MSWOA: Multi-strategy Whale Optimization Algorithm for Engineering Applications

Ronghe Zhou, Yong Zhang*, Xiaodong Sun, Haining Liu and Yingying Cai

Abstract—**The Whale Optimization Algorithm (WOA) is a novel algorithm that was motivated by the prey behavior of humpback whales. WOA has attracted a lot of interest due to its few parameters and easy implementation, but it also has sluggish convergence speed, poor convergence accuracy, and is is easy to get a local optima. In this paper, a multi-strategy WOA called MSWOA is constructed to address these drawbacks. It includes dimensional updating, nonlinear convergence factor, global perturbation factor, firefly perturbation, and vertical and horizontal crossover learning strategy. First, a strategy was developed to update each dimension differently to avoid MSWOA from reaching a local optima. Second, a nonlinear convergence factor is devised to better balance the MSWOA's search ability. Third, a global perturbation factor is considered during the exploration phase, to enrich the whale population. Fourth, a firefly perturbation strategy is employed to increase convergence accuracy. Fifth, a vertical and horizontal strategy is applied to accelerate convergence. Finally, twelve CEC2022 benchmark functions and three engineering cases are adopted to evaluate the performance of MSWOA. The results confirm that MSWOA is superior and competitive.**

Index Terms—**Whale Optimization Algorithm, dimensional updating, firefly perturbation, vertical and horizontal crossover strategy, engineering applications**

I. Introduction

In real world, many complex problems are often optimization problems, such as continuous optimization problems [1], [2], production scheduling problems [3], [4], [5], path planning problems [6], [7], [8], assembly line balancing problems [9], [10], neural network training [11], power systems [12], feature selection [13], [14], site selection [15], and wireless sensor network [16] etc. As you can see, optimization is ubiquitous, optimization is our daily life all the time to face a problem, we always want to use the least cost, to achieve maximum economic benefits, so optimization issues and we are closely related to and is essential. Therefore, optimization problems are closely related to us and are of vital importance. Only by establishing efficient solution models according to the problems can we better solve the various challenges in life and promote the development of society. Generally speaking, they can be divided into these types: biologically inspired intelligent optimization algorithms, like Coati Optimization Algorithm (COA) [17] and Mountain Gazelle Optimizer (MGO) [18]. Socially inspired intelligent optimization algorithms, similar to Cultural Algorithm (CA) [19]. Physics-inspired intelligent optimization algorithms, as Gravity Search Algorithm (GSA) [20] and Simulated Annealing (SA) [21]. Chemical-inspired intelligent optimization algorithms, such as Chemical Reaction Optimization (CRO) [22]. Mathematically-inspired intelligent optimization algorithms, like Estimation of Distribution (ED) [23]. Of course, there may be more classifications, and roughly speaking, intelligent algorithms can be divided into the above categories. With a significant number of evolutionary algorithms proposed, intelligent algorithms may have more categories, but in any case, intelligent algorithms also simulate various laws of nature in real life, and then perform mathematical modeling. These algorithms tend to be highly randomized, exhaustively enumerating the solutions within a certain range of objectives, and then performing greedy selection to eventually obtain a near-optimal solution.

Manuscript received December 29, 2023; revised June 25, 2024.

Ronghe Zhou is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (e-mail: ronghezhou@ustl.edu.cn).

Yong Zhang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (*Corresponding author, phone: 86-0412-889699; e-mail: zy9091@163.com).

Xiaodong Sun is an engineer of Dagushan Pelletizing Mill of Anshan Steel Group Mining Company, Anshan, Liaoning 114046, PR China (e-mail: sunle1020@163.com).

Haining Liu is a lecturer of Anshan College of Technology, Anshan, Liaoning 114013, PR China (e-mail: 13841262060@163.com).

Yingying Cai is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (e-mail: AC18364996132@163.com).

WOA is proposed by Mirjalili and Lewis [24]. Since its introduction, WOA has garnered significant attention from scholars in various fields. The WOA is also an algorithm that simulates biological evolution in nature and is proposed by group intelligence. This type of algorithm is generally strong in solving ability, better adaptability, and can adapt to a variety of optimization problems. Hemasian-Etefagh and Safi-Esfahani [25] came up with a new concept in whale grouping to overcome the WOA convergence phenomenon that had appeared earlier. Jiang et al. [26] constructed an improved WOA with armed force program and strategic adjustment. Kaveh and Ghazaan [27] developed a enhanced WOA called EWOA, and it is applied to four optimization problems of skeletal structure and proved to have better performance. Lee and Zhuo [28] designed a hybrid version of WOA called GWOA-TEO. This variant combines genetic algorithms and thermal exchange optimization to enhance the global optimization capability of WOA. Li et al. Akyol and Alatas [29] applied WOA in conjunction with optimization based on social impact theory for sentiment classification in online social media. Luo and Shi [30] employed a hybrid version of WOA called MDE-WOA, which incorporates improved differential evolution for addressing global optimization problems. Sun and Chen [31] presented a multi-population improved WOA called MIWOA, specifically designed for high-dimensional optimization. Wu et al. [32] utilizes a non-linear control based on arcsine function to optimize the WOA. Chakraborty et al. [33] combined the hunger games search with WOA and demonstrated the effectiveness of the newly designed algorithm.

We are all familiar with the No Free Lunch (NFL) theorem [34], this means that our algorithm is not suitable for everything and needs to be dynamically adapted to the characteristics of each issue. We need to weigh the performance of the algorithm, speed and other indicators when designing the algorithm. An algorithm may be efficient in solving one problem, but perform poorly in solving another. Therefore, it is necessary to consider and adopt different algorithms when facing different optimization problems, and efficient and intelligent optimization algorithms should be constructed based on the specific characteristics of each problem, which is our goal.

Although WOA performs well in many areas, it is not without its problems. One major issue is the improper balance of the convergence factor, this can result in difficulties in escaping local optima. Additionally, in the end the iteration, there is a lack of strategies to enhance population diversity.Of course, each algorithm may have its own shortcomings and strengths, we try to keep the advantages of the algorithm and make up for the shortcomings of the algorithm when we carry out the design of the algorithm, or we can combine the two algorithms to complement each other's advantages.

To address these issues, five strategies were constructed for improving WOA. Firstly, we design a strategy to update each individual's dimension in a different manner. In the original WOA, individuals are updated in the same way for each dimension, which leads to a single update of the solution, and in evolutionary algorithms we generally want the solution to move so that the diversity of the population is enriched. So we design a different updating way for each dimension, which will result in a greater diversity of solutions produced. In the standard WOA, all dimensions of an individual are updated in the same way, which makes the algorithm prone to getting in local optima. Thus, in this paper, we recompute the parameters when updating each dimension of each individual. Second, a non-linear convergence factor aiming to better balance the MSWOA. At the beginning of the population iteration, a bigger convergence factor can better enhance the global search ability, a smaller convergence factor can be better exploited. In evolutionary computation, we generally expect the global optimization capability to be stronger in the early stage. But in the later stage, this is not helpful for the algorithm to converge, so we hope that in the later stage of the algorithm, the local search ability is stronger, and the nonlinear convergence factor solves this problem very well. Third, a global perturbation factor is intended to enhance the algorithm's ability to explore. In the end of the iteration, randomly selecting a search agent will cause the algorithm to fall into local optima. Fourth, the firefly perturbation strategy is used for enhancing the search agents learn from the optimal individual. And through learning, other individuals can move towards the best whale to increase the convergence accuracy. Fifth, the vertical and horizontal crossover strategy is used to accelerate convergence, the vertical crossover allows individuals to learn independently, and use horizontal crossover to ensure that different individuals learn from each other in pairs, thereby improving the algorithm's exploratory capability.

There are five contributions in this paper, which are:

(1) A strategy was designed to update each individual dimension in a different way. This results in richer and better quality solutions being produced.

(2) A nonlinear convergence factor was constructed to better balance the MSWOA.

(3) A global perturbation factor was added to the exploration phase to enhance the variety of population .

(4) The firefly perturbation strategy was used to increase the convergence accuracy.

(5) A novel vertical and horizontal crossover strategy was developed to accelerate convergence.

In addition, the MSWOA algorithm was applied to solve the CEC2022 test function as well as three real engineering problems. The MSWOA was compared with 10 classical algorithms, and the experimental results demonstrated that MSWOA outperformed the others.

The paper is arranged so. The Section II is a short description of WOA. The Section III is a statement of MSWOA. Test functions and parameter settings are shown in Section IV. Section V is experimental results. The conclusion is discussed in Section VI.

II. Standard WOA

The WOA is proposed in 2016 [24]. The process of WOA is roughly divided into three parts, search for prey, encircling prey and bubble-net attacking and each process will be described separately below.

A. Exploration

This phase is dominated by vector $\vec{A}$. When $\vec{A}$ exceeds 1 or falls below -1, a whale is randomly selected. The formulas are defined as Eqs.(1) and (2).

$$\overrightarrow{Distan} = \left|\vec{C} \cdot \overrightarrow{X_{\text{rnd}}} - \vec{X}\right| \tag{1}$$

$$\vec{X}(iter+1) = \overrightarrow{X_{\text{rnd}}} - \vec{A} \cdot \overrightarrow{Distan} \tag{2}$$

where $iter$ indicates the current iteration, $\vec{X}(iter+1)$ is the position of next generation of whales, $\overrightarrow{X_{\text{rnd}}}$ is a random position vector. $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}$ is the position vector, $|\ |$ is the absolute value. $\overrightarrow{Distan}$ indicates the distance vector of the i-th whale to the prey (a random whale). $\vec{A}$ and $\vec{C}$ are two parameter vectors, and they are calculated by Eqs.(3) and (4).

