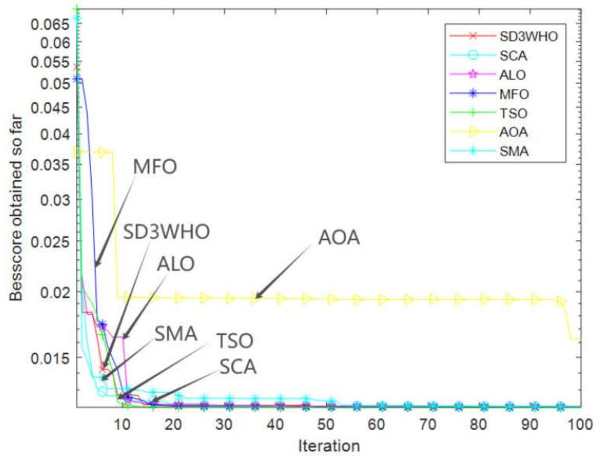


Fig. 7 Convergence diagram of the improved algorithm for optimizing/stretching compression springs.

TABLE IV. BEST SOLUTION TO THE TENSION/COMPRESSION SPRING PROBLEM FROM IMPROVED ALGORITHMS

	WHO	SWHO	SD1WHO	SD2WHO	SD3WHO	SD4WHO	SD5WHO
$f(X)$	0.0121078	**0.01204146**	0.01204577	0.01223485	0.01205398	0.01519645	0.0123115
X_1	0.0532	0.0567	0.0545	0.0535	0.0548	0.0657	0.0543
X_2	0.2500	0.3144	0.2724	0.2673	0.2739	0.4405	0.2670
X_3	10.8235	7.1708	9.4680	9.4215	9.1357	3.5995	9.7896

TABLE V. THE RESULTS OBTAINED BY THE IMPROVED ALGORITHM ON THE TENSION/COMPRESSION SPRING PROBLEM

	WHO	SWHO	SD1WHO	SD2WHO	SD3WHO	SD4WHO	SD5WHO
Ave	0.0128	0.0196	0.0126	**0.0122**	0.0123	0.0124	0.0138
Std	0.0022	0.0073	7.1542e-04	**1.9766e-04**	5.6006e-04	6.2778e-04	0.0063
Best	0.0120	0.0120	0.0120	0.0120	0.0120	0.0120	0.0121

TABLE VI. THE BEST SOLUTION OBTAINED FROM IMPROVED ALGORITHMS FOR THE TENSION/COMPRESSION SPRING PROBLEM

	SD3WHO	SCA	ALO	MFO	TSO	AOA	SMA
$f(X)$	**0.01204032**	0.01208771	0.01228761	0.01204364	0.012107760	0.01934496	0.01210463
X_1	0.0544	0.0567	0.0595	0.0543	0.0532	0.0500	0.0533
X_2	0.2719	0.3163	0.3764	0.2919	0.2500	0.2500	0.2508
X_3	9.3421	6.6855	5.0000	8.1528	10.8284	9.1114	10.7961

Objective function: $f(X) = 0.6224X_1X_3X_4 + 1.7781X_2X_3^2 + 3.1661X_1^2X_4 + 19.84X_1^2X_3$

Constraints: $g_1(X) = -X_1 + 0.0193X_3 \le 0$

$g_2(X) = -X_2 + 0.00954X_3 \le 0$

$g_3(X) = -\pi X_3^2X_4 - \frac{4}{3}\pi X_3^2 + 1296000 \le 0$

$g_4(X) = X_4 - 240 \le 0$

Boundary constraints: $0 \le X_1 \le 99$, $0 \le X_2 \le 99$, $10 \le X_3 \le 200, 10 \le X_4 \le 200$.

The WHO and the improved algorithm were experimented on the pressure vessel design problem, and their convergence curves were compared, as shown in Fig. 9. The results of the experiments are summarized in Table VIII, where the maximum number of iterations for each method was 100 and 20 trials were averaged. In Table IX, we present the optimal, mean, and variance data.The results of SD3WHO demonstrate the superior optimization performance of SD3WHO when solving pressure vessel design problems.From the results in Table IX, the mean, variance and optimal results of SD3WHO are better than other algorithms. The improved WHO with the dual weight starvation strategy and random convergence factor and other algorithms were used to solve the pressure vessel design problem, and their convergence curves were shown in Fig. 10. The maximum number of iterations is 100. The experimental results in Table X show that after an average of 20 experiments, the SD3WHO algorithm performs well in optimizing the pressure vessel design problem. In terms of optimal data, mean and variance, the mean and maximum results of SD3WHO in Table XI outperform the other algorithms.

