*Journal of Network Intelligence* ©2024 ISSN 2414-8105 (Online)
**Taiwan Ubiquitous Information** Volume **9**, Number **3**, August **2024**

# An Improved Reptile Search Algorithm (IRSA)

Yu-Xin Wu*, An-Chen Wang

School of Electrical Engineering
Northeast Electric Power University
169 Changchun Road, Jilin 132012, China
wuyuxin1007@126.com, 13206832021@163.com

*Corresponding author: Yu-Xin Wu

Received September 25, 2023, revised January 16, 2024, accepted April 13, 2024.

Abstract. *This paper proposes an improved reptile search algorithm (IRSA) to overcome decreasing population diversity and convergent local optima in middle and late iterations. This paper presents improvements to updating the belly-walking position of reptiles throughout the exploration phase. Add an approach for learning from the best individuals and incorporate a greedy selection mechanism to expedite population convergence. Add roulette wheel selection alongside the proposition of a new hybrid mutation equation aimed at enhancing the overall diversity within the population. The failure restart mechanism disrupts the optimal solution site to refresh the population, boosting the algorithm's ability to escape local optima. IRSA is tested by simulating 28 CEC2013 test functions, and the results reveal that IRSA has faster convergence, higher accuracy, and better global optimization than the original algorithm.*

**Keywords:** reptile search algorithm, single objective optimization, metaheuristic, CEC2013 test

1. **Introduction.** Optimization technology is very nascent and has extensive application across various domains, including science, engineering, the economy, management, and industry. This technology studies the optimal solution of problems defined by mathematical methods [1]. Within the realm of optimization techniques, traditional methods are characterized by their arduous nature in solving complex problems that are large-scale, high-dimensional, and nonlinear in nature. These types of problems are commonly encountered in real-world scenarios, and their inherent complexity often results in the algorithm converging towards a local optimum solution [2]. Therefore, in this case, inspired by the principle of bionics, a new optimization method—metaheuristic optimization—has emerged.

Metaheuristic optimization techniques can be divided into four basic categories: The majority of optimization algorithms based on evolution draw inspiration from natural and biological evolution. These algorithms aim to simulate the evolutionary process observed in biology, namely the theory of natural selection, to solve optimization issues optimally. Representative algorithms include the Differential Evolution (DE) algorithm [3], the Biogeography-Based Optimizer (BBO) algorithm [4], the Evolutionary Programming (EP) algorithm [5], Genetic Programming [6](GP), Genetic Algorithm [7](GA), etc.

Physics-based optimization algorithms are based on natural phenomena, including gravity, inertia force, temperature, and electromagnetic fields. Common physics-based optimization algorithms: Simulated Annealing (SA) algorithm [8], Central Force optimization (CFO) algorithm [9], Black Hole (BH) algorithm [10], Water Wave optimization (WWO) algorithm [11], Galactic Swarm optimization (GSO) algorithm [12], Equilibrium Optimizer (EO) algorithm [13], etc.

Human-based algorithms encompass algorithms that emulate both physical and non-physical behaviors in humans to effectively identify optimal problem-solving approaches. Examples of such algorithms include the Imperialist Competitive algorithm (ICA) [14], Teaching-Learning-based Optimization optimization (TLBO) [15], and Social-Based algorithm (SBA) [16].

Swarm intelligence-based optimization algorithms simulate complex social behavior like information exchange and mutual cooperation between biological groups in nature to optimize. Among them, Ant Colony optimization (ACO) algorithm [17] is the most representative algorithm. Ants track and release pheromones during movement, and at the same time, the concentration of pheromones determines the probability of walking paths, forming a positive information feedback phenomenon. Through this kind of indirect communication mechanism, we can realise the shortest path and search targets together. Similarly, Particle Swarm optimization (PSO) algorithm [18], Artificial Bee Colony (ABC) algorithm [19], Bat algorithm [20](BA), Cuckoo Search (CS) Algorithm [21], Krill Herd Algorithm [22](KH), Firefly Algorithm [23](FA), Grey Wolf Optimizer (GWO) Algorithm [24], Whale optimization Algorithm [25](WOA), Squirrel Search Algorithm [26](SSA), etc. In 2022, Abualigah et al. [27] proposed the Reptile Search Algorithm (RSA).

