Improved Egret Swarm Optimization Algorithm with Multiple Strategies

Wu Bixia

School of Big Data and Statistics, Guizhou University of Finance and Economics, Guiyang, Guizhou Province

Received: 05 September 2024/ Revised: 13 September 2024/ Accepted: 19 September 2024/ Published: 30-09-2024

Abstract— *The Egrets Swarm Optimization Algorithm is a recently proposed heuristic algorithm that simulates the hunting behavior of egrets. To address the limitations of the original algorithm, such as insufficient development capability and decreased population diversity, a multi-strategy improved Egrets Swarm Optimization Algorithm is proposed. First, in the population initialization phase, Logistic chaotic mapping is introduced to generate chaotic sequences, enriching population diversity. Next, a dynamic perception factor is introduced to replace the step size factor in the idle strategy, which allows for more effective exploration and discovery of potential optimal solutions. Furthermore, to increase the breadth and depth of exploration, a crayfish foraging strategy is incorporated into the random walk strategy, and a roulette wheel strategy is added to the encircling mechanism to enhance the algorithm's search ability and avoid ineffective actions. Additionally, a distribution estimation strategy and a new exploration and exploitation strategy based on whale spiral ascent are introduced to improve the overall efficiency and functionality of the algorithm. Finally, testing on 20 classical functions shows that the improved algorithm enhances optimization performance. The algorithm is also applied to solve engineering constraint problems, demonstrating its practicality.*

Keywords— Egret Optimization Algorithm; Roulette Wheel Strategy; Distribution Estimation Strategy; Spiral Ascent Strategy; Crayfish Foraging Strategy.

I. INTRODUCTION

In recent years, with the increasing complexity of problems and the ambiguity of the final results, the demand for optimization algorithms has grown. Optimization problems can be represented as a continuous or combinatorial design search space, where the process involves finding the maximum or minimum of a function.

Metaheuristic algorithms are a class of heuristic methods based on natural phenomena or species behaviors [1], designed to solve optimization problems by simulating the behaviors of organisms or species in nature [2]. These algorithms typically exhibit some degree of randomness and adaptability, enabling them to search for optimal or near-optimal solutions within the search space [3]. Their inspiration comes from various biological phenomena, such as animal collective behavior, plant growth patterns, and microbial reproduction methods. According to the No Free Lunch theorem [4], many metaheuristic algorithms have emerged, including the Zunhai Qiao algorithm [5], Artificial Bee Colony algorithm [6], Butterfly Optimization Algorithm [7], Grasshopper Optimization Algorithm [8], Golden Sine Algorithm [9], Slime Mold Optimization Algorithm [10], Seagull Optimization Algorithm [11], Sparrow Search Algorithm [12], and Teaching-Learning-Based Optimization Algorithm [13]. Optimization problems are widespread across various fields, including engineering optimization, economics, logistics planning, machine learning, and artificial intelligence. Traditional optimization methods often face difficulties in solving complex high-dimensional, nonlinear, multimodal problems, whereas metaheuristic algorithms can effectively address various complex issues in the real world.

The Egret Swarm Intelligence Optimization Algorithm is an optimization algorithm designed based on the foraging behavior of egrets [14], simulating the collaborative and competitive behaviors of egret flocks during foraging to achieve optimization search. However, the Egret Swarm Intelligence Optimization Algorithm has some drawbacks, such as strong local convergence, sensitivity to parameters, and slow convergence speed. To address these issues, scholars have proposed improvements, such as the Sine-Cosine Egret Swarm Optimization Algorithm [15]. To enhance the ESOA algorithm, this paper combines strategies from the Whale Optimization Algorithm [16] and the Crawfish Optimization Algorithm [17], as well as strategies from the Hybrid Artificial Bee Colony Algorithm [18], to improve its search efficiency and optimization performance. The aim is to make the Egret Swarm Intelligence Optimization Algorithm more robust and efficient, thereby better applying it to practical engineering and scientific problems. This paper will present our improvement plan for the Egret Swarm Intelligence Optimization Algorithm and validate the effectiveness and performance advantages of the improved algorithm through a series of test functions [18][19][20] and experimental comparative analysis.

II. EGRETS SWARM OPTIMIZATION ALGORITHM

The Egrets Swarm Optimization Algorithm (ESOA) is an intelligent optimization algorithm introduced by Zuyan Chen and colleagues in 2022. It simulates the collective behavior of egrets during their hunting process. Egrets typically forage in groups, collaborating and competing in wetland environments to capture food, aiming to achieve the best foraging outcome. In the Egrets Swarm Optimization Algorithm, there are two main strategies: waiting and attacking. Suppose an egret team consists of three egrets: Egret A employs a guiding mechanism, Egret B uses random walk, and Egret C adopts an encircling mechanism. The behaviors of these egrets are quantified using corresponding mathematical models, as follows:

2.1 Waiting Strategy:

The observation equation for the i-th egret A can be described as: $y = f(x_i)$, where x_i represents the current position of the egret, and y denotes the assessment of potential prey at this position by Egret A. The parameterized assessment equation is given by Equation (1), where w_i is the weight of the assessment method, updated according to Equation (4). e_i represents the assessment error, and $\hat{g}_i$ is obtained by taking the partial derivative of e_i through Equation (2), representing the actual gradient of w_i.

$$\begin{aligned} \hat{y}_i &= \omega_i \cdot x_i \\ e_i &= \frac{\| \hat{y}_i - y_i \|^2}{2} \end{aligned} \tag{1}$$

$$\begin{aligned} \hat{g}_i &= \frac{\partial \hat{e}_i}{\partial w_i} = \frac{\partial \| \hat{y}_i - y_i \|^2 /2}{\partial w_i} = (\hat{y}_i - y_i) \cdot x_i \\ \hat{d}_i &= \hat{g}_i / | \hat{g}_i | \end{aligned} \tag{2}$$

$$\begin{aligned} \hat{g}_i &= \frac{\partial \hat{e}_i}{\partial w_i} = \frac{\partial \| \hat{y}_i - y_i \|^2 /2}{\partial w_i} = (\hat{y}_i - y_i) \cdot x_i \\ \hat{d}_i &= \hat{g}_i / | \hat{g}_i | \end{aligned} \tag{3}$$

$$\begin{aligned} \hat{g}_i &= \frac{\partial \hat{e}_i}{\partial w_i} = \frac{\partial \| \hat{y}_i - y_i \|^2 /2}{\partial w_i} = (\hat{y}_i - y_i) \cdot x_i \\ \hat{d}_i &= \hat{g}_i / | \hat{g}_i | \end{aligned} \tag{4}$$

Egret A achieves more stable rewards with lower energy consumption by constructing a pseudo-gradient estimator and updating its position based on historical experience. In Equation (3), d_{ibest} denotes the optimal direction within the team, d_{gbest} represents the optimal direction of all teams, while $d_{h,i}$ is the directional correction for the best position within the team, and $d_{h,i}$ is the directional correction for the best position among all teams. Subsequently, the pseudo-gradient of the observation equation weight is calculated using Equation (5), and finally, Egret A's position and fitness are updated based on Equation (6).