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r_2} \tag{4}$$

where $\vec{a}$ is linearly decreased from 2 to 0 and $\vec{r_1}$ and $\vec{r_2}$ are uniformly distributed random vectors in [0,1].

B. Exploitation phase

The exploitation phase occurs when the humpback whale identifies its prey and initiates an attack.

1) Encircling prey: In this stage, when a humpback whale detects its prey, it initiates the prey encirclement phase. It is assumed that the prey represents the optimal individual within the existing whales. As a result, the other search agents adjust their positions towards the current optimal individual. This adjustment process can be described by Eq.(5) and Eq.(6).

$$\overrightarrow{Distan} = \left|\vec{C} \cdot \overrightarrow{Globalbest} - \vec{X}(iter)\right| \tag{5}$$

$$\vec{X}(iter+1)=\overrightarrow{Globalbest}-\vec{A}\cdot\overrightarrow{Distan} \tag{6}$$

where $\overrightarrow{Globalbest}$ is the best solution at present, if there is a better individual for the current iteration, it will be updated.

2) Bubble-net attacking: In this stage, the humpback whale utilizes a technique of spitting out bubbles to attack its prey. The mathematical models describing this stage are expressed in Eqs.(7) and (8).

$$\overrightarrow{Distan'}=\left|\overrightarrow{Globalbest}-\vec{X}(iter)\right| \tag{7}$$

$$\vec{X}(iter+1)=\overrightarrow{Distan'}\cdot e^{bl}\cdot\cos(2\pi l)+\overrightarrow{Globalbest} \tag{8}$$

where $\overrightarrow{Distan'}$ represents the distance vector between the existing whales and the best whale individual, l is in the domain of [-1,1], b is a constant and is equal to 1.

When a whale captures its prey, the behaviors of bubble netting to attack and encircle the prey occur simultaneously. It can be defined as Eq.(9). where p is in [0,1].

The detailed information oft the WOA, one can refer to reference [24].

III. MSWOA

For solving the drawbacks of WOA, five strategies are constructed in MSWOA, which are dimensional updating strategy, non-linear convergence factor, global perturbation factor, firefly perturbation, and vertical and horizontal crossover. We propose all these strategies with the ultimate goal of enabling the solution to move so that more ranges can be found and how to improve the likelihood of finding an even better solution. The detailed description will be shown in the next sections.

A. Dimensional updating strategy

In the standard WOA algorithm, at each iteration, each search agent is used with the same parameters p, r_1 and A, this leads to a single way of population renewal and a reduction in population diversity. For solving this problem, we designed the dimensional updating strategy in MSWOA. As a more visual demonstration of it, the pseudo-code of the dimensional updating is listed in Algorithm 1.

In the Algorithm 1, the selection probabilities p, $r1$ and A are recalculated when updating each dimension of each individual, which leads to a different update method when updating different dimensions of each individual and improve the diversity of the whales' population. In previous algorithm designs, different dimensions of each individual are often updated in the same way, which tends to lead to a reduction of the population diversity, which is one of the reasons why the standard WOA population has low diversity and fall into local optima.

B. Non-linear convergence factor

In standard WOA, the population iteration uses a uniformly varying linear convergence factor a from the beginning to the end, which results in an algorithm that does not balance the exploration ability and exploitation ability well. We are familiar that in the early iterations, a convergence parameter with a larger variation is more favorable for global search, while in the end, a convergence parameter with a smaller variation is more favorable for local exploitation. For this reason, the nonlinear convergence factor can be calculated using Eq.(10).

$$a=2-2\cdot\sqrt{\frac{iter}{T}} \tag{10}$$

where T is the maximum number of runs of the MSWOA.

The Fig. 1 depicts the trend of the convergence factor a follows $iter$. To show the trend of parameter a more intuitively, in our experiment, two different maximum iterations are set. From the figure, we can see that the parameter a gradually decreases from 2 to 0. The change of parameter a is large in the early iterations, and the trend becomes smaller as the iterations proceed, which is better for MSWOA.

C. Global perturbation factor

As can be seen from the exploration phase of the standard WOA, the search agent randomly selects individuals for the global search. At the beginning, this operation has little impact on the algorithm due to the high population diversity. However, as the iteration goes on, the population diversity gradually lowers, and randomly choosing individuals increases the probability of repeatedly selecting the same individuals. To solve this problem, this paper introduces a perturbation factor in the exploration phase of MSWOA.The $\vec{X}$ is changed to Eq.(11).

$$\vec{X}(iter+1)=\vec{G}\cdot\overrightarrow{X_{\text{rnd}}}-\vec{A}\cdot Dis\vec{tan} \tag{11}$$

where G is in [0,1].

To better visualize the effect of the added perturbation factors, Fig. 2 clearly demonstrate the change in population diversity with and without the perturbation factors. The population diversity is defined according to reference [35], using Eq.(12) and Eq.(13).

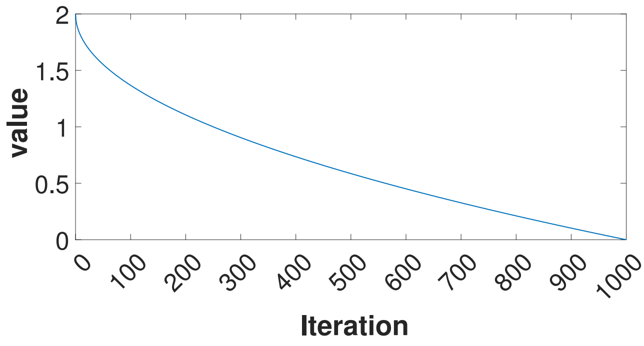
$$\text{diversity}(NS)=\frac{1}{NS}\sum_{i=1}^{NS}\sqrt{\sum_{j=1}^{D}\left(X_i^j-\overline{X^j}\right)^2} \tag{12}$$

$$\overline{X}=\left(\frac{1}{NS}\sum_{i=1}^{NS}X_i^1,\frac{1}{NS}\sum_{i=1}^{NS}X_i^2,\cdots\frac{1}{NS}\sum_{i=1}^{NS}X_i^D\right) \tag{13}$$

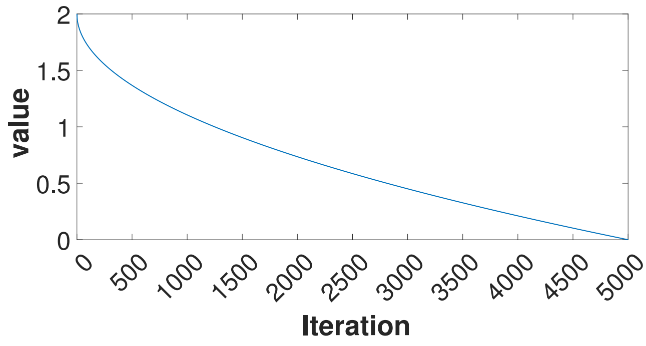
where NS is the size of whales, and D is the dimensionality. X_i^j is the variable on the j-th dimension of the search agent i and $\overline{X^j}$ represents the average variable value on the j-th dimension for all search agents in the population.

The lack of diversity tends to cause the algorithm to trap into optima or appear premature convergence. The unimodal function f_1 was selected to check the difference in diversity between with global perturbation factors and without global perturbation factors. In this experiment, all experimental parameters are the same except for the global perturbation factor. From the Fig. 2, we can see that the population diversity with the addition of the perturbation factor is significantly better than that without the convergence factor, which indicates that the addition of the perturbation factor increases the population diversity. It is obvious in functions f_1 that the population diversity is significantly higher in the

$$\vec{X}(iter+1) = \begin{cases} \overrightarrow{Globalbest} - \vec{A} \cdot \overrightarrow{Distan} & p < 0.5 \\ \overrightarrow{Distan'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{Globalbest} & p \geq 0.5 \end{cases} \tag{9}$$

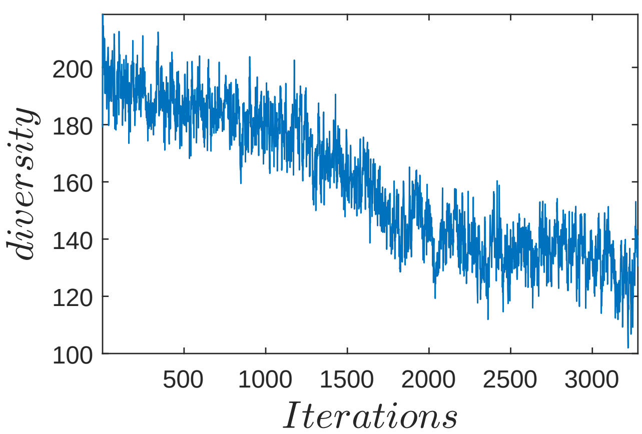


(a) Iteration 1000 of a

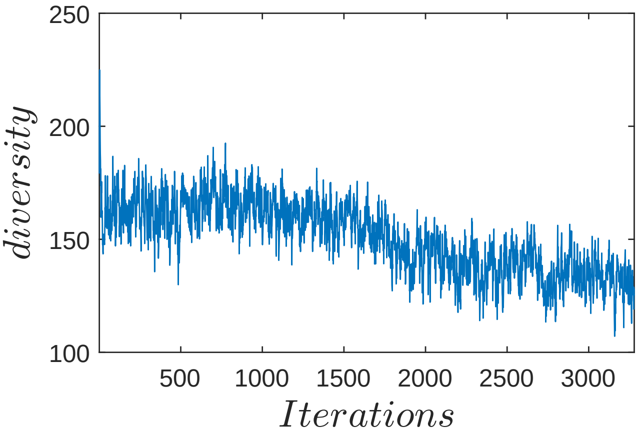


(b) Iteration 5000 of a

Fig. 1: Trend chart of a with different maximum iterations.