Achieving higher optimization performance while minimizing computational resources is a key aspiration in swarm intelligence optimization algorithms. No algorithm is ideal for all optimization problems, in theory. Consequently, this compels researchers to seek out optimization strategies that are more efficient. In this context, this paper proposes an improved reptile search algorithm (IRSA) that aims to enhance both accuracy and speed of convergence when applied to difficult optimization problems. The primary aspects pertaining to innovation and motivation are as follows:

1. To improve the exploration phase. Improve the methodology for updating the reptile belly walk position during the encircling phase. Incorporation of the strategy of reptiles learning for their individual optimum is implemented to uphold population variety and enhance the pace of convergence.

2. To add a mutation mechanism. A new individual position update equation with hybrid mutations is proposed. The mutation population uses roulette wheel selection, which includes the global best, individual history best, and global worst. This method improves algorithm convergence speed and population variety, balancing performance.

3. To add a failure-restart mechanism. Once the number of failures surpasses a predetermined threshold, the restart equation is employed to perturb the global optimal individual as the central point, update the population, and increase population diversity, thereby enhancing the capacity to escape local optima and improving search precision.

The CEC 2013 test results show that on the test set with RSA and compared to other four kinds of typical optimization algorithms, the proposed IRSA in terms of convergence speed, precision, and stability has a significant advantage.

This paper's contents are as follows: Section 2 summarizes some improved algorithm and RSA algorithm research. Section 3 explains RSA and process principles. Section 4 analyses the drawbacks of the RSA algorithm and further proposes the IRSA algorithm. Section 5 displays the CEC2013 test set simulation results of the IRSA, RSA, and other prominent swarm intelligence algorithms. Section 6 summarizes this paper's algorithm.

## 2. **Related work.**

Numerous scholars have studied heuristic algorithms to improve their performance. In 2014, Wang et al. [28] proposed a biogeography-based approach to krill swarm optimization using a migration algorithm, which used standard KH to reduce search speed and added a krill migration (KM) operator to improve search efficiency, but the parameters needed to be adjusted to solve different problems. In 2015, Kiran et al. [29] proposed a varied search strategy artificial bee colony method, which selects one or more search strategies according to the characteristics of the numerical function to obtain better quality of the solution, but the algorithm exhibits suboptimal efficiency in terms of optimization. In 2016, Wang et al. [30] proposed an improved BAT algorithm, combining the mutation operator of the difference algorithm to reduce accurate evaluations of candidate solutions, but its application field is relatively limited. In 2017, Wang et al. [31] proposed a new variant of the firefly algorithm and established a new neighborhood attraction model to reduce the mutual attraction of fireflies, which can effectively reduce the oscillation in the process of calculation iteration and the complexity of calculation time, but using the same neighborhood size for different search stages is less flexible. In 2020, Hu et al. [32] came up with an Improved Binary Grey Wolf Optimizer to help with discrete issues like feature selection, looked at the AD range values of GWO in binary conditions, added a new transfer function, and offered a new equation for updating parameters based on the AD range value on the transfer function to balance global and local search and improve classification accuracy. In 2022, Shaik et al. [33] proposed Gaussian Mutation-Spider Monkey Optimization for feature selection. Gaussian variation changes where the solution is located after exploration, making it easier to use feature selection to pick out the right features for better classification and getting around the issues of overfitting and data imbalance, but the proposed model is computationally too complex. In 2023, Deng et al. [34] proposed a multi-strategy improved slime mushroom algorithm and proposed a new balanced exploration and exploitation formula that uses dynamic random search techniques to increase the adaptive mutation probability to sustain population variety and improve local optima breakout. Chen et al. [35] proposed a genetic algorithm that encodes multi-pathway genes using simulated annealing and chromosome creation and uses VNS search to identify solutions. Avoiding local optimal solutions increased performance, but the algorithm's convergence was inconsistent throughout the calculation, but the stability of the algorithm's convergence is not consistent throughout the calculation process.