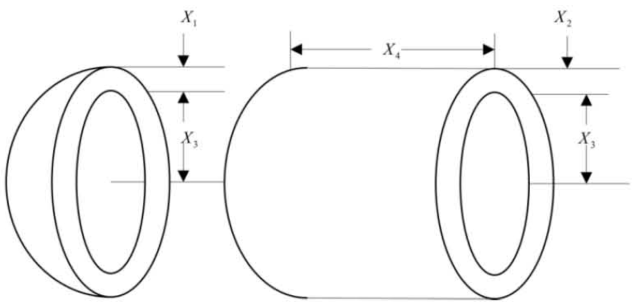


Fig. 8 Model diagram of pressure vessel design problem.

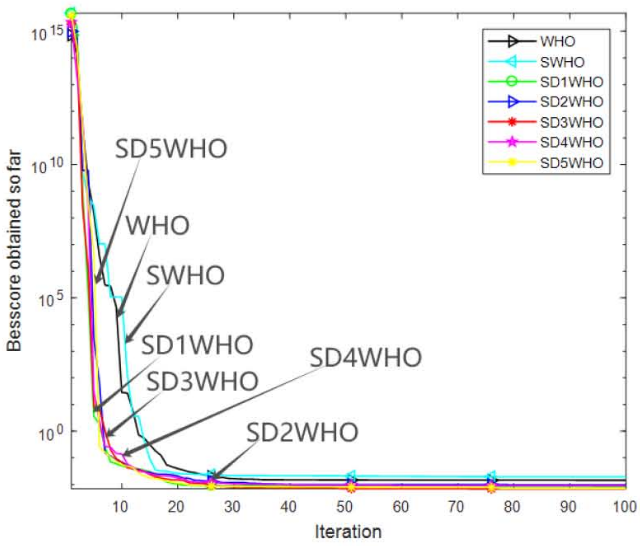


Fig. 9 Convergence diagram of improved algorithm to optimize pressure vessel design.

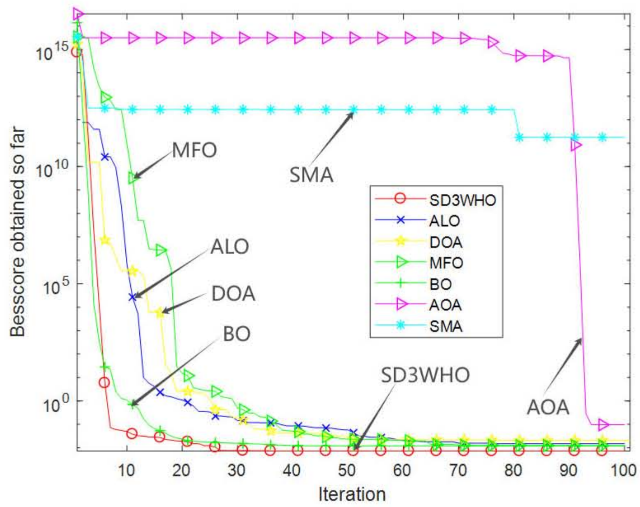


Fig. 10 Convergence diagram of the improved algorithm compared with other algorithms for optimizing pressure vessel design.

TABLE VII. THE BEST SOLUTION OBTAINED FROM OTHER ALGORITHMS AND IMPROVED ALGORITHMS FOR THE TENSION/COMPRESSION SPRING PROBLEM

	SD3WHO	SCA	ALO	MFO	TSO	AOA	SMA
Ave	**0.01204617**	0.01208515	0.01209152	0.01205621	0.01204718	0.01622770	0.01208608
Std	**1.0921e-05**	3.3109e-05	1.1124e-04	1.2608e-05	3.9455e-05	0.0027	3.3885e-05
Best	0.01203017	0.01204409	0.01202211	0.01201961	**0.01201954**	0.01362958	0.01202854