(a) With global perturbation factors



(b) Without global perturbation factors

Fig. 2: The population diversity changes with global perturbation factors and without global perturbation factors of f_1

first period with the addition of the convergence factor, which is more favorable to the MSWOA for global exploration. Once again, it is shown that MSWOA is able to achieve a better balance with the addition of a perturbation factor.

D. Firefly perturbation

The firefly perturbation strategy is proposed in firefly algorithms [36]. In standard WOA, the location update is done by the distance among the current whale and the best whale in the encircling phase. As the iteration proceeds, the whales' diversity will diminish, which will cause a decrease in the accuracy of the solution, so we introduce the firefly perturbation and used in the MSWOA. The specific definition is expressed as follows Eqs.(14) and (15).

$$r_{ij} = \|X_i(iter) - Globalbest\| = \sqrt{\sum_{j=1}^{D} \left(X_i^j(iter) - Globalbest^j\right)^2} \tag{14}$$

$$X_i(iter+1) = X_i(iter) + \beta_0 e^{-\gamma r_{ij}^2} \cdot (X_i(iter) - Globalbest) + \alpha \cdot \left(rand - \frac{1}{2}\right) \tag{15}$$

where r_{ij} is the Cartesian distance, $Globalbest$ is the best individual at the moment, D is the dimension, $rand$ is in [0, 1], β_0 is equal to 2, γ is equal to 1, α is set to 0.2.

E. Vertical and horizontal crossover strategy

In standard WOA, each individual whale is updated according to the search formula, as the iteration goes on, the population diversity gradually drops and the lack of information exchange between populations leads to slow convergence. To address the drawback, a novel vertical and horizontal crossover learning strategy inspired by Meng et al. [37] is proposed, which is divided into horizontal crossover learning and vertical crossover learning. Cross-learning between populations is useful for improving the MSWOA, and it promotes mutual learning between different individuals.

Horizontal crossover learning refers to the two-by-two exchange learning of different individuals in a population, with a probability p for the same dimension of two different individuals. It is defined as Eqs.(16) and (17), where X_{n_1} and X_{n_2} are two different individuals randomly choosen from the existing whales, c_1 and c_2 are in [0,1], $XNew_{n_1}$ and $XNew_{n_2}$ are newly created individuals through X_{n_1} and X_{n_2}.

Vertical crossover learning refers to the $Globalbest$ self-learning of the current population. In the end of iteration, the population tends to get trapped in local optima. By performing vertical crossover with a certain probability on two different dimensions of $Globalbest$, the population can escape local optima and accelerate convergence. The $newGbest_{j_1}$ can be obtained using Eq.(18).

$$\begin{aligned} newGbest_{j_1} = q \cdot Globalbest_{j_1} + (1-q) \cdot \\ (Globalbest_{j_1} + Globalbest_{j_2}) \end{aligned} \tag{18}$$

where j_1 and j_2 are two different dimensions, q is in [0,1].

F. Framework of MSWOA algorithm

In summary, the MSWOA incorporates five strategies: dimensional updating strategy, non-linear convergence factor, global perturbation factor, firefly perturbation, and vertical and horizontal crossover learning strategy. To provide a clearer understanding of the flow and implementation of the proposed MSWOA, Algorithm 1 presents the pseudocode of MSWOA, and Fig. 3 visualizes the flowchart of MSWOA. The pseudocodes and flowchart can visually help to understand the program framework of MSWOA.

Algorithm 1: Algorithm description of the proposed MSWOA

Initialize: Whales' Population NS, Dimensionality D, Max function evaluations $maxFEs$, location boundary [X_{min}, X_{max}]
    Output: Best Solution
%=== Initialize Population ===
$X = (X_1, X_2, \cdots, X_{NS})$;
Evaluate the fitnesses of X;
Set $Globalbest$ and $gbestvalue$;
Update FEs;% FEs: function evaluations
%=== Iteration ===
$T = maxFEs/NS$;
$iter = 0$;
while $FEs <= maxFEs$ && $iter <= T$ **do**
    $iter = iter + 1$;
    Update a by Eq.(10);
    for $i = 1$ ***to*** NS **do**
        for $j = 1$ ***to*** D **do**
            Update r_1, p and A;
            if $p < 0.5$ **then**
                if $|A| >= 1$ **then**
                    Eq.(11);
                end
                if $|A| < 1$ **then**
                    Eq.(15);
                end
            else
                Update X_i^j by Eq.(9);
            end
        end
        Check boundaries and evaluate fitnesses of X_i;
        $FEs = FEs + 1$;
        Update $Globalbest$ and $gbestvalue$;
    end
    %=== Vertical and Horizontal Crossover Learning Strategy Process ===
    Update X by Eq.(16) and Eq.(17);
    for *i=1 to N* **do**
        Check boundaries and evaluate fitnesses of X_i;
        Update $Globalbest$, $gbestvalue$ and FEs;
    end
    Evaluate $Globalbest$ by Eq.(18);
    Check boundaries and evaluate fitnesses of $Globalbest$;
    Update $Globalbest$, $gbestvalue$ and FEs;
end
Return $Globalbest$

IV. Benchmark test functions and parameter settings

A. Benchmark test functions

The CEC2022 benchmark test set is used to verify the performance of the MSWOA, and detailed information about the CEC2022 test set can be referred to [38]. The CEC2022 test functions consist of four types of functions, which can provide strong evidence of the algorithm's effectiveness and its ability to avoid local optima.

B. Parameter settings

Table I lists the comparison algorithms and their parameter settings, which are taken from the relevant references.

The population size is 30, the maximum number of function evaluations is 200000, and the number of running times for each algorithm is 30.

V. Experimental results and analysis

We qualitatively analyze the results of MSWOA in comparison with other algorithms on various test functions. Additionally, three practical engineering problems were employed to further validate the performance of MSWOA. Qualitative analysis can help us further analyze the operation mechanism of the algorithm and the reasons for achieving such excellent performance as it is currently. It helps us to visualize how the algorithm works, which is crucial for us to design the algorithm. Verifying the performance of MSWOA in real

$$XNew_{n_1}(iter+1) = X_{n_1}(iter) + c_1 \cdot (X_{n_1}(iter) - X_{n_2}(iter)) + (1-c_1) \cdot (X_{n_1}(iter) - X_{n_2}(iter)) \quad (16)$$

$$XNew_{n_2}(iter+1) = X_{n_2}(iter) + c_2 \cdot (X_{n_1}(iter) - X_{n_2}(iter)) + (1-c_2) \cdot (X_{n_1}(iter) - X_{n_2}(iter)) \quad (17)$$

TABLE I: Comparison algorithm parameter settings.

Algorithm	Reference	Year	Parameters settings
PSO	[39]	1995	c_1 = 2,c_2 = 2, ω = 0.4 $\sim$ 0.9
DE	[40]	1995	Crossover=0.9, Scale factor (F)=0.5
GWO	[41]	2014	a=2 $\sim$ 0, A=2 $\sim$ 0, C=2·rand(0,1)
SCA	[42]	2016	a=2 $\sim$ 0, r_1=r_2=r_3=r_4=rand(0,1)
WOA	[24]	2016	a=2 $\sim$ 0, A=2 $\sim$ 0, C=2·rand(0,1), l=-1 $\sim$ 1, b=1
ASO	[43]	2018	Multiplier weight=0.2,Depth weight=50
SOA	[44]	2019	Control Parameter (A)=2 $\sim$ 0, f_c=2
BOA	[45]	2019	Modality=0.01, Probability=0.8
WOASCALF	[46]	2021	p=0 $\sim$ 1, β=0 $\sim$ 2, σ_v=1
SCSO	[47]	2022	r_G=2 $\sim$ 0, R=-2·r_G $\sim$ 2·r_G
MSWOA	present	present	

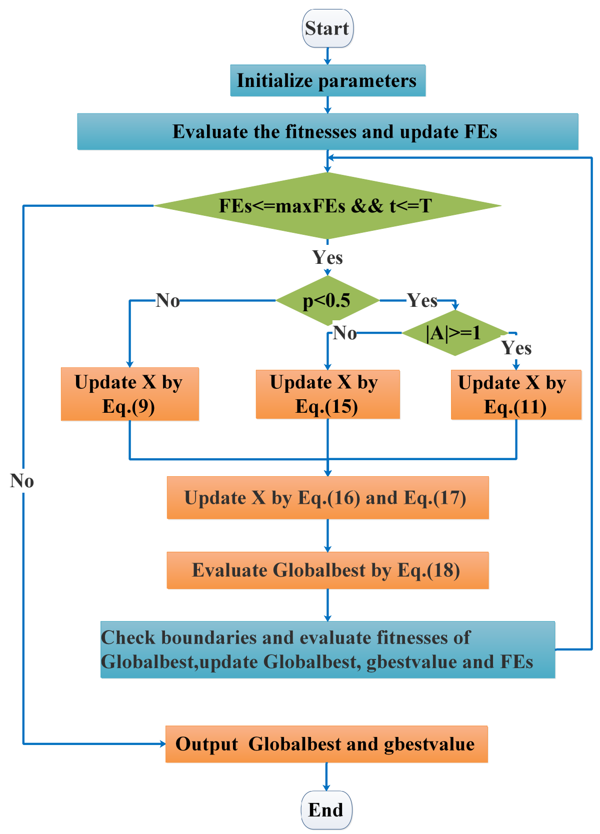


Fig. 3: Flowchart of MSWOA.

problems can be more relevant to our real life, because the test function is often not comprehensive enough, and we can understand the performance characteristics of MSWOA more deeply through the solution of real problems.