For optimal reptile algorithm performance, some researchers have done some work. Almodfer et al. [36] proposes a quantum mutation reptile search algorithm that leverages the quantum mutation search strategy to enhance population diversity, achieve higher accuracy in identifying optimal solutions, and utilize the RSA algorithm for exploration and development. It gets to resolve the imbalance and speed up search convergence. Elgamal et al. [37] proposed an improved reptile search algorithm that utilizes simulated annealing as its underlying principle. The utilization of a circular chaotic map was implemented at the initialization step of the RSA in order to augment the algorithm's capacity for exploration within the search space. Yuan et al. [38] proposed an adaptive chaotic reverse learning variant RSA to optimize the starting population. A shift distribution estimation strategy is employed to regulate the individual evolution direction of population information. An elite alternative pool strategy is implemented to govern the reference points that follow the population, thereby achieving a balance between algorithm development and exploration capabilities, ultimately leading to improved overall algorithm performance. Huang et al. [39] proposed an interactive crossover reptile search algorithm. Interactive crossover strategies improve crossover operator efficiency through iterative approaches. This results in improved convergence accuracy and enhanced local exploration precision. The Lévy flight strategy eliminates premature convergence and enhances the global search algorithm.

Based on the aforementioned research conducted on RSA, compared to RSA, the improved RSA has better convergence time and accuracy. Nevertheless, there remains plenty of room for enhancing convergence performance in the resolution of intricate problems.

3. **Reptile Search Algorithm.** The foraging hunting behavior of reptiles, which hunt in packs as predators, served as the inspiration for the metaheuristic RSA. Reptile behaviour comprises two phases: encircling and hunting. RSA is a population-based and gradient-free algorithm that possesses the capability to effectively address optimization problems of varying complexity, subject to specific constraints. The pseudocode, as shown in Algorithm 1.

**Algorithm 1** RSA

**Input:** $N$, $dim$, $G$, $\alpha$, $\beta$
**Output:** The optimal solution and its fitness value
Initialize population ($X_1$, $X_2$, …,$X_i$, …, $X_N$)
**while** $g$<$G$ **do**
  Calculate each individual's fitness value of the population
  Find the optimal position so far
  Using Equation (6) to update $ES$
  **for** $i$=1 to $N$ **do**
    **for** $j$=1 to $dim$ **do**
      Using Equation (4, 5 and 7) to update parameters $\gamma$, $R$, and $P$
      **if** $g > \frac{G}{4}$ **then**
        Update position using the High Walking Equation (2)
      **else if** $g > \frac{G}{4}$ and $g < 2\frac{G}{4}$ **then**
        Update position using the Belly walking Equation (3)
      **else if** $g > 2\frac{G}{4}$ and $g < 3\frac{G}{4}$ **then**
        Update position using the Hunting coordination Equation (9)
      **else** $g > 3\frac{G}{4}$ and $g < G$
        Update position using the Hunting cooperation Equation (10)
      **end if**
    **end for**
  **end for**
  g=g+1
**end while**
Return the best solution.

3.1. **Initialization.** In RSA, $N$ candidate solutions are generated, each of dimension $dim$, and the $i$th solution is $X_i = [X_{i,1}, X_{i,2}, \ldots, X_{i,j} \ldots, X_{i,dim}]$. The initialized equation for the $i$th solution to the $j$th dimension is Equation (1).

$$X_{i,j} = X_j^{LB} + rand \times \left(X_j^{UB} - X_j^{LB}\right) \tag{1}$$

Where $X_j^{UB}$ is the upper boundary and $X_j^{LB}$ is the lower boundary. $rand$ is a uniformly distributed random number from $-1$ to 1.