Here, t is the current iteration count, $t_{\max}$ is the maximum iteration count, hop is the range of the solution space (i.e., the difference between upper and lower bounds), $step_a$ is the step size factor for Egret A, set to 0.1, and $x_{a,i}$ is the expected next position of Egret B.

$$g_i = (1 - r_h - r_g) \cdot \hat{d}_i + r_h \cdot d_{h,r} + r_g \cdot d_{g,i} \tag{5}$$

$$x_{a,i} = x_i + step_a \cdot exp(- t/(0.1 \cdot ti_{max} \tag{6}$$

$$y_{a,i} = f(x_{a,i}) \tag{7}$$

2.2 Attacking Strategy:

Egret B uses a high-energy-consuming random search strategy, which results in higher potential rewards. Its position and fitness are updated according to Equation (9), where $r_{b,i}$ is a random number in $(-\frac{\pi}{2},\frac{\pi}{2})$, $step_b$ is the step size factor for Egret B, and $x_{b,i}$ is the expected next position of Egret B. Egret C employs an encircling mechanism; once it locates prey, it will pursue it until captured. The position and fitness updates are given by Equation (12), where D_h is the difference matrix between the team's optimal position and the current position, D_g is the difference matrix between the optimal position of all teams and the current position, and r_h and r_h are random numbers in $[0,0.5)$, with $x_{c,i}$ being the expected next position of Egret B.

$$\begin{aligned} x_{b,i} &= x_i + step_b \cdot \tan(r_{b,i}) \cdot hop / (1+t) \\ y_{b.i} &= f(x_{b,i}) \end{aligned} \tag{8}$$

$$\begin{aligned} D_h &= x_{ibest} - x_i \\ D_g &= x_{gbest} - x_i \\ x_{c,i} &= (1 - r_h - r_g) \cdot x_i + r_h \cdot D_h + r_g \cdot D_g \\ y_{c,i} &= f(x_{c,i}) \end{aligned} \tag{9}$$

2.3 Discrimination Conditions:

After the egret team members compute their expected positions, they will determine the updated positions of the egret team based on the discrimination condition in Equation (10). Specifically, the expected positions and fitness of the four egrets are compared with their positions and fitness from the previous iteration. If any egret's expected position has better fitness than the previous iteration, it will be adopted for updating. Conversely, if all egrets' expected positions have worse fitness than before, a response is made based on a random number r. If r is less than 0.3, there is a 30% chance of accepting a worse solution for position updating.

$$\begin{aligned} x_{s,i} &= [x_{a,i}, x_{b,i}, x_{c,i}, x_{d,i}] \\ y_{s,i} &= [y_{a,i}, y_{b,i}, y_{c,i}, y_{d,i}] \\ c_i &= \arg\min(y_{s,i}) \\ x_i &= \begin{cases} x_{s,i} \mid_{c_i}, if \quad y_{s,i} < y_i \quad or \quad r < 0.3 \\ \qquad x_i, else \end{cases} \end{aligned} \tag{10}$$

III. IMPROVED EGRET SWARM OPTIMIZATION ALGORITHM

Through the analysis of the ESOA algorithm, it is evident that within an egret team comprising Egret A, Egret B, and Egret C, each egret employs a different strategy for hunting activities. Egret A uses a waiting strategy, where each position update depends on changes in fitness values and is largely influenced by the optimal values of the egret team and the egret population. Egret B, which employs a random walk, symbolizes the aggressive predatory behavior of egrets in nature. While it has a higher probability of achieving greater rewards, it also consumes a significant amount of energy. Egret C, which uses an encircling mechanism, is also prone to getting trapped in local optima.

Therefore, in the ESOA algorithm, the strategies of the egret team members exhibit a degree of blindness, which restricts the range of optimization results. The algorithm suffers from insufficient development capability, slow convergence speed, and a decline in population diversity.

To address these issues, this paper proposes multi-strategy improvements to the parallel framework of the ESOA algorithm, with the aim of enhancing the search capability of the improved Egret Swarm Optimization Algorithm (IESOA) and effectively balancing global and local search abilities.

3.1 Chaotic Local Search Strategy:

Logistic chaotic mapping is one of the simplest and most effective chaotic systems, easy to implement and apply. In this paper, Logistic chaotic mapping is introduced during the initialization of the population in the IESOA algorithm. The output of the Logistic chaotic mapping is used as a random factor to guide the search direction and strategy of the population, increasing the algorithm's randomness and diversity. As shown in Equation (11), where μ is a control parameter with a value of 3.9, the Logistic chaotic mapping exhibits high randomness and can generate high-quality random number sequences.

$$x(i+1) = \mu x(i)(1 - x(i)) \tag{11}$$

3.2 Dynamic Perception Factor:

The step size factor is crucial for finding the optimal solution. Therefore, this paper introduces a Dynamic Perception (DP) factor into the step size of the waiting strategy in the ESOA algorithm. The DP factor changes with each iteration of the algorithm, as described in Equation (12). It can enhance the global search capability of the algorithm in the early stages, expanding the search space, and improve the local search capability in the later stages, thereby enhancing the algorithm's ability to find the optimal solution.

$$DP = \begin{cases} a \cdot (1 - (\frac{t}{t_{\max}})^2), t > \frac{t_{\max}}{2}, \\ a \cdot (1 - \frac{t}{t_{\max}})^2, t \leq \frac{t_{\max}}{2}. \end{cases} \tag{12}$$

3.3 Crayfish Foraging Strategy:

In the crayfish foraging strategy, the mathematical model is as follows: crayfish calculate the size of food q according to Equation (13) and choose different foraging methods. Food x_{best} represents the optimal solution. When the size of q is suitable for the crayfish, it will move closer to the food. Equation (15) uses a combination of sine and cosine functions to simulate the alternating process. When q is too large and there is a significant difference between the crayfish and the optimal solution, the size of x_{best} should be reduced to bring it closer to the food and control the crayfish's feeding amount. Given the asynchronous nature of the egret swarm, the crayfish foraging phase strategy is introduced into the random walk strategy of the ESOA algorithm with a probability of $\frac{1}{2}$. Specifically, the egret swarm selects different approaches based on the fitness of the optimal individual, as described in Equations (15) or (16), to approach the optimal solution, thereby updating the expected position and fitness of Egret B.