A. The qualitative results of MSWOA

This section utilizes the function f_1 to verify the performance of MSWOA. The dimension is equal to 2, whales' population number is equal to 4, and algorithm's iteration is 100. In order to characterize the performance of MSWOA, We plotted the history of individual searches in the algorithm, the variation of the population fitness values ground and the trajectory curve. The result is shown in Fig. 4.

The graph provides a visual representation of the individuals's position transformation. In the beginning of the iteration, the individuals are more scattered, indicating a strong exploration ability. It is obvious from the search history graph, all the individuals gradually converge towards the best solution as the iteration proceeds. From the graph, we can see that the distribution of individuals is relatively wide, again indicating that the MSWOA's global search ability is very good. From the change in the curve of the average fitness of the population, we can see thar MSWOA fluctuates in the beginning stage, showing that MSWOA is performing global optimization, then converges near the optimal solution. From the trajectory graph of x1, it can be seen that the four individuals fluctuate more at the beginning, but then gradually converge, indicating that the four individuals are approaching the optimal solution.

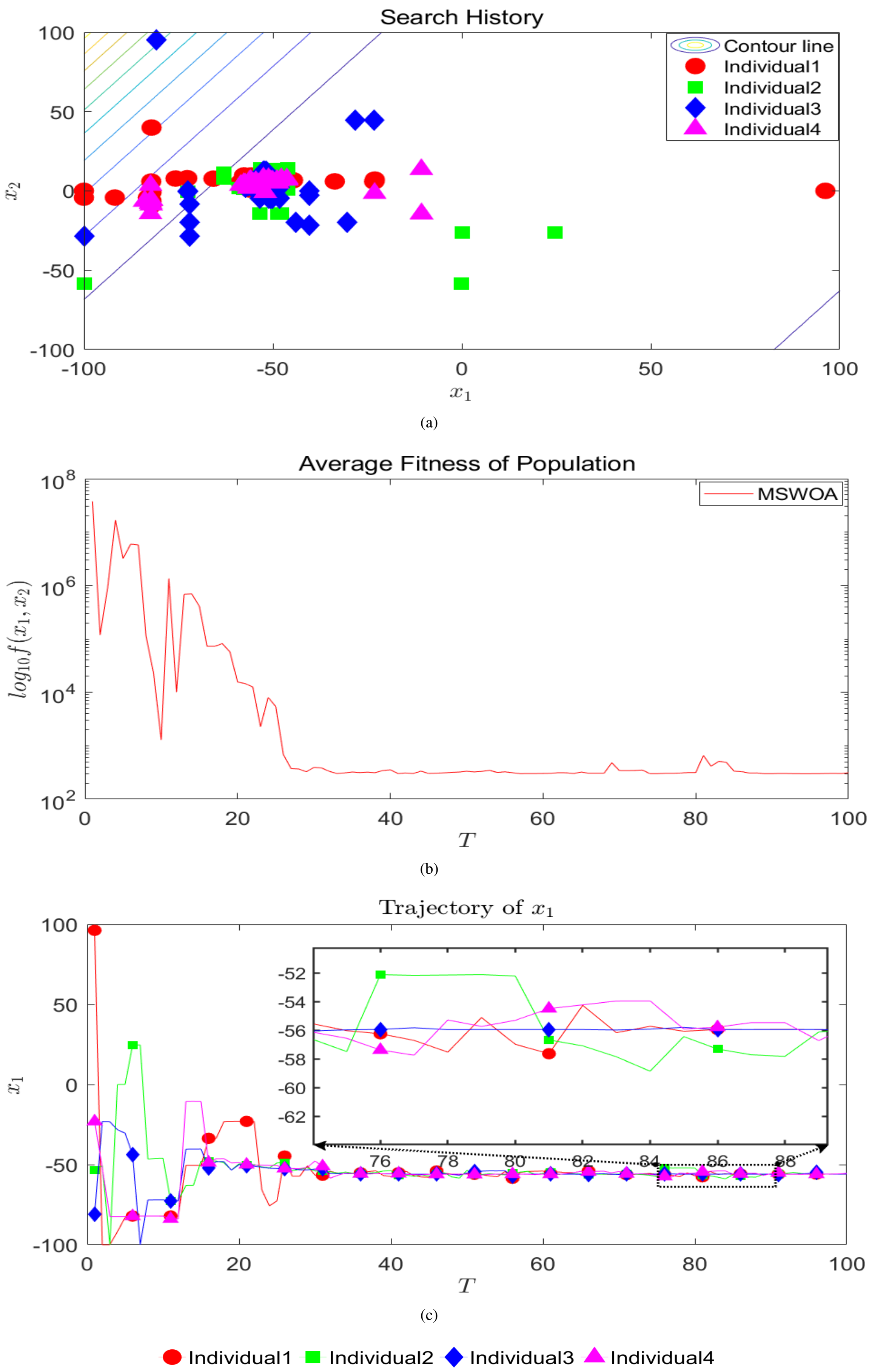
B. Comparison on CEC2022 test set

In this section, the performance of MSWOA is compared with PSO, GWO,SCA, SCSO, SOA, DE, ASO, BOA and variants of WOA. The results are listed in Table II. The convergence curves are depicted in Fig. 5. Analyzing the 10-dimensional problem from Table II, we observe that MSWOA outperforms the other 10 algorithms, like f_1, f_2, f_3, f_5, f_7, f_8 and f_9, in terms of overall performance. However, it has a bad performance on f_4, f_6, f_{10}, f_{11} and f_{12}. Nevertheless, when considering the average performance across all functions, MSWOA still emerges as the top performer.

The boxplots are presented in Fig. 6. In the boxplots, red plus signs are outliers. From the figures, it is evident that the boxplots of MSWOA demonstrate competitive performance across the majority of the tested functions. This suggests that MSWOA exhibits excellent stability.

TABLE II: Results for $10D$ of different algorithms.