3.2. **Encircling phase.** Based on reptile behavior, RSA has four stages. In the first two stages, namely the encircling phase, the fundamental purpose is to explore the high-density solution space for a more detailed exploration. The behavior of reptiles in mathematics can be modeled as Equations (2) and (3).

$$X_{(i,j)}(g+1) = X_j^{Best}(g) - \gamma_{(i,j)}(g) \times \beta - P_{(i,j)}(g) \times rand, \quad \text{if } g \leq \frac{G}{4} \tag{2}$$

$$X_{(i,j)}(g+1) = X_j^{Best}(g) \times X_{(r_1,j)}(g) \times ES(g) \times rand, \quad \text{if } g > \frac{G}{4} \ and \ g \leq \frac{2G}{4} \tag{3}$$

Where $X_{(i,j)}(g+1)$ is the updated position of the reptile. $X_j^{Best}(g)$ is the optimal position in the $j$th dimension of the reptile population in the $g$th iteration number. $\beta$ is the parameter that controls the sensitivity of search performance and has a fixed value of 0.005. $\gamma_{(i,j)}(g)$ represents the hunting operator of reptile, and the value is determined through the utilization of Equation (4).

$$\gamma_{(i,j)}(g) = X_j^{Best}(g) \times R_{(i,j)}(g) \tag{4}$$

Where $R_{(i,j)}$ is the current position's ratio to the global best position's $j$th dimensional position, and the value is calculated using Equation (5).

$$R_{(i,j)}(g) = \alpha + \frac{X_{(i,j)}(g) - M_{X_i}(g)}{X_j^{Best}(g) \times (X_j^{UB} - X_j^{LB}) + \epsilon} \tag{5}$$

Where $\alpha$ represents the parameter controlling the sensitivity of the exploration performance, and its value is fixed to 0.1. Use $\epsilon$ as a minimum constant to avoid zero division in the denominator. $M_{X_i}(g)$ denotes the mean position of the $i$th reptile in each dimension, as defined by Equation (6).

$$M_{X_i}(g) = \frac{1}{dim} \sum_{j=1}^{dim} X_{(i,j)}(g) \tag{6}$$

$P_{(i,j)}(g)$ reduces reptilian search space. The size is calculated using Equation (7).

$$P_{(i,j)}(g) = \frac{X_j^{Best}(g) - X_{(r_2,j)}(g)}{X_j^{Best}(g) + \epsilon} \tag{7}$$

Where $X_{(r_2,j)}(g)$ is the position of the random reptile. Exploration parameter $ES$ controls evolution direction and randomly chooses $-2$ to 2, which is a probability-based ratio whose value is calculated by Equation (8).

$$ES(g) = 2 \times r_3 \times (1 - \frac{1}{G}) \tag{8}$$

Where $r_3$ is a random number between $[-1, 1]$, 2 makes $ES$ deliver a value between [0, 2], and $G$ is the total number of iterations.

3.3. **Hunting phase.** After the iterative process of two phases, there is the hunting phase for reptiles. The hunting coordination strategy is performed in the iteration range of $g > \frac{2G}{4}$ and $g \leq \frac{3G}{4}$, while hunting cooperation is performed in the range of $g > \frac{3G}{4}$ and $g \leq G$. The equations are given in (9) and (10).

$$X_{(i,j)}(g+1) = X_j^{Best}(g) \times R_{(i,j)}(g) \times rand, \quad \text{if } g > \frac{2G}{4} \ and \ g \leq \frac{3G}{4} \tag{9}$$

$$X_{(i,j)}(g+1) = X_j^{Best}(g) - \gamma_{(i,j)}(g) \times \epsilon - P_{(i,j)}(g) \times rand, \quad \text{if } g > \frac{3G}{4} \ and \ g \leq G \tag{10}$$

4. **Improved Reptile Search Algorithm(IRSA).** The RSA demonstrates simplicity and effectiveness in addressing straightforward problems. But when confronted with complicated problems, the RSA has limited global optimization capabilities and decreases population variety in search space. During the course of evolution, reptiles tend to move towards the best positions within a population, leading to the consequent premature convergence of evolutionary processes. Therefore, in order to overcome these problems, RSA is improved in this paper, and the pseudocode for Algorithm 2 is displayed.