TABLE VIII. THE BEST SOLUTION OBTAINED FROM IMPROVED ALGORITHMS FOR THE PRESSURE VESSEL DESIGN PROBLEM

	WHO	SWHO	SD1WHO	SD2WHO	SD3WHO	SD4WHO	SD5WHO
$f(X)$	0.0121	0.0129	2.8738e-43	**9.2361e-45**	1.4222e-44	1.0952e-41	0.0121
X_1	0.0540	0.0523	3.1148e-15	1.0557e-15	1.2109e-15	9.8473e-15	0.0534
X_2	0.2630	0.2351	2.4684e-15	6.9061e-16	8.0829e-16	9.4120e-15	0.2605
X_3	10	10	10	10	10	10	10
X_4	10	10	10	10	10	10	10

TABLE IX. THE BEST SOLUTION OBTAINED FROM OTHER ALGORITHMS AND IMPROVED ALGORITHMS FOR THE PRESSURE VESSEL DESIGN PROBLEM

	WHO	SWHO	SD1WHO	SD2WHO	SD3WHO	SD4WHO	SD5WHO
Ave	0.0144	0.0195	0.0086	0.0096	**0.0069**	0.0090	0.0071
Std	0.0169	0.0285	0.0179	0.0172	**0.0061**	0.0182	0.0063
Best	6.3733e-23	8.5243e-47	2.0340e-46	1.4700e-47	**8.3853e-49**	6.9602e-45	2.2411e-47

TABLE X. THE BEST SOLUTION OBTAINED FROM OTHER ALGORITHMS AND IMPROVED ALGORITHMS FOR THE PRESSURE VESSEL DESIGN PROBLEM

	SD3WHO	ALO	DOA	MFO	BO	AOA	SMA
$f(X)$	0.0121	0.0121	0.0121	0.0121	0.0121	0.0300	1.7672e+12
X_1	0.0542	0.0533	0.0541	0.0540	0.0540	0.0429	17.8479
X_2	0.2604	0.2505	0.2655	0.2630	0.2630	0.0740	2.7081
X_3	10	11.0117	10	10	10	200	61.0049
X_4	10	121.4207	10	200	10	200	10

TABLE XI. THE RESULTS OBTAINED FROM CHAOS IMPROVED ALGORITHMS FOR THE PRESSURE VESSEL DESIGN PROBLEM

	SD3WHO	ALO	DOA	MFO	BO	AOA	SMA
Ave	**0.0073**	0.0150	0.0209	0.0124	0.121	0.965	1.7753e+11
Std	0.0062	0.0101	0.0278	9.1683e-04	**1.2930e-18**	0.0526	5.5855e+11
Best	**3.8440e-45**	9.2170e-13	0.0121	0.0121	0.0121	1.7311e-42	0.0247

V. CONCLUSION

Since the global search performance of WHO is poor and prone to local optimum, this paper puts forward a improved WHO that incorporates a dual weight starvation strategy and a random convergence factor, aiming to strike a balance between the global and local search ability of the algorithm. It well solves the disadvantages of local optimization and enhances the global search performance, and also improves the convergence speed and convergence accuracy of the algorithm. The optimization capability of the improved algorithm is examined using 23 benchmark test functions and a comparison is made with seven algorithms (WHO, WOA, MFO, RSO, MVO, GWO, and ALO) with respect to the optimization performance as well as comparing the experimental data for each function. Seen form the experimental data results, it can be clearly seen that the proposed WHO with dual weight starvation strategy and the random convergence factor has the best optimization seeking ability. In solving two real engineering problems, comparing with the WHO with the improved algorithm and other algorithms, SD3WHO can find a faster and better solution with the best optimization.

REFERENCES

[1] N. Singh, and J. Kaur, “Hybridizing sine-cosine algorithm with harmony search strategy for optimization design problems,” *Soft Computing*, vol. 25, no. 16, pp. 11053-11075, 2021.

[2] S. Sharma, and V. Kumar, “A comprehensive review on multi-objective optimization techniques: past, present and future,” *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5605-5633, 2022.

[3] L. Abualigah, A. Diabat, and Z. W. Geem, “A comprehensive survey of the harmony search algorithm in clustering applications,” *Applied Sciences*, vol. 10, no. 11, pp. 3827, 2020.

[4] A. Kumar, and J. S. Dhillon, “Enhanced harris hawk optimizer for hydrothermal generation scheduling with cascaded reservoirs,” *Expert Systems with Applications*, vol. 226, pp. 120270, 2023.