f(x)	Index	PSO	GWO	SCA	SCSO	WOA	WOASCALF	SOA	DE	ASO	BOA	MSWOA
f_1	Mean	1.6079E+03	8.1961E+02	6.5449E+02	8.6529E+02	4.8110E+03	4.0658E+03	5.7445E+02	8.9413E+03	2.4345E+04	1.0414E+04	**3.0007E+02**
	Std	3.3367E+03	1.0810E+03	1.2262E+02	8.9937E+02	2.2545E+03	1.4465E+03	5.0809E+02	2.1449E+03	9.9409E+03	3.0460E+03	3.3768E-02
	Best	3.0000E+02	3.0021E+02	4.2869E+02	3.1730E+02	9.9871E+02	1.4038E+03	3.3995E+02	2.0964E+03	3.5540E+03	4.8555E+03	3.0002E+02
	Rank	6	4	3	5	8	7	2	9	11	10	**1**
f_2	Mean	4.6568E+02	4.1524E+02	4.4748E+02	4.2366E+02	4.1721E+02	4.9446E+02	4.2756E+02	5.9770E+02	6.5045E+02	3.0871E+03	**4.1092E+02**
	Std	7.2263E+01	1.7392E+01	2.0573E+01	2.6288E+01	2.4381E+01	1.9380E+01	5.7651E+01	4.9686E+01	1.6594E+02	1.3760E+03	1.7394E+01
	Best	4.0824E+02	4.0012E+02	4.1849E+02	4.0016E+02	4.0004E+02	4.4795E+02	4.0022E+02	4.7091E+02	4.6574E+02	8.3076E+02	4.0000E+02
	Rank	7	2	6	4	3	8	5	9	10	11	**1**
f_3	Mean	6.0350E+02	6.0066E+02	6.1438E+02	6.1203E+02	6.2996E+02	6.3366E+02	6.0586E+02	6.4419E+02	6.4098E+02	6.5907E+02	**6.0011E+02**
	Std	5.2663E+00	1.2637E+00	2.6240E+00	7.3299E+00	1.3736E+01	3.7674E+00	3.7514E+00	5.0389E+00	1.7250E+01	7.3301E+00	3.3186E-02
	Best	6.0000E+02	6.0001E+02	6.0746E+02	6.0215E+02	6.0827E+02	6.2506E+02	6.0018E+02	6.3036E+02	6.0560E+02	6.4505E+02	6.0003E+02
	Rank	3	2	6	5	7	8	4	10	9	11	**1**
f_4	Mean	8.2073E+02	8.1370E+02	8.3261E+02	8.2613E+02	8.4017E+02	8.4755E+02	8.1976E+02	8.6424E+02	8.4814E+02	8.6427E+02	8.2405E+02
	Std	7.2958E+00	6.5228E+00	4.2736E+00	8.2447E+00	1.4017E+01	6.0287E+00	6.0325E+00	7.0982E+00	1.7794E+01	9.3396E+00	9.8407E+00
	Best	8.0696E+02	8.0310E+02	8.2512E+02	8.1257E+02	8.1805E+02	8.3715E+02	8.0722E+02	8.4334E+02	8.0895E+02	8.3680E+02	8.0995E+02
	Rank	3	1	6	5	7	8	2	10	9	11	4
f_5	Mean	9.2274E+02	9.1020E+02	9.4938E+02	1.0096E+03	1.2111E+03	1.1910E+03	9.6246E+02	1.7447E+03	1.7793E+03	1.6179E+03	**9.0109E+02**
	Std	6.9043E+01	1.5396E+01	2.3672E+01	1.2448E+02	2.0888E+02	7.2792E+01	5.3592E+01	1.7999E+02	7.0845E+02	1.6547E+02	5.3446E+00
	Best	9.0000E+02	9.0001E+02	9.1939E+02	9.0135E+02	9.3924E+02	1.0532E+03	9.0026E+02	1.4398E+03	9.0000E+02	1.3732E+03	9.0000E+02
	Rank	3	2	4	6	8	7	5	10	11	9	**1**
f_6	Mean	5.9479E+03	5.6457E+03	7.5226E+05	3.9687E+03	3.2089E+03	5.0222E+06	6.2690E+03	6.1307E+06	1.0018E+08	1.2290E+09	4.8927E+03
	Std	2.0519E+03	2.4817E+03	7.3061E+05	1.6443E+03	1.6157E+03	3.7676E+06	2.3574E+03	4.7999E+06	2.5975E+08	1.1465E+09	2.3479E+03
	Best	1.9621E+03	1.9363E+03	6.9803E+04	1.8559E+03	1.8547E+03	4.5371E+04	2.0308E+03	4.3460E+05	3.5328E+03	2.6309E+07	1.8283E+03
	Rank	5	4	7	2	1	8	6	9	10	11	3
f_7	Mean	2.0281E+03	2.0246E+03	2.0464E+03	2.0403E+03	2.0475E+03	2.0740E+03	2.0292E+03	2.0752E+03	2.0971E+03	2.1217E+03	**2.0163E+03**
	Std	2.5105E+01	1.0556E+01	6.6204E+00	1.6490E+01	1.3525E+01	6.6326E+00	7.4434E+00	1.0471E+01	4.5124E+01	1.9330E+01	8.2934E+00
	Best	2.0010E+03	2.0000E+03	2.0355E+03	2.0144E+03	2.0247E+03	2.0596E+03	2.0203E+03	2.0538E+03	2.0351E+03	2.0904E+03	2.0002E+03
	Rank	3	2	6	5	7	8	4	9	10	11	**1**
f_8	Mean	2.2213E+03	2.2205E+03	2.2283E+03	2.2239E+03	2.2293E+03	2.2371E+03	2.2230E+03	2.2381E+03	2.2388E+03	2.9773E+03	**2.2197E+03**
	Std	1.0126E+01	8.3172E+00	3.2820E+00	5.8197E+00	4.5281E+00	3.5221E+00	3.7622E+00	3.5172E+00	1.1048E+01	2.4851E+03	4.9883E+00
	Best	2.2009E+03	2.2003E+03	2.2217E+03	2.2027E+03	2.2240E+03	2.2288E+03	2.2045E+03	2.2329E+03	2.2237E+03	2.2577E+03	2.2013E+03
	Rank	3	2	6	5	7	8	4	9	10	11	**1**
f_9	Mean	2.5685E+03	2.5616E+03	2.5519E+03	2.5635E+03	2.5381E+03	2.5911E+03	2.5411E+03	2.6259E+03	2.6514E+03	2.8967E+03	**2.5342E+03**
	Std	6.2132E+01	3.3369E+01	1.5224E+01	3.3941E+01	2.9769E+01	2.0748E+01	2.4430E+01	2.2246E+01	9.0816E+01	1.1200E+02	2.6826E+01
	Best	2.5293E+03	2.5293E+03	2.5322E+03	2.5293E+03	2.5293E+03	2.5492E+03	2.5293E+03	2.5765E+03	2.5293E+03	2.7187E+03	2.5293E+03
	Rank	7	5	4	6	2	8	3	9	10	11	**1**
f_{10}	Mean	2.5779E+03	2.5623E+03	2.5014E+03	2.5508E+03	2.5614E+03	2.5090E+03	2.5005E+03	2.5193E+03	2.7150E+03	3.1139E+03	2.5049E+03
	Std	6.0747E+01	5.5312E+01	3.7102E-01	6.0686E+01	1.0830E+02	2.2446E+00	9.8384E-02	4.0788E+00	3.6911E+02	3.8088E+02	2.5429E+01
	Best	2.5004E+03	2.5002E+03	2.5007E+03	2.5002E+03	2.5003E+03	2.5038E+03	2.5003E+03	2.5101E+03	2.5114E+03	2.7314E+03	2.5002E+03
	Rank	9	8	2	6	7	4	1	5	10	11	3
f_{11}	Mean	2.9289E+03	2.7781E+03	2.7622E+03	2.7347E+03	2.7975E+03	2.8166E+03	2.7328E+03	2.9143E+03	3.0738E+03	4.3162E+03	2.7387E+03
	Std	2.5708E+02	1.6447E+02	5.7004E+00	1.6705E+02	1.6178E+02	1.9077E+01	8.2918E+01	5.3038E+01	3.5111E+02	4.0934E+02	1.4721E+02
	Best	2.7668E+03	2.6001E+03	2.7449E+03	2.6003E+03	2.6002E+03	2.7696E+03	2.6028E+03	2.8215E+03	2.7914E+03	3.4543E+03	2.6006E+03
	Rank	9	5	4	2	6	7	1	8	10	11	3
f_{12}	Mean	2.8727E+03	2.8651E+03	2.8676E+03	2.8682E+03	2.8809E+03	2.8715E+03	2.8626E+03	2.8799E+03	2.8678E+03	3.2814E+03	2.8629E+03
	Std	2.4037E+01	5.4443E+00	1.3124E+00	7.3139E+00	2.3251E+01	1.7437E+00	1.6879E+00	5.4315E+00	2.5741E+00	1.7642E+02	1.8438E+00
	Best	2.8624E+03	2.8587E+03	2.8656E+03	2.8629E+03	2.8626E+03	2.8681E+03	2.8596E+03	2.8707E+03	2.8624E+03	2.9750E+03	2.8586E+03
	Rank	8	3	4	6	10	7	1	9	5	11	2
Total Rank		66	40	58	57	73	88	38	106	115	129	22
Final Rank		6	3	5	4	7	8	2	9	10	11	**1**



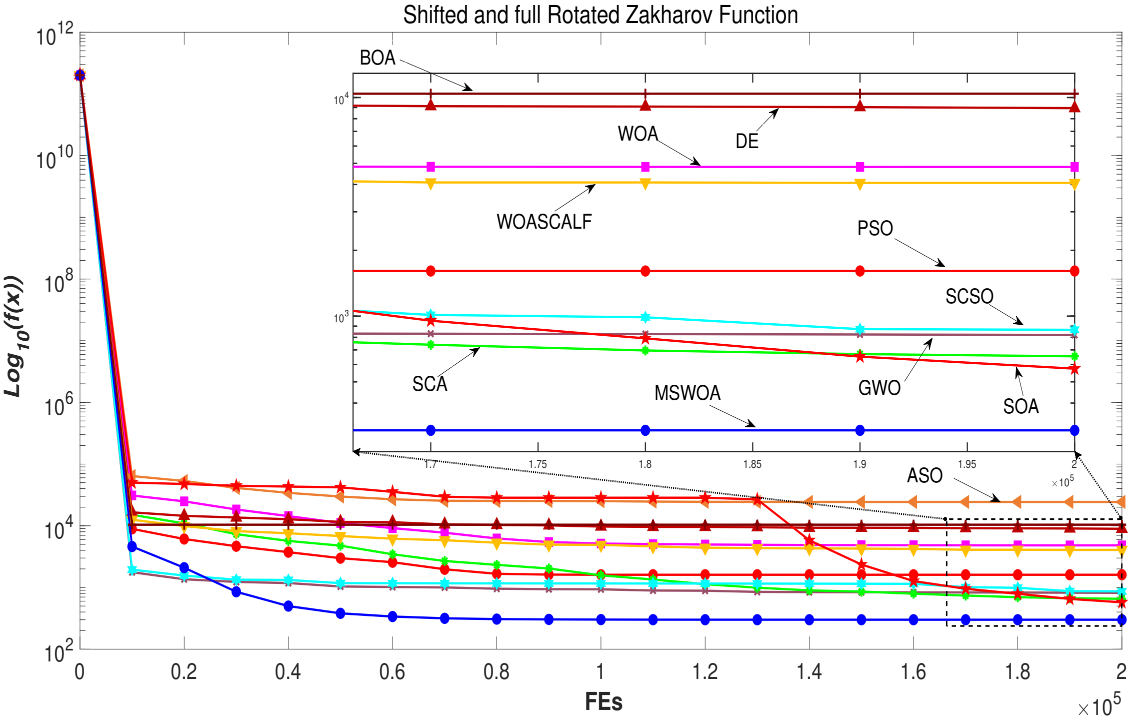
(a)