4.1. **Improved reptile location update mechanism.** In the RSA belly walking equation, the reptile in question exhibits poor search efficiency, mostly due to an excessively large search step size during the initial iterations. Consequently, this results in diminished search accuracy and poses challenges in locating a position superior to the global optimum. The previous optimal position $X_i^{pbest}(g)$ and the current position $X_i(g)$ of the reptile are regarded as the reptile's own crawling experience, that is, its own cognitive item. The optimal position $X^{Best}(g)$ updated by all reptiles in the whole group so far is regarded as the reptile's peer's experience, that is, the group cognitive item. The randomly selected reptile-individual is improved so that its group cognition is controlled by the evolution operator, which is closer to the global optimum and searches more purposefully and in a direction in the space around the global optimum individual, as shown in Equation (11).

$$X_i^{new}(g) = X^{Best}(g) + (X^{Best}(g) - X_{r_1}(g)) \times ES(g), \quad \text{if } g > \frac{G}{4} and\ g \leq \frac{2G}{4} \tag{11}$$

Where $X_i^{new}(g)$ is the new position of the reptile after the $i$th individual moves, $X^{Best}(g)$ is the global optimal position of all reptiles, and $X_{r_1}(g)$ is the position of a random reptile. $ES$ in RSA is a normal distributed random number with respect to the maximum iteration number $g$, ranging from $-2$ to 2. It controls the evolution of a random reptile individual when the reptile is walking and adds the current iteration number $g$. The nonlinear dynamic trend of the reptile evolution operator $ES$ is gradually reduced so that it can perform a fine search near the current global optimal solution. Not only avoid the evolution operator because it is too large and skips the optimal solution, but also increase algorithm convergence precision. $ES$ is determined using Equation (12).

$$ES(g) = 2 \times rand \times (1 - \frac{g}{G}) \tag{12}$$

In the original algorithm, $\beta$ is a constant used to control the exploration ability when the reptile is high. $\alpha$ is a constant to control the exploration accuracy, which is used when reptiles are walking high, hunting coordination, and hunting cooperation. The values of $\alpha$ and $\beta$ are optimized using a linear decreasing strategy in order to enhance the algorithm's global search capability during the initial stages of development and to improve its local exploration capacity in later stages. as shown in Equations (13) and (14).

$$\alpha(g) = \alpha_{max} - (g-1) \times \frac{\alpha_{max} - \alpha_{min}}{G-1} \tag{13}$$

$$\beta(g) = \beta_{max} - (g-1) \times \frac{\beta_{max} - \beta_{min}}{G-1} \tag{14}$$

In the equation, $\alpha_{max}$ is the maximum value of exploration accuracy; after many experiments, the effect is best when it is 0.1, and $\alpha_{min}$ is the minimum value of exploration accuracy, which is 0.01. Equation (13) indicates that $\alpha$ decreases from 0.1 to 0.01, similarly, Equation (14) indicates that $\beta$ decreases from 0.005 to 0.001. After the RSA location update, add a strategy to improve the equilibrium between local and global search and preserve diversity by learning the most optimal position in the evolutionary history of each reptile. As shown in Equation (15).

$$X_i^{new}(g) = X_i^{new}(g) + (X_i^{pbest}(g) - X^{Best}(g)) \times rand \tag{15}$$

Where $X_i^{pbest}(g)$ is the historical optimal location of the individual. Then, the inclusion of the greedy selection strategy involves the comparison of the optimal value between the newly acquired position and the initial reptile's position, which promotes reptile population and convergence speed, as shown in Equation (16).

$$X_i(g) = \begin{cases} X_i^{new}(g), & \text{if } \ f_{X_i}^{new} < f_{X_i} \\ X_i(g), & \text{else} \end{cases} \tag{16}$$

In summary, the improved algorithm retains the global optimal position as the guide when the reptile moves in the original algorithm, effectively screens out the individuals with higher quality to be developed, and makes them move gradually to the optimal position with a more flexible step size. The improved algorithm controls the individual learning information carried on the optimal solution with iteration time by $\alpha$ and $\beta$, so as to ensure the direction of individual progress. The information exchange between populations is strengthened, the influence of original random factors on the algorithm is reduced, the pace of convergence is increasing, leading to enhanced algorithmic accuracy.