Including the crayfish foraging phase strategy in the random walk strategy of the ESOA algorithm ensures that during the search process, individuals randomly choose a new position to move with a certain probability. This preserves the randomness and exploratory nature of the search process, allowing individuals to escape from local optima and find better solutions. It also enables the IESOA algorithm to progressively approach the optimal solution, enhancing its development capability and convergence ability.

$$q = c_3 \times rand * (\frac{f_i}{f_{ibest}}) \tag{13}$$

$$x_{best} = \exp(-\frac{1}{q}) \times f_{ibest}$$
$$l = 0.2 \times (\frac{1}{3\sqrt{2\pi}}) \times e^{\frac{-(u-25)^2}{18}} \quad (14)$$

$$x_{b,i} = x_i + x_{best} \times l \times (cos(2 \times \pi \times rand) - sin(2 \times \pi \times rand)) \quad (15)$$

$$x_{b,i} = (x_i - x_{best}) \times l + p \times rand \times x_i \quad (16)$$

$$y_{b,i} = f(x_{b,i}) \quad (17)$$

3.4 Roulette Wheel Strategy:

The roulette wheel strategy, originating from genetic algorithms, essentially simulates the process of "survival of the fittest" in biological evolution. The fitness value of an individual serves as an indicator of how well the individual "adapts." Considering a population of np individuals, each with its own fitness function value, the probability of an individual being selected is defined by Equation (18). The higher the fitness value of an individual, the greater its probability of being selected, which is the principle behind the roulette wheel strategy. Compared to pure random search algorithms, this strategy has the advantage of more effectively utilizing the fitness information of individuals, thereby improving the algorithm's search efficiency and quality.

In the ESOA algorithm, for egret C when searching for prey, individuals adjust their speed and position based on the information of the global optimal position and their own optimal position to approach the optimal solution. This encircling mechanism helps the algorithm better balance exploration and the use of existing information during the search process, improving both global search capability and convergence speed. In the IESOA algorithm presented in this paper, the roulette wheel strategy with a stochastic element is introduced with a probability of $\frac{1}{2}$ during the position update process of egret C. This increases the likelihood of selecting individuals with higher fitness, making the IESOA algorithm more targeted, aiding in retaining high-quality individuals, enhancing the algorithm's convergence and global search capability, while also introducing some randomness to maintain population diversity and avoid local optima.

$$p_i = \frac{f_i}{\sum_{k=1}^{np} f_k} \quad (18)$$

3.5 Distribution Estimation Strategy:

The distribution estimation strategy is a random optimization algorithm based on experimental analysis, guiding population evolution through the construction of probabilistic models, sampling, and updating operations. Its main idea is to guide the search process by estimating the distribution of solutions in the problem space. Specifically, the strategy dynamically adjusts the search direction and strategy based on the known distribution of solutions to explore the solution space more effectively. This accelerates the algorithm's convergence speed, improves its global search capability, and enhances the algorithm's adaptability and robustness.

In the ESOA algorithm, the distribution estimation strategy is introduced with a probability of $\frac{1}{2}$ to simulate the position update process of egret D. During this process, the top $\frac{1}{2}$ individuals from the advantageous population are selected, avoiding the reduction in population diversity and the risk of falling into local optima caused by various strategies aimed at approaching the optimal solution. Additionally, a normal distribution estimate is incorporated into the distribution estimation strategy to further reduce the likelihood of the IESOA algorithm falling into local optima. Overall, introducing the distribution estimation strategy into the IESOA algorithm helps avoid local optima, enhances global search capability, and speeds up the convergence rate by facilitating quicker identification of regions close to the optimal solution, thus improving the algorithm's efficiency and performance and making it more suitable for complex optimization problems.

The mathematical model of this strategy is as follows: np represents the population size, and x_i denotes the top $\frac{np}{2}$ promising solutions ranked by fitness values from high to low. In Equation (19), ω_i represents the weight coefficients in descending order of fitness values in the advantageous population, with larger weights indicating higher ranks. In Equation (20), x_{mean} is the weighted average of the solutions from the top $\frac{np}{2}$ advantageous individuals. According to Equation (21), cov represents the weighted covariance matrix of the advantageous population, while Equation (22) y represents a random number that follows a normal distribution with mean $(0, \text{cov})$. Finally, Equation (22) is used to update the expected position and fitness of egret D.

$$\omega_i = \frac{ln(np/2+0.5)-ln(i)}{\sum_{j=1}^{np/2}(ln(np/2+0.5)-ln(j))} \tag{19}$$

$$x_{mean} = \sum_{j=1}^{np/2}(\omega_j \cdot x_i) \tag{20}$$

$$\begin{aligned} mean &= (x_{ibest} + x_{mean} + x_i)/3 \\ cov &= \frac{1}{np/2}\sum_{j=1}^{np/2}((x_i - x_{mean})\cdot(x_j - x_{mean})^T) \end{aligned} \tag{21}$$

$$\begin{aligned} x_{d,i} &= \begin{cases} mean + y, & p < 0.5 \\ x_{mean} + rand(x_{mean} - x_i), & p \geq 0.5 \end{cases} \\ f_{d,i} &= f(x_{d,i}) \end{aligned} \tag{22}$$

3.6 Spiral Ascending Strategy:

The spiral ascending strategy is used in the whale optimization algorithm for spiral position updates during bubble-net attacks, where whales continuously approach the optimal position through spiral swimming. In the IESOA algorithm, the spiral ascending strategy is introduced with a probability of $\frac{1}{2}$ to simulate the position update process of egret D, enhancing the algorithm's global search capability and aiding in finding the global optimal solution. This strategy combines attraction and repulsion behavior patterns. In the attraction mode, the strategy aims to move towards the global optimal solution, while in the repulsion mode, it attempts to avoid local optima, balancing exploration and exploitation during the search process. This improves the algorithm's convergence speed, increases search diversity, and enhances its robustness and global search capability.