(b)

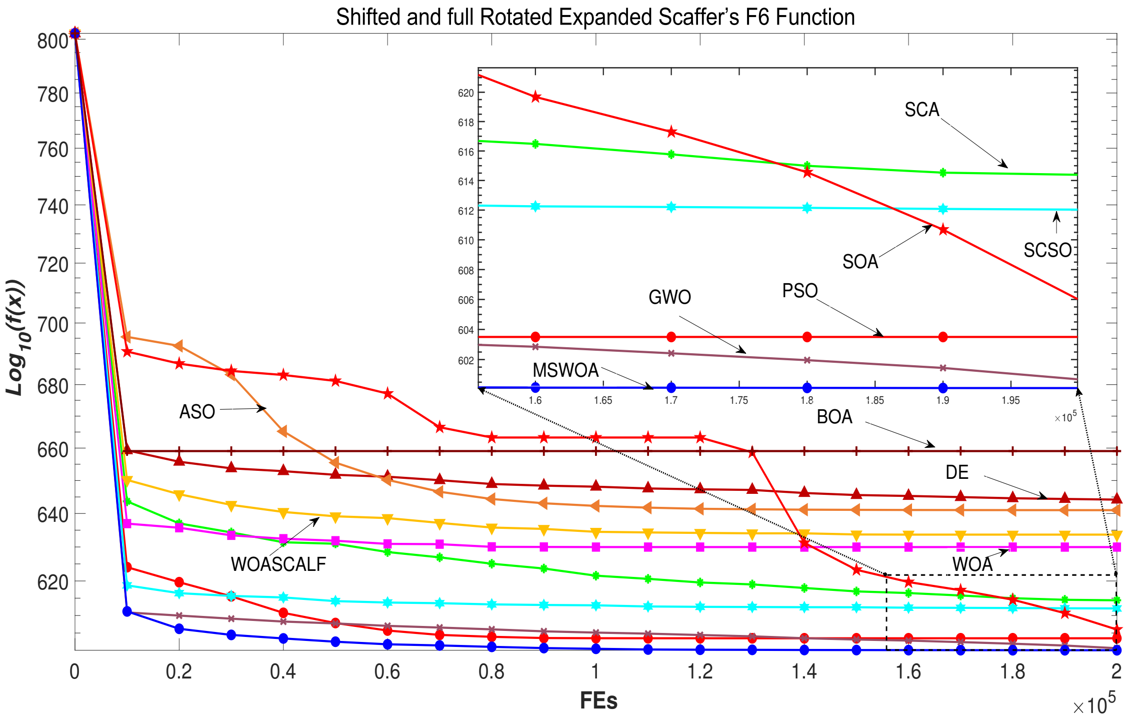
(c)

(d) Curve legend

Fig. 4: Qualitative results



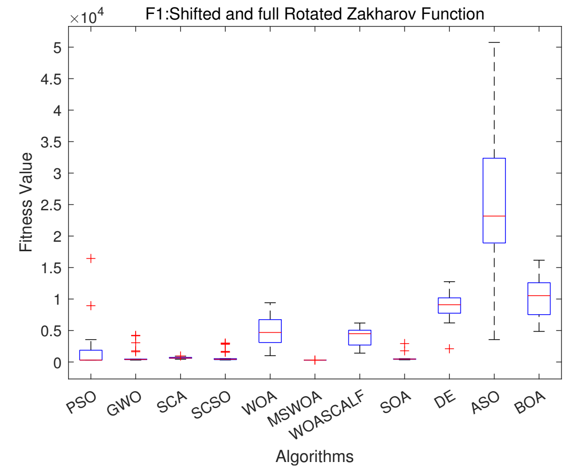
(a) f_1-$10D$



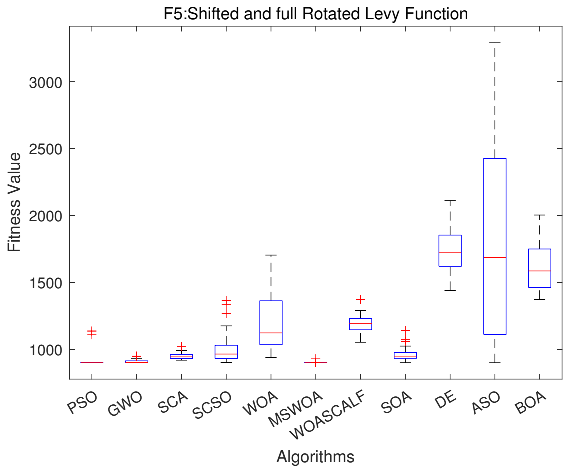
(b) f_3-$10D$

(c) Curve legend

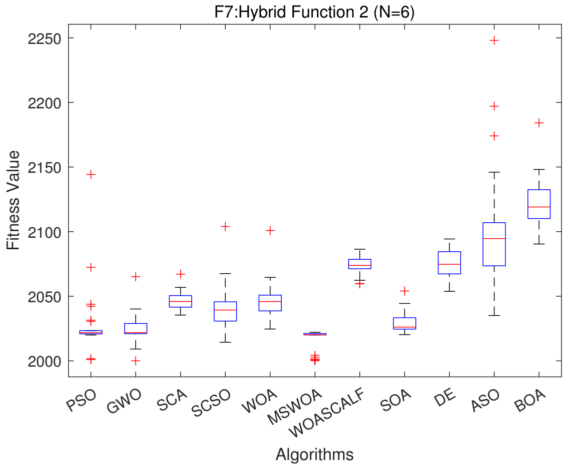
Fig. 5: Convergence curves



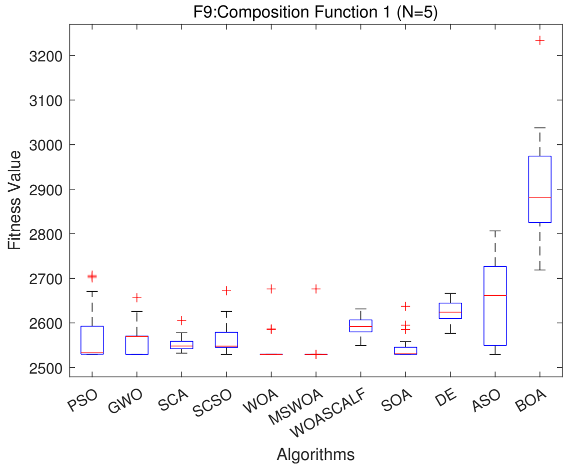
(a) f_1-10D



(b) f_5-10D



(c) f_7-10D



(d) f_9-10D

Fig. 6: Boxplots.

C. *Non-parametric statistics*

In this section, the experimental results of nonparametric statistics are presented, all of which were obtained within the given space. The statistical analysis in this paper includes the Friedman test and Wilcoxon signed rank test.

The Friedman test is used to determine if there are significant differences among multiple overall distributions [48]. In this paper, it was employed to examine whether MSWOA exhibits significant differences compared to other algorithms. In Table III, which demonstrate that MSWOA outperforms the other algorithms, indicating a significant difference among the 11 algorithms.

To further visualize whether MSWOA is better than other algorithms, the Wilcoxon signed rank test are examined [49]. In tables IV and V, the results are listed. R+ denotes the total ranking of our algorithm over the comparison algorithms, and R− means the sum of ranks where our algorithm underperforms. '+' indicates that MSWOA was superior to the comparison algorithm. '=' shows no obvious difierences among the two algorithms, and '−' signifies the MSWOA is inferior to the comparison algorithm at the 0.05 significance level [50]. The last row shows the sum of counts in the (+/ = /−) format. From Tables IV and V, it can be observed that the MSWOA has a notable preference over the other algorithms for most of the tested functions.

TABLE III: Friedman-test.

Algorithms	MeanRank	Ranking
PSO	5.5000	6
GWO	3.3333	3
SCA	4.8333	5
SCSO	4.7500	4
WOA	6.0833	7
WOASCALF	7.3333	8
SOA	3.1667	2
DE	8.8333	9
ASO	9.5833	10
BOA	10.7500	11
MSWOA	**1.8333**	**1**
p-value	1.1941E-14	

TABLE IV: Statistical comparisons of WSRT for MSWOA vs. PSO, GWO, SCA, SCSO and WOA on the $10D$.

fun	MSWOA vs. PSO				MSWOA vs. GWO				MSWOA vs. SCA				MSWOA vs. SCSO				MSWOA vs. WOA			
	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−
f_1	0.67328	253	212	=	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_2	6.34E-06	13	452	+	0.027029	125	340	+	6.34E-06	13	452	+	0.020671	120	345	+	0.89364	226	239	=
f_3	1.64E-05	23	442	+	0.54401	203	262	=	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_4	0.20589	294	171	=	4.07E-05	432	33	−	0.00057064	65	400	+	0.41653	193	272	=	0.00041955	61	404	+
f_5	0.82901	222	243	=	3.41E-05	31	434	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_6	0.025637	124	341	+	0.20589	171	294	=	1.73E-06	0	465	+	0.13059	306	159	=	0.0064242	365	100	−
f_7	0.0011138	74	391	+	0.00045336	62	403	+	1.73E-06	0	465	+	2.60E-06	4	461	+	1.73E-06	0	465	+
f_8	0.093676	151	314	=	0.097772	152	313	=	1.73E-06	0	465	+	0.0007157	68	397	+	1.73E-06	0	465	+
f_9	4.07E-05	33	432	+	2.84E-05	29	436	+	3.11E-05	30	435	+	2.84E-05	29	436	+	1.92E-06	1	464	+
f_{10}	5.22E-06	11	454	+	0.00017423	50	415	+	3.11E-05	30	435	+	7.51E-05	40	425	+	1.24E-05	20	445	+
f_{11}	1.13E-05	19	446	+	0.57165	205	260	=	3.33E-02	129	336	+	7.34E-01	249	216	=	1.66E-02	116	349	+
f_{12}	1.80E-05	24	441	+	7.73E-03	103	362	+	1.73E-06	0	465	+	5.31E-05	36	429	+	4.29E-06	9	456	+
Total				8/4/0				7/4/1				12/0/0				9/3/0				11/0/1