[5] L. Abualigah, and A. Diabat, “A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications,” *Neural Computing and Applications*, vol. 32, no. 19, pp. 15533-15556, 2020.

[6] Q. D. Yang, Y. Liu, J. S. Wang, Wang, S. H. Jiang, and X. L. Li, “Soft-sensing modeling of SMB chromatographic separation process based on ELM with variable excitation functions,” *Engineering Letters*, vol. 30, no. 2, pp. 835-846, 2022.

[7] D. Wang, J. S. Wang, S. Y. Wang, and C. Xing, “Adaptive soft-sensor modeling of SMB chromatographic separation process based on dynamic fuzzy neural network and moving window strategy,” *Journal of Chemical Engineering of Japan*, vol. 54, no. 12, pp. 657-671, 2021.

[8] A. Osyczka, and S. Kundu, “A modified distance method for multicriteria optimization, using genetic algorithms,” *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 871-882, 1996.

[9] S. Dan, “Biogeography-based optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702-713, 2009.

[10] K. M. Htay, R. R. Othman, A. Amir, amd J. M. H. Alkanaani, “Gravitational search algorithm based strategy for combinatorial t-way test suite generation,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 4860-4873, 2022.

[11] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: a nature-inspired algorithm for global optimization,” *Neural Computing and Applications*, vol. 27, no. 2, pp. 495-513, 2016.

[12] W. Zhao, L. Wang, and Z. Zhang, “Atom search optimization and its application to solve a hydrogeologic parameter estimation problem,” *Knowledge-Based Systems*, vol. 163, no. 1, pp. 283-304, 2019.

[13] O. K. Erol, and I. Eksin, “A new optimization method: big bang-big crunch,” *Advances in Engineering Software*, vol. 37, no. 2, pp. 106-111, 2006.

[14] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281-295, 2006.

[15] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, “Particle swarm optimization: a comprehensive survey,” *IEEE Access*, vol. 10, pp. 10031-10061, 2022.

[16] M. Jain, V. Saihjpal, and N. Singh, “An overview of variants and advancements of PSO algorithm,” *Applied Sciences*, vol. 12, no. 17, pp. 8392, 2022.

[17] S. Mirjalili, and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.

[18] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228-249, 2015.

[19] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, “A novel algorithm for global optimization: rat swarm optimizer,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8457-8482, 2020.

[20] S. Mirjalili, S. M. Mirjalil, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[21] S. Mirjalili, “The ant lion optimizer,” *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.
[22] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006.
[23] S. Z. Koohi, N. A. W. A. Hamid, M. Othman, and G. Ibragimov, “Raccoon optimization algorithm,” *IEEE Access*, vol. 7, pp. 5383-5399, 2019.
[24] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, “Grasshopper optimization algorithm: theory, variants, and applications,” *IEEE Access*, vol. 9, pp. 50001-50024, 2021.
[25] A. G. Hussien, M. Amin, M. Wang, G. Liang, and H. Chen, “Crow search algorithm: theory, recent advances, and applications,” *IEEE Access*, vol. 8, pp. 173548-173565, 2020.
[26] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization,” *Engineering Applications of Artificial Intelligence*, vol. 90, pp. 103541-103541, 2020.
[27] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: a nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, pp. 113377-113377, 2020.
[28] C. J. Burnett, S. C. Funderburk, J. Navarrete, A. Sabol, J. Liang-Guallpa, T. M. Desrochers, and M. J. Krashes, “Need-based prioritization of behavior,” *eLife*, vol. 8, no. 6, pp. 44527, 2019.
[29] G. Vytenis, and J. G. Jean-Guy, “Foraging under the risk of predation in juvenile Atlantic salmon (Salmo salarL.): effects of social status and hunger,” *Behavioral Ecology and Sociobiology*, vol. 29, no. 4, pp. 255-261, 2004.
[30] J. S. Wang, S. Han, and Q. P. Guo, “Echo state networks based predictive model of vinyl chloride monomer convention velocity optimized by artificial fish swarm algorithm,” *Soft Computing*, vol. 18, no. 3, pp. 457-468, 2014.
[31] N. Iraj, and K. Farshid, “Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems,” *Engineering with Computers*, vol. 38, no. 4, pp. 3025-3056, 2022.