The mathematical model of this strategy is as follows: d' represents the distance between the whale's current position and the optimal value, b is a constant describing the logarithmic spiral shape, and b is a random number within the range of $[-1,1]$. Finally, Equation (23) is used to update the expected position and fitness of egret D.

$$\begin{aligned} d' &= | x_{gbest} - x_i | \\ x_{d,i} &= d' \times e^{bl} \times \cos(2\pi l) + x_{gbest} \end{aligned} \tag{23}$$

IV. FLOWCHART OF THE MULTI-STRATEGY IMPROVED EGRET SWARM OPTIMIZATION ALGORITHM

For the original ESOA algorithm, the improvements in the IESOA algorithm described in this paper mainly include:

1) In the ESOA algorithm, Egret A constructs a pseudo-gradient estimator to estimate the descending plane and searches based on the gradient of the cutting plane parameters. In the IESOA algorithm, a dynamic perception factor replaces the original step size factor of 0.1 in the waiting strategy of Egret A.
2) In the ESOA algorithm, Egret B performs global roaming. The IESOA algorithm introduces a shrimp foraging phase strategy with a certain probability to speed up convergence and efficiency.
3) In the ESOA algorithm, Egret C uses an enclosing mechanism for selective search based on better egret positions. The IESOA algorithm introduces a roulette strategy with a certain probability to avoid local optima.

4) The IESOA algorithm introduces distribution estimation and spiral ascent strategies with a probability of 1/2, simulating the position update process of Egret D to reduce the blindness in the actions of individual egrets in the original ESOA algorithm.

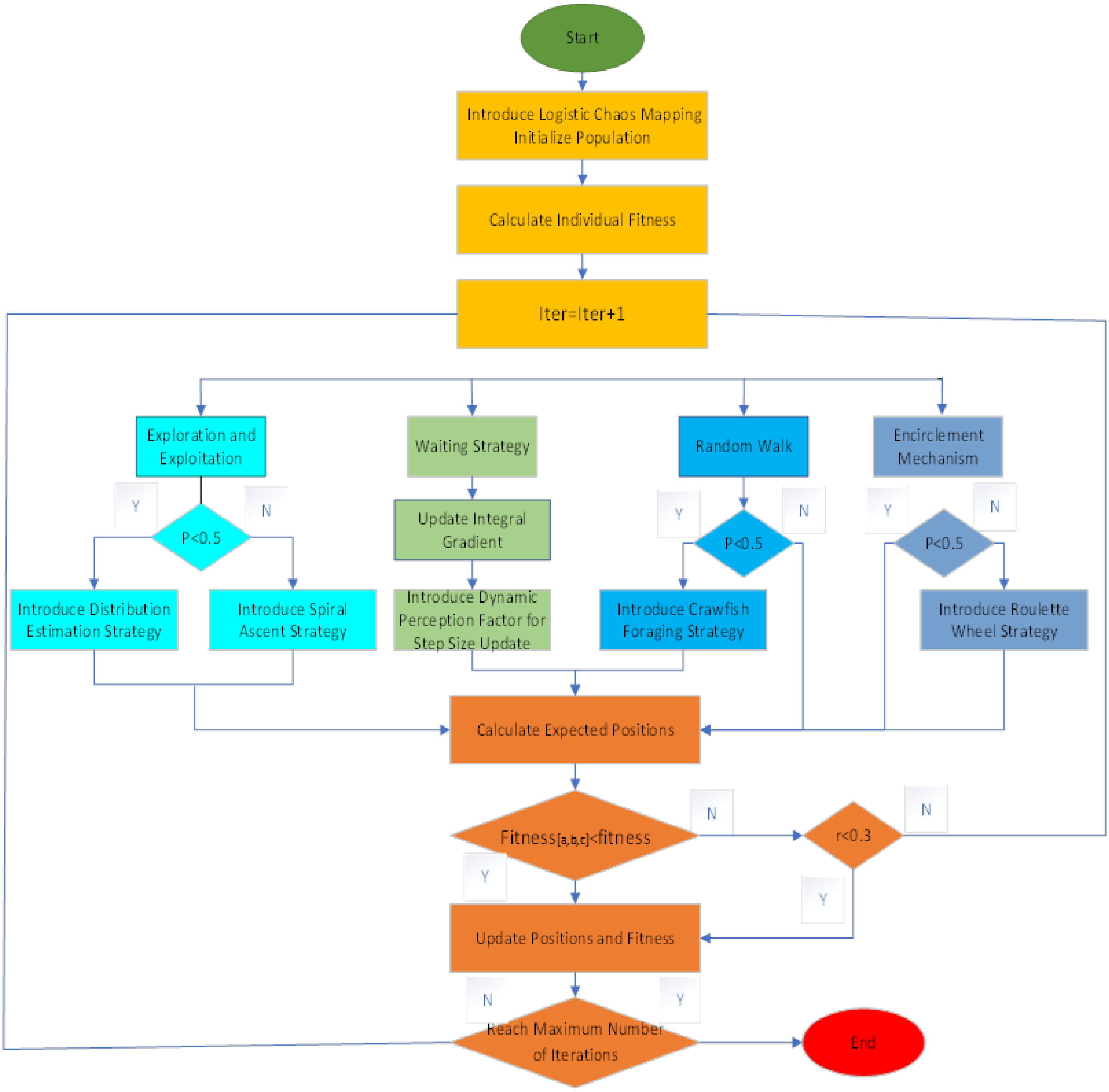

FIGURE 1: Flowchart of the IESOA Algorithm

V. SIMULATION EXPERIMENTS AND RESULTS ANALYSIS

This paper evaluates the convergence speed, solution accuracy, global search capability, and ability to avoid local optima of the multi-strategy improved Egret Swarm Optimization Algorithm (IESOA) using 20 classic test functions. Among these functions, F1 to F7 are single-peak functions used to assess convergence speed and solution accuracy; F8 to F19 and F21 are multi-peak functions containing multiple local optima and one theoretical optimum, used to evaluate global search capability and the ability to avoid local optima.

First, the IESOA algorithm was compared with the original ESOA algorithm using the 20 classic test functions, as shown in Figure 2. The results indicate that there is significant room for improvement in the original ESOA algorithm, with the IESOA algorithm demonstrating superior convergence speed and solution accuracy.