TABLE V: Statistical comparisons of WSRT for MSWOA vs. WOASCALF, SOA, DE, ASO and BOA on the $10D$.

fun	MSWOA vs. WOASCALF				MSWOA vs. SOA				MSWOA vs. DE				MSWOA vs. ASO				MSWOA vs. BOA			
	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−	p-Value	R+	R−	+/=/−
f_1	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_2	1.73E-06	0	465	+	0.082206	148	317	=	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_3	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_4	2.35E-06	3	462	+	0.093676	314	151	=	1.73E-06	0	465	+	2.88E-06	5	460	+	1.73E-06	0	465	+
f_5	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+	4.73E-06	10	455	+	1.73E-06	0	465	+
f_6	1.73E-06	0	465	+	0.027029	125	340	+	1.73E-06	0	465	+	0.00077122	69	396	+	1.73E-06	0	465	+
f_7	1.73E-06	0	465	+	2.13E-06	2	463	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_8	1.73E-06	0	465	+	3.41E-05	31	434	+	1.73E-06	0	465	+	1.73E-06	0	465	+	1.73E-06	0	465	+
f_9	1.97E-05	25	440	+	3.11E-05	30	435	+	1.92E-06	1	464	+	6.34E-06	13	452	+	1.73E-06	0	465	+
f_{10}	3.11E-05	30	435	+	6.32E-05	38	427	+	3.11E-05	30	435	+	9.32E-06	17	448	+	1.73E-06	0	465	+
f_{11}	0.0087297	105	360	+	0.61431	257	208	=	2.60E-05	28	437	+	1.49E-05	22	443	+	1.73E-06	0	465	+
f_{12}	1.73E-06	0	465	+	0.39333	274	191	=	1.73E-06	0	465	+	3.88E-06	8	457	+	1.73E-06	0	465	+
Total				12/0/0				8/4/0				12/0/0				12/0/0				12/0/0

TABLE VI: The results of welded beam design problem.

Algorithms	Mean	Std	Best
PSO	2.067303886	0.022490667	2.063119357
GWO	2.063563866	0.001859404	2.063129649
SCA	2.159383501	0.029624906	2.091213674
SCSO	2.063839457	0.002578851	2.063130534
WOA	2.931558806	1.020613137	2.080573141
WOASCALF	2.260926583	0.086858961	2.089275094
SOA	2.071467719	0.009486322	2.063349216
DE	2.684349581	0.268320409	2.20411344
ASO	3.102896539	0.597234512	2.213858995
BOA	3.011462187	0.776869775	2.292081596
MSWOA	2.085982605	0.022996344	2.063180561

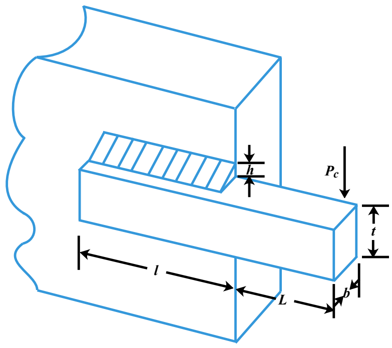


Fig. 7: Welded beam design problem.

TABLE VII: The results of tension/compression spring design problem.

Algorithms	Mean	Std	Best
PSO	0.014664715	0.002568925	0.012719054
GWO	0.013233665	0.001610042	0.012668234
SCA	0.012841261	9.41744E-05	0.012709034
SCSO	0.013297956	0.001593893	0.012666827
WOA	0.014000501	0.001820726	0.012665451
WOASCALF	0.013054561	0.00012049	0.01282136
SOA	0.012729666	2.22504E-05	0.012694641
DE	0.013102269	0.000163336	0.012771281
ASO	0.017522186	0.001404218	0.013328927
BOA	0.01653001	0.001627999	0.013601317
MSWOA	0.013399058	0.00161601	0.012676888

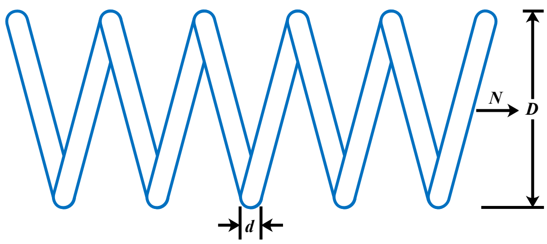


Fig. 8: Tension/compression spring design problem.

D. MSWOA in classical engineering problems

1) Welded beam design problem: In this case, a welded beam is designed with minimum cost under the constraint of shear stress (τ), bendding stress (σ) in the beam, buckling

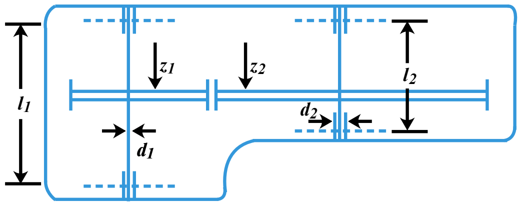


Fig. 9: Speed reducer design problem.

load on the bar (P_c), end deflection of the beam (δ), and side constraints [51]. Regarding the welded beam design problem, the corresponding results are given in Table VI. As observed from the table, most of the comparison algorithms in this paper struggle to effectively solve the problem. Additionally, the optimal values obtained by MSWOA are not in close proximity to the theoretical optimal values. However, MSWOA still achieves high rankings among all the comparison algorithms. It is important to note that not all algorithms can successfully solve every optimization problem, and there is still a need to explore and develop better algorithms for addressing this problem.

2) Tension/compression spring design problem: Arora [52] presented the design problem of a tension/compression spring with the aim of minimizing the weight of the extension/compression spring ($f(x)$) under the constraints of minimum deformation, shear stress, impact frequency, outer diameter limitation, and design variables. The results for the tension/compression spring design problem are presented in Table VII. From the table, it is evident that MSWOA is capable of finding solutions closer to the optimal solution. This indicates that MSWOA exhibits a relatively strong capability for tension/compression spring design problems.

3) Speed reducer design problem: The speed reducer design problem [53] is a classical problem, the weight of speed reducer is to be minimized subject to constraintson bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shafts. The results for the speed reducer design problems are presented in Table VIII. From the table, it is evident that both MSWOA and PSO achieve optimal solutions that are very close to the theoretical optimum in solving the reducer design problem. However, when comparing the mean and variance, MSWOA outperforms PSO significantly, indicating the proposed algorithm's high stability.

TABLE VIII: The results of speed reducer design problem.

Algorithms	Mean	Std	Best
PSO	103010.4262	305103.2303	2996.24304
GWO	3003.781232	3.673188731	2998.767315
SCA	3093.081023	30.63197038	3052.148065
SCSO	3003.530872	4.17133293	2997.108144
WOA	3269.985992	466.5589827	3006.329475
WOASCALF	3162.051397	41.23944616	3059.582591
SOA	3027.48693	14.00818364	3003.502511
DE	3077.220081	22.36541239	3024.583057
ASO	3176.003645	64.13939119	3051.591945
BOA	936938.2841	253619.1064	3362.31196
MSWOA	3000.549197	2.785161664	2996.733089

VI. Conclusion

In this paper, a multi-strategy Whale Optimization Algorithm (MSWOA) is proposed. This enhanced version incorporates several improvements, including dimensional updating, nonlinear convergence factor, global perturbation factor, firefly perturbation strategy, and vertical and horizontal crossover strategies. These enhancements are aimed at improving the performance of MSWOA. First, a strategy was devised to update each individual dimension differently. Second, a nonlinear convergence factor was introduced. Third, a global perturbation factor was incorporated into MSWOA, enhancing the diversity of populations during exploration. Fourth, a firefly perturbation strategy was employed to improve convergence accuracy. Finally, the vertical and horizontal crossover strategy was utilized to expedite convergence.The three qualitative indicators demonstrate that MSWOA excels in achieving a balance between exploration and exploitation. When compared to other algorithms, MSWOA exhibits superior performance. Additionally, the boxplots indicate that our algorithm's solution average and volatility are competitive with other algorithms. The Wilcoxon signed-rank test reveals that MSWOA outperforms at least 7 test functions. Moreover, the results of the Friedman's test indicate that our algorithm significantly outperforms other algorithms. In all three engineering design problems, our algorithm demonstrates favorable rankings, particularly in the speed reducer design problem where it significantly outperforms the compared algorithms. While no algorithm can solve all optimization problems, our algorithm exhibits greater competitiveness in addressing these engineering design problems.

In future work, it would be valuable to investigate whether the strategies are useful to other algorithms. Additionally, it is important to conduct more in-depth research. Furthermore, our algorithm requires further enhancements to effectively address a wider range of practical optimization problems.