4.2. **Hybrid mutation.** The reptile movement process is carried out around the global optimal $X^{Best}(t)$, at this time, the algorithm after the reptile position update mechanism is still prone to local optimization, resulting in the stagnation of the update iteration of the global optimal position of the algorithm. To tackle this issue, the diversity of the reptile population must be increased, and this paper adds a mutation process to the reptile population as a means to enhance diversity and mitigate the aforementioned issue. The roulette wheel selection mechanism employs a calculation to determine the probability of each individual being chosen, taking into account the fitness of the updated position [7], as shown in Equation (17).

$$p_{(X_i)}(g) = \frac{f(X_i(g))}{\sum_{i=1}^{n} f(X_i(g))} \tag{17}$$

According to the fitness value, a new population $X^L(g)$ composed of $M$ worse individuals in the population is selected, and through the test, it is better when M is set to 30. The mutation operation is performed on the individuals in $X^L(g)$. This paper propose a new hybrid mutation equation, as shown in Equation (18).

$$X_i(g+1) = \begin{cases} X^{Best}(g) + X_i^{StepB}(g), & \text{if } \ f_{X_i^L} > f_{min} \\ X_i(g) + X_i^{StepB}(g), & \text{else} \end{cases} \tag{18}$$

In instances where the reptile achieves a more advantageous position compared to the prevailing global best position, it will climb to a position near itself. The precise distance is contingent upon the ratio between the disparity in position between the worst position and its own position and the disparity in quality between the current food and the worst food. Additionally, it is influenced by the ratio between the disparity in position between the optimal position and its own position and the disparity in quality between the current food and the optimal food, in Equation (20). If the reptile is not in the optimal position, it will crawl near the current optimal position to search. In Equation (18), $X_i^{StepB}(g)$ is the step size of $X_i^{Step}(g)$ after boundary processing, and the mutated individual step size obtained from Equation (20) is guaranteed to be within the upper and lower boundaries, as shown in Equation (19).

$$X_i^{StepB}(g) = \begin{cases} X^{UB}, & \text{if } X_i^{Step} > X^{UB} \\ X^{LB}, & \text{if } X_i^{Step} < X^{LB} \end{cases} \tag{19}$$

$$X_i^{Step}(g) = \begin{cases} rand\times(X_i^L(g)-X_i^{Best}(g))\times C_1+rand\times(X_i^L(g)-X_i^{pbest}(g))\times C_2, & \text{if } f_{X_i^L} > f_{min} \\ \dfrac{rand\times|X_i^L(g)-X^{Worse}(g)|}{f_i(g)-f_{max}(g)+\epsilon}\times C_3+\dfrac{rand\times|X_i^L(g)-X^{Best}(g)|}{f_i(g)-f_{min}(g)+\epsilon}\times C_4, & \text{else} \end{cases} \tag{20}$$

Where $X_i^{Step}(g)$ is the step size of the $i$th mutated reptile in the mutant population, $X_i^L(g)$ is the original position of the $i$th reptile in the mutant population, $f_{max}(g)$ is the fitness value of the worst position $X^{Worse}(g)$ of the reptile, and $f_{min}(g)$ is the fitness value of the reptile $X^{Best}(g)$ in the optimal position. When the fitness value obtained by the $i$th reptile is poor, the reptile decides the step size of mutation based on its own individual experience. $X_i^{pbest}(g)$ and the group optimal experience $X^{Best}(g)$. The larger $C_1$ is, the greater the degree of learning from the group optimal position; on the other hand, the greater the degree of learning from the individual historical optimal experience. But if the fitness value attained by the $i$th reptile is superior, it will abandon its current location and proceed to explore alternative positions. The worst position $X^{Worse}(g)$ searched and updated by all reptiles in the whole group is also a group cognitive item. At this time, the two group experiences updated by the algorithm are used to help the reptile climb to other positions. By varying the value of the weight parameter $C$, the reptile is capable of attaining various places. In a similar vein, when the value of $C_3$ is increased, it facilitates a greater degree of learning from the poorest performing individual within the group, hence enhancing the overall diversity of the population. Based on extensive experimentation, setting $C_1$=$C_2$=1, $C_3$=0.6, and $C_4$=0.4 works best.