Next, the improved IESOA algorithm was independently tested 50 times on the 20 classic functions, alongside algorithms like BOA, GOA, GSA, and SOA. The average and standard deviation of 50 iterations for functions F1—F19 and F21 were calculated for each algorithm, and rankings for these 8 algorithms across the 20 functions were compared. As shown in Table-1, the IESOA algorithm achieved the best results for all 20 functions. For single-peak functions, the IESOA algorithm outperformed all other algorithms, showing strong development capability. For multi-peak functions, the IESOA algorithm maintained good population diversity and effectively avoided local optima, indicating its strong performance and potential for solving complex real-world problems.

Finally, to more intuitively demonstrate the performance or convergence of the IESOA algorithm, Figure 3 presents convergence curve plots of IESOA and nine other algorithms for the 20 classic test functions, analyzing convergence accuracy and efficiency. It can be seen that for single-peak functions F1-F7, the other nine algorithms prematurely fell into local optima, while the IESOA algorithm gradually converged to the global optimum. For multi-peak functions F8-F19 and F21, some functions showed multi-gradient descent trends, indicating the IESOA algorithm's excellent ability to escape local optima, and even exhibited unique local search or optimization performance in the early stages of iterations for some functions like F9. In summary, both in terms of convergence accuracy and speed, the IESOA algorithm demonstrates the best efficiency and performance.

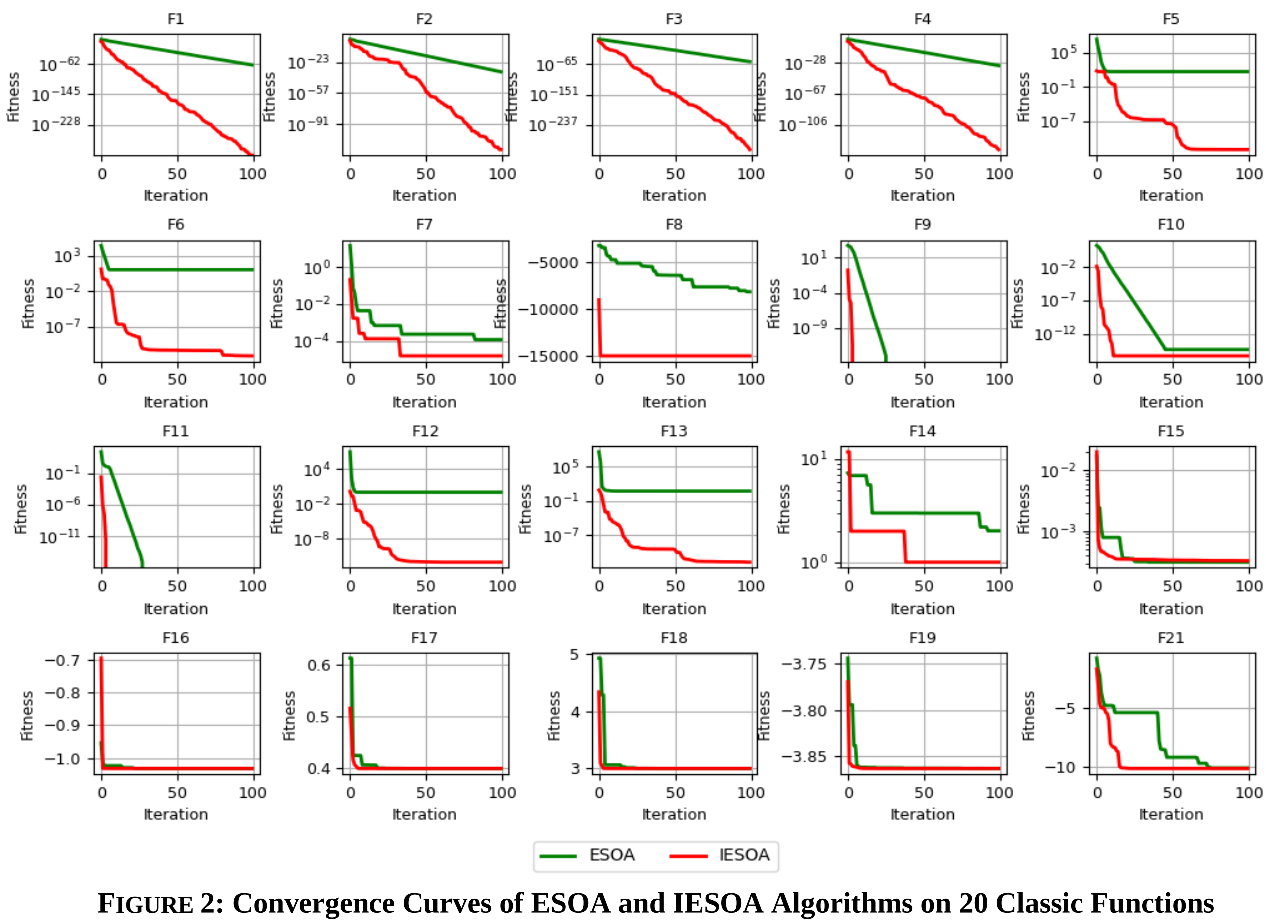

FIGURE 2: Convergence Curves of ESOA and IESOA Algorithms on 20 Classic Functions