References

[1] H. Tao, N. M. Al-Aragi, I. Ahmadianfar, M. H. Naser, R. H. Shehab, J. M. Zain, B. Halder, and Z. M. Yaseen, "Ranked-based mechanism-assisted biogeography optimization: Application of global optimization problems," *Advances in Engineering Software*, vol. 174, p. 103301, 2022.

[2] T. Kundu and H. Garg, "Lsma-tlbo: A hybrid sma-tlbo algorithm with lévy flight based mutation for numerical optimization and engineering design problems," *Advances in Engineering Software*, vol. 172, p. 103185, 2022.

[3] I. Attiya, M. Abd Elaziz, L. Abualigah, T. N. Nguyen, and A. A. Abd El-Latif, "An improved hybrid swarm intelligence for scheduling iot application tasks in the cloud," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6264–6272, 2022.

[4] J.-q. Li, J.-w. Deng, C.-y. Li, Y.-y. Han, J. Tian, B. Zhang, and C.-g. Wang, "An improved jaya algorithm for solving the flexible job shop scheduling problem with transportation and setup times," *Knowledge-Based Systems*, vol. 200, p. 106032, 2020.

[5] M. Squires, X. Tao, S. Elangovan, R. Gururajan, X. Zhou, and U. R. Acharya, "A novel genetic algorithm based system for the scheduling of medical treatments," *Expert Systems with Applications*, vol. 195, p. 116464, 2022.

[6] J.-X. Lv, L.-J. Yan, S.-C. Chu, Z.-M. Cai, J.-S. Pan, X.-K. He, and J.-K. Xue, "A new hybrid algorithm based on golden eagle optimizer and grey wolf optimizer for 3d path planning of multiple uavs in power inspection," *Neural Computing and Applications*, vol. 34, no. 14, pp. 11 911–11 936, 2022.

[7] H. Kundra, W. Khan, M. Malik, K. P. Rane, R. Neware, and V. Jain, "Quantum-inspired firefly algorithm integrated with cuckoo search for optimal path planning," *International Journal of Modern Physics C*, vol. 33, no. 02, p. 2250018, 2022.

[8] M. A. Javarone, "An evolutionary strategy based on partial imitation for solving optimization problems," *Physica A: Statistical Mechanics and its Applications*, vol. 463, pp. 262–269, 2016.

[9] S.-J. Song, C.-H. Qiu, L.-G. Peng, and S. Hu, “An assembly line multi-station assembly sequence planning method based on particle swarm optimization algorithm,” *Journal of Computers*, vol. 33, no. 1, pp. 115–125, 2022.

[10] E. K. Aydoğan, Y. Delice, U. Özcan, C. Gencer, and Ö. Bali, “Balancing stochastic u-lines using particle swarm optimization,” *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 97–111, 2019.

[11] H. Moayedi, H. Nguyen, and L. Kok Foong, “Nonlinear evolutionary swarm intelligence of grasshopper optimization algorithm and gray wolf optimization for weight adjustment of neural network,” *Engineering with Computers*, vol. 37, no. 2, pp. 1265–1275, 2021.

[12] X. Lü, Y. Wu, J. Lian, Y. Zhang, C. Chen, P. Wang, and L. Meng, “Energy management of hybrid electric vehicles: A review of energy optimization of fuel cell hybrid power system based on genetic algorithm,” *Energy Conversion and Management*, vol. 205, p. 112474, 2020.

[13] H. Khosravi, B. Amiri, N. Yazdanjue, and V. Babaiyan, “An improved group teaching optimization algorithm based on local search and chaotic map for feature selection in high-dimensional data,” *Expert Systems with Applications*, vol. 204, p. 117493, 2022.

[14] F. Kılıç, Y. Kaya, and S. Yildirim, “A novel multi population based particle swarm optimization for feature selection,” *Knowledge-Based Systems*, vol. 219, p. 106894, 2021.

[15] J. GUO, P. ZHAO, and J. NI, “Fire station location planning model based on genetic algorithm,” *Journal of Computer Applications*, 2020.

[16] O. I. Khalaf, G. M. Abdulsahib, and B. M. Sabbar, “Optimization of wireless sensor network coverage using the bee algorithm.” *Journal of Information Science and Engineering*, vol. 36, no. 2, pp. 377–386, 2020.

[17] P. D. Kusuma and F. C. Hasibuan, “Enriched coati osprey algorithm: A swarm-based metaheuristic and its sensitivity evaluation of its strategy.” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 2, pp. 277–285, 2024.

[18] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, “Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems,” *Advances in Engineering Software*, vol. 174, p. 103282, 2022.

[19] R. G. Reynolds, “An introduction to cultural algorithms,” in *Proceedings of The Third Annual Conference on Evolutionary Programming*, vol. 24. World Scientific, 1994, pp. 131–139.

[20] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “Gsa: a gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[21] P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing,” in *Simulated Annealing: Theory and Applications*. Springer, 1987, pp. 7–15.

[22] A. Y. Lam and V. O. Li, “Chemical-reaction-inspired metaheuristic for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381–399, 2009.

[23] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2001, vol. 2.

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

[25] F. Hemasian-Etefagh and F. Safi-Esfahani, “Group-based whale optimization algorithm,” *Soft Computing*, vol. 24, no. 5, pp. 3647–3673, 2019.

[26] R. Jiang, M. Yang, S. Wang, and T. Chao, “An improved whale optimization algorithm with armed force program and strategic adjustment,” *Applied Mathematical Modelling*, vol. 81, pp. 603–623, 2020.

[27] A. Kaveh and M. I. Ghazaan, “Enhanced whale optimization algorithm for sizing optimization of skeletal structures,” *Mechanics Based design of structures and Machines*, vol. 45, no. 3, pp. 345–362, 2017.

[28] C.-Y. Lee and G.-L. Zhuo, “A hybrid whale optimization algorithm for global optimization,” *Mathematics*, vol. 9, no. 13, p. 1477, 2021.

[29] S. Akyol and B. Alatas, “Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization,” *Physica A: Statistical Mechanics and its Applications*, vol. 540, p. 123094, 2020.

[30] J. Luo and B. Shi, “A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems,” *Applied Intelligence*, vol. 49, no. 5, pp. 1982–2000, 2019.

[31] Y. Sun and Y. Chen, “Multi-population improved whale optimization algorithm for high dimensional optimization,” *Applied Soft Computing*, vol. 112, p. 107854, 2021.

[32] X. Wu, S. Zhang, W. Xiao, and Y. Yin, “The exploration/exploitation tradeoff in whale optimization algorithm,” *IEEE Access*, vol. 7, pp. 125 919–125 928, 2019.

[33] S. Chakraborty, A. K. Saha, R. Chakraborty, M. Saha, and S. Nama, “Hswoa: An ensemble of hunger games search and whale optimization algorithm for global optimization,” *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 52–104, 2022.

[34] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[35] T. Krink, J. S. VesterstrOm, and J. Riget, “Particle swarm optimisation with spatial particle extension,” in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1474–1479.

[36] I. Fister, I. Fister Jr, X.-S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” *Swarm and Evolutionary Computation*, vol. 13, pp. 34–46, 2013.

[37] A.-b. Meng, Y.-c. Chen, H. Yin, and S.-z. Chen, “Crisscross optimization algorithm and its application,” *Knowledge-Based Systems*, vol. 67, pp. 218–229, 2014.

[38] A. Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi, and P. N. Suganthan, “Problem definitions and evaluation criteria for the cec 2022 special session and competition on single objective bound constrained numerical optimization,” *Technical Report*, 2021.

[39] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[40] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[41] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[42] S. Mirjalili, “Sca: a sine cosine algorithm for solving optimization problems,” *Knowledge-based systems*, vol. 96, pp. 120–133, 2016.

[43] 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, pp. 283–304, 2019.

[44] G. Dhiman and V. Kumar, “Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems,” *Knowledge-Based Systems*, vol. 165, pp. 169–196, 2019.

[45] S. Arora and S. Singh, “Butterfly optimization algorithm: a novel approach for global optimization,” *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.

[46] A. Seyyedabbasi, “Woascalf: A new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems,” *Advances in Engineering Software*, vol. 173, p. 103272, 2022.

[47] A. Seyyedabbasi and F. Kiani, “Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems,” *Engineering with Computers*, pp. 1–25, 2022.

[48] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[49] H. Liu, X.-W. Zhang, and L.-P. Tu, “A modified particle swarm optimization using adaptive strategy,” *Expert Systems with Applications*, vol. 152, p. 113353, 2020.

[50] P. Sun, H. Liu, Y. Zhang, L. Tu, and Q. Meng, “An intensify atom search optimization for engineering design problems,” *Applied Mathematical Modelling*, vol. 89, pp. 837–859, 2021.

[51] C. A. Coello Coello, “Use of a self-adaptive penalty approach for engineering optimization problems,” *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[52] A. D. Belegundu and J. S. Arora, “A study of mathematical programming methods for structural optimization. part i: Theory,” *International Journal for Numerical Methods in Engineering*, vol. 21, no. 9, pp. 1583–1599, 1985.

[53] L. Abualigah, M. A. Elaziz, A. M. Khasawneh, M. Alshinwan, R. A. Ibrahim, M. A. Al-Qaness, S. Mirjalili, P. Sumari, and A. H. Gandomi, “Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results,” *Neural Computing and Applications*, vol. 34, no. 6, pp. 4081–4110, 2022.