TABLE 1
STATISTICAL RESULTS OF EIGHT ALGORITHMS ON 20 CLASSIC FUNCTIONS

		BOA	GOA	GSA	MFO	SOA	TLBO	WOA	IESOA
F1	Mean	0.000741	1910	2.46E-49	109000	2.91E-35	1910	0.0108	9.26E-200
	Std	0.0000529	649	1.28E-48	5900	1.97E-34	518	0.0245	0
F2	Mean	5.92E+23	42.8	8.75E-22	3.63E+19	2.09E-22	24	0.0101	3.47E-97
	Std	4.1E+24	28	6.11E-21	1.05E+20	1.3E-21	3.43	0.00712	2.43E-96
F3	Mean	0.000727	13600	3.48E-37	211000	245000	19600	2350	3.04E-177
	Std	0.000102	7850	2.44E-36	28600	81800	5070	2280	0
F4	Mean	0.0193	17.2	7.67E-23	88.5	0.915	21.8	4.26	5.97E-93
	Std	0.00136	2.93	4.96E-22	1.74	5.3	2.47	2.43	4.18E-92
F5	Mean	48.9	202000	0.204	410000000	18.3	276000	49.8	9.68
	Std	0.0309	119000	0.491	38700000	21.6	105000	1.67	19.4
F6	Mean	10.4	1730	0.00417	108000	0.635	2030	2.76	3.75E-12
	Std	0.715	679	0.00728	7740	0.561	495	0.647	9.79E-12
F7	Mean	0.000504	0.595	0.000525	327	0.000563	0.567	0.00638	0.0000672
	Std	0.000392	0.24	0.000496	32.9	0.000423	0.211	0.0049	0.0000626
F8	Mean	-3220	-10400	-20900	-6480	-20700	-6980	-11800	-19100
	Std	626	917	37.5	884	262	1010	515	3640
F9	Mean	191	270	0	721	0	174	16.8	0
	Std	174	50.3	0	24.1	0	40.5	15.2	0
F10	Mean	0.0136	9.17	4.44E-16	20.2	6.57E-16	9.59	0.196	4.44E-16
	Std	0.00076	1.17	9.86E-32	0.0993	8.44E-16	0.982	0.684	0
F11	Mean	0.00387	19	0	979	0.0195	19.3	0.0396	0
	Std	0.000452	5.47	0	67.1	0.136	4.25	0.0589	0
F12	Mean	0.906	21	0.000126	889000000	0.0176	30	0.19	0.000145
	Std	0.152	10.5	0.0002	129000000	0.0179	71.9	0.171	0.000946
F13	Mean	5.02	11200	0.000872	1770000000	0.281	50500	2.43	5.69E-12
	Std	0.158	22000	0.00144	198000000	0.189	83800	0.406	1.28E-11
F14	Mean	1.56	1.08	1.43	1.58	3.16	0.998	1	0.998
	Std	0.676	0.335	1.03	0.655	3.33	0.000116	0.0102	1.8E-14
F15	Mean	0.00058	0.00708	0.000448	0.00223	0.00149	0.000902	0.000645	0.000437
	Std	0.000319	0.011	0.00021	0.000707	0.00106	0.000318	0.000396	0.000179
F16	Mean	-0.609	-1.03	-1.02	-1.03	-1.03	-1.03	-1.03	-1.03
	Std	0.157	2.22E-16	0.0102	0.00494	0.0000237	0.000326	3.78E-08	2.93E-16
F17	Mean	0.403	0.398	0.404	0.403	0.402	0.398	0.398	0.398
	Std	0.0058	1.4E-09	0.0111	0.00974	0.0101	0.000362	0.00000428	0.000000095
F18	Mean	3.52	3	7.67	3.19	7.6	3	3	3
	Std	1.38	4.8E-09	10.1	0.432	10.6	0.0081	0.00000495	6.79E-15
F19	Mean	-3.42	-3.7	-3.77	-3.86	-3.75	-3.86	-3.86	-3.86
	Std	0.305	0.196	0.0791	0.00301	0.173	0.00086	0.00312	1.14E-09
F21	Mean	-0.588	-6.29	-10.1	-2.44	-6.85	-6.14	-5.35	-8.73
	Std	0.268	3.29	0.0364	0.988	3.04	2.4	1.12	2.29

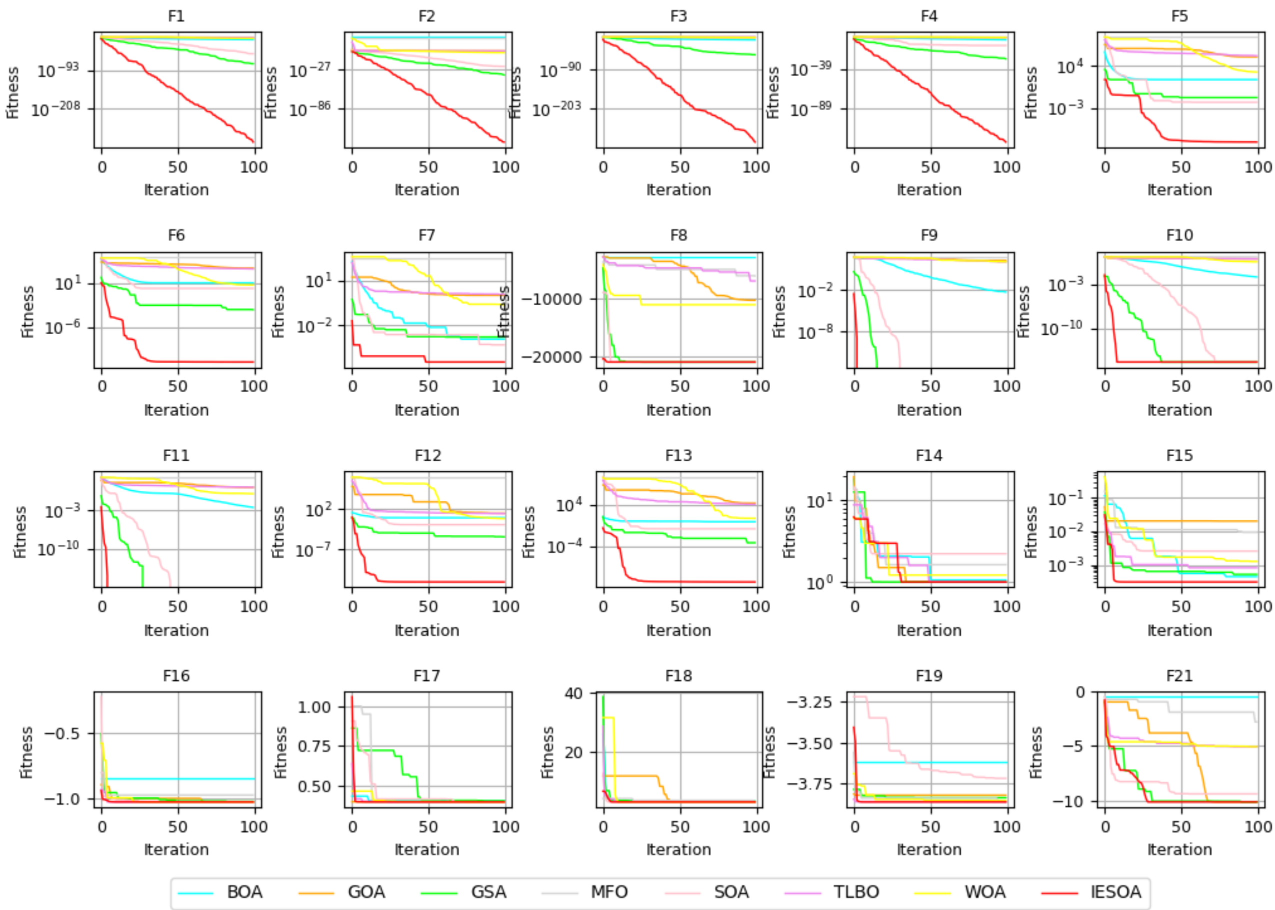

FIGURE 3: Convergence Curves of Eight Algorithms on 20 Classic

5.1 Engineering Constraint Problems:

5.1.1 Gearbox Design Problem:

In mechanical systems, a gearbox is one of the crucial components of a gear train and can be used in various applications. In this optimization problem, the goal is to minimize the weight of the gearbox under 11 constraints. This problem involves seven variables: face width $b = (x_1)$, gear module $m = (x_2)$, $z = (x_3)$, the length of the first shaft between bearings $l_1 = (x_4)$, and the diameter of the bearings $l_1 = (x_4)$ and the first shaft $d_1 = x_6$, $d_2 = x_7$. The mathematical model for this problem is expressed as follows:

a) Objective function:

$$f(X) = 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9334 x_3 - 43.0934) - 1.508 x_1 (x_6^2 + x_7^2)$$
$$+ 7.4777 (x_6^3 + x_7^3) + 0.7854 (x_4 x_6^2 + x_5 x_7^2)$$

b) Constraints:

$$g_1(X) = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0$$

$$g_2(X) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0$$

$$g_3(X) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \leq 0$$

$$g_4(X) = \frac{1.93x_5^3}{x_2 x_7^4 x_3} - 1 \leq 0$$

$$g_5(X) = \frac{\sqrt{(\frac{745x_4}{x_2 x_3})^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \leq 0$$

$$g_6(X) = \frac{\sqrt{(\frac{745x_5}{x_2 x_3})^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \leq 0$$

$$g_7(X) = \frac{x_2 x_3}{40} - 1 \leq 0$$

$$g_8(X) = \frac{5x_2}{x_1} - 1 \leq 0$$

$$g_9(X) = \frac{x_1}{12x_2} - 1 \leq 0$$

$$g_{10}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

c) Range of Values:

$2.6 \leq x_1 \leq 3.6$

$0.7 \leq x_2 \leq 0.8$

$x_3 \in \{17,18,19,\ldots,28\}$

$7.3 \leq x_4$

$x_5 \leq 8.3$

$2.9 \leq x_6 \leq 3.9$

$5 \leq x_7 \leq 5.5$

The optimization results of various algorithms are shown in Figure 4. It can be observed that, compared to the other seven optimization algorithms, the IESOA algorithm has the best accuracy and stability in solving the gearbox design problem. The optimal value obtained by the IESOA algorithm is approximately 2996.86, and the best solution provided by the IESOA algorithm is [3.500, 0.700, 17.000, 7.300, 7.800, 3.350, 5.287]. Therefore, IESOA achieves the best results in addressing this engineering problem.

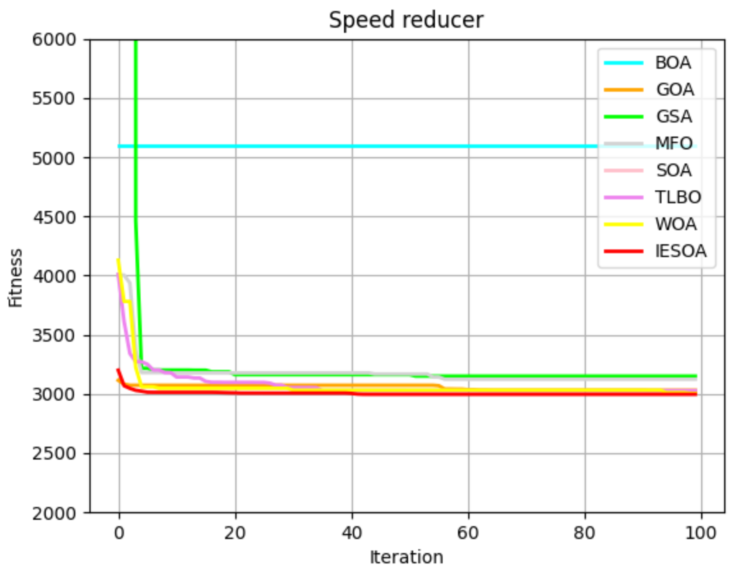

FIGURE 4: Comparison of Eight Optimization Algorithms in Solving the Gearbox Design Problem

TABLE 2
COMPARISON OF OPTIMAL SOLUTIONS FOR THE GEARBOX DESIGN PROBLEM

Algorithm	x1	x2	x3	x4	x5	x6	x7	Optimal weight
IESOA	3.5	0.7	17	7.3	7.8	3.35	5.287	2996.86
WOA	3.508	0.7	17	7.396	7.887	3.353	5.289	3004.73
TLBO	3.527	0.7	17	8.3	8.118	3.369	5.287	3027.89
SOA	3.501	0.7	17	7.398	7.864	3.565	5.299	3011
MFO	3.6	0.704	17.021	8.232	8.272	3.427	5.333	3124.31
GSA	3.502	0.7	17	8.125	7.8	3.794	5.311	3150.23
GOA	3.5	0.7	17	7.32	7.904	3.35	5.337	3031.02
BOA	3.54	0.701	25.862	7.931	7.916	3.744	5.354	5095.34

5.1.2 Tension/Compression Spring Design Problem:

In the tension/compression spring design problem, the optimization goal is to achieve the minimum spring mass using three variables and four constraints. The problem requires solving under constraints such as shear force, deflection, vibration frequency, and outer diameter. There are three design variables: coil diameter d , mean coil diameter D, and number of coils N . The variables in the design problem are coil diameter d , mean coil diameter D, and the number of effective coils N . The constraints include minimum deviation (g_1) , shear stress (g_2) , impact frequency (g_3) , and outer diameter limits (g_4) . By incorporating each variable into the constraints, the goal is to determine the minimum spring mass $f(x)$ .The mathematical model for this problem is given by equations (27)–(29):

d) Objective function:

$$f(x) = (x_3 + 2)x_2x_1^2$$

e) Constraints:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_4)} + \frac{1}{5108x_1^2} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

f) Range of Values:

$$0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2.0 \leq x_3 \leq 15$$

The optimization results of various algorithms are shown below in Figure 5. It can be observed that, compared to the other seven optimization algorithms, the IESOA algorithm has the best accuracy and stability when solving the tension/compression spring design problem. The best result obtained by the IESOA algorithm is approximately 0.0127, with the optimal solution being [0.050, 0.317, 14.087]. Therefore, IESOA achieves the best results in addressing this engineering problem.

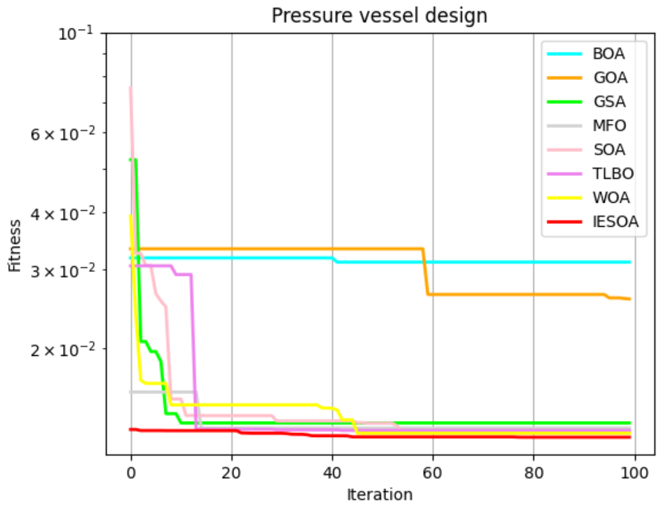

FIGURE 5: Comparison of Eight Optimization Algorithms in Solving the Tension Spring Design Problem

TABLE 3
COMPARISON OF OPTIMAL SOLUTIONS FOR THE TENSION SPRING DESIGN PROBLEM

Algorithm	x1	x2	x3	Optimal Cost
IESOA	0.05	0.317	14.087	0.0127
WOA	0.055	0.453	7.346	0.013
TLBO	0.05	0.31	15	0.0132
SOA	0.056	0.48	6.573	0.0131
MFO	0.05	0.314	15	0.0134
GSA	0.058	0.519	5.881	0.0137
GOA	0.064	0.738	6.481	0.0258
BOA	0.068	0.867	5.802	0.031

VI. CONCLUSION

The Egrets Swarm Optimization Algorithm (ESOA) is a new metaheuristic algorithm introduced in recent years. It features a straightforward and easy-to-understand principle, is user-friendly, and is suitable for integration with other metaheuristic algorithms to address complex problems involving high dimensions or multiple optima. To enhance the efficiency of this algorithm, this paper proposes an improved version of the Egrets Swarm Optimization Algorithm, known as the Improved Egrets Swarm Optimization Algorithm (IESOA).

This improvement incorporates a chaotic local search strategy to boost the performance of the original ESOA algorithm. To address the issue of single strategy blindness in parallel algorithms, it introduces the "lost phase" and "roulette wheel" strategies from the crayfish algorithm. Additionally, to increase population diversity, a new parallel strategy is added to the original ESOA algorithm, which includes distribution estimation algorithms and spiral ascent strategies. This new strategy helps better guide the population towards more optimal solutions.

Comparative experiments with 20 classic benchmark functions and nine other intelligent optimization algorithms demonstrate that the performance of the IESOA algorithm has significantly improved over the ESOA algorithm. It is hoped that the IESOA algorithm can be applied to more engineering problems in the future, providing a feasible solution approach for addressing various practical issues.

REFERENCES

[1] H. J H .Adaptation in Natural and Artificial Systems:An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence[M].The MIT Press:1992-04-29.

[2] Beheshti, Zahra, and Siti Mariyam Hj Shamsuddin. "A review of population-based meta-heuristic algorithms." Int. j. adv. soft comput. appl 5.1 (2013): 1-35.

[3] Singh P, Kottath R. An ensemble approach to meta-heuristic algorithms: comparative analysis and its applications[J]. Computers & Industrial Engineering, 2021, 162: 107739.

[4] WOLPERT D H, MACREADY W G. No free lunch theo-rems for optimization[J]. IEEE Transactions on Evolution-ary Computation, 1997, 1(1): 67-82.

[5] Mirjalili S, Gandomi A H, Mirjalili S Z, et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems[J]. Advances in engineering software, 2017, 114: 163-191.

[6] Karaboga D. Artificial bee colony algorithm[J]. scholarpedia, 2010, 5(3): 6915.

[7] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing, 2019, 23: 715-734.

[8] Meraihi Y, Gabis A B, Mirjalili S, et al. Grasshopper optimization algorithm: theory, variants, and applications[J]. Ieee Access, 2021, 9: 50001-50024.

[9] Tanyildizi E, Demir G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm[J]. Advances in Electrical & Computer Engineering, 2017, 17(2).

[10] Li S, Chen H, Wang M, et al. Slime mould algorithm: A new method for stochastic optimization[J]. Future Generation Computer Systems, 2020, 111: 300-323.

[11] Dhiman G, Kumar V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems[J]. Knowledge-based systems, 2019, 165: 169-196.

[12] Xue, J. (2020). Research and Application of a New Type of Swarm Intelligence Optimization Technology [Master's thesis, Donghua University]. DOI: 10.27012/d.cnki.gdhuu.2020.000178

[13] Rao R V, Savsani V J, Vakharia D P. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems[J]. Information sciences, 2012, 183(1): 1-15.

[14] Zuyan C ,Adam F ,Shuai L , et al.Egret Swarm Optimization Algorithm: An Evolutionary Computation Approach for Model Free Optimization.[J].Biomimetics (Basel, Switzerland),2022,7(4):144-144.

[15] Tianyi L ,Yanmei L ,Zhen C .Application of Sine Cosine Egret Swarm Optimization Algorithm in Gas Turbine Cooling System[J].Systems,2022,10(6):201-201.

[16] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[17] Jia H, Rao H, Wen C, et al. Crayfish optimization algorithm[J]. Artificial Intelligence Review, 2023, 56(Suppl 2): 1919-1979.

[18] Shen, Z., Wu, T., Zhou, Q., et al. (2023). Computing Offloading Strategy Based on Hybrid Artificial Bee Colony Algorithm in Mobile Edge Computing. Journal of Nanjing University of Information Science and Technology (Natural Science Edition), 1-13. Retrieved December 15, 2023, from https://doi.org/10.13878/j.cnki.jnuist.20230731005Mishra S K. Some new test functions for global optimization and performance of repulsive particle swarm method[J]. Available at SSRN 926132, 2006.

[19] Mishra S K. Some new test functions for global optimization and performance of repulsive particle swarm method[J]. Available at SSRN 926132, 2006.

[20] Molga M, Smutnicki C. Test functions for optimization needs[J]. Test functions for optimization needs, 2005, 101: 48.