4.3. **Failure restart.** To enhance the algorithm's capacity to escape local convergence and expand the search space for the best site, a failure restart mechanism is incorporated. In every iteration of the global optimization process, if the updated optimal position and solution are not achieved, the count of failures is incremented by one, denoted as $fail = fail + 1$. When a position better than the global optimal position has not been found after S consecutive failures, the restart equation is executed, as shown in Equation (21).

$$X_i(g) = X^{Best}(g) + rand \times \frac{X^{UB} - X^{LB}}{2} \tag{21}$$

The oscillation search is conducted by utilizing the global optimal position $X^{Best}(g)$ as the central point for updating the positions of the entire reptile population, hence enhancing the population's diversity. After testing, it has been determined that the experimental effect is most optimal when the value of $S$ is increased by a factor of 150.

5. **Experimental Results and Analysis.** To comprehensively evaluate the efficacy of IRSA by conducting a comparative analysis of its convergence accuracy and speed. The evaluation is performed by comparing IRSA with the RSA as well as four other improved prominent swarm intelligence algorithms known for their superior performance on the CEC2013 test set [40], including the Arithmetic-Trigonometric Optimization Algorithm (ATOA) [41], the Chaotic Lichtenberg Algorithm (CLA) [42], the Improved Dung Beetle Optimizer (IDBO) [43] algorithm, and the improved Crow Search Algorithm (ICSA) [44]. The population number of all algorithms is $N$=50, and the maximum function evaluation number is $MaxFEs$=150000 to assure fairness of comparison. Table 1 shows all the setting parameters of the algorithms. All algorithms for Windows 10 of the operating

system, the CPU for i7-8750H computer operation, and MATLAB R2021a programming.

Table 1. Parameter Settings related to each algorithm.

| Algorithm | *Parameter* |
|---|---|
| IRSA | $\alpha_{max}=0.1$; $\alpha_{min}=0.01$; $\beta_{max}=0.005$; $\beta_{min}=0.001$; $C_1=C_2=1$; $C_3=0.6$; $C_4=0.4$ |
| RSA | $\alpha=0.1$; $\beta=0.005$ |
| ATOA | $MOP_{max}=1$; $MOP_{min}=0.2$; $\alpha=5$; $\mu=0.499$ |
| CLA | $c=0.6$ |
| IDBO | $P_percen=0.1$ |
| ICSA | $AP=0.3$; $FL=1.2$; $\beta_{max}=0.9$; $\beta_{min}=0.4$ |

Table 2. Results of IRSA and other comparison algorithms on the 30-dimensional CEC2013 test set

| | ATOA | | CLA | | IDBO | | ICSA | | RSA | | IRSA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| F1 | 1.79E+04 | 4.09E+03 | 2.23E+00 | 6.50E-01 | 3.31E+01 | 3.22E+01 | 4.46E+03 | 9.88E+02 | 4.95E+04 | 8.66E+03 | **0.00E+00** | **0.00E+00** |
| F2 | 9.16E+07 | 4.50E+07 | 2.73E+07 | 6.81E+06 | 7.25E+07 | 3.16E+07 | 8.24E+07 | 1.93E+07 | 7.17E+08 | 4.34E+08 | **4.07E+06** | **2.18E+06** |
| F3 | 3.63E+09 | 1.41E+10 | **0.00E+00** | **0.00E+00** | 2.62E+07 | 1.03E+08 | 4.84E+05 | 2.65E+06 | 1.14E+18 | 4.31E+18 | **0.00E+00** | **0.00E+00** |
| F4 | 5.43E+04 | 7.96E+03 | 3.22E+04 | 8.88E+03 | 5.54E+04 | 8.20E+03 | 4.91E+04 | 6.15E+03 | 6.46E+04 | 3.45E+03 | **1.69E+04** | **4.03E+03** |
| F5 | 1.88E+03 | 4.93E+02 | 3.81E+01 | 1.86E+01 | 1.82E+02 | 5.43E+01 | 3.09E+03 | 8.13E+02 | 9.66E+04 | 2.70E+04 | **0.00E+00** | **0.00E+00** |
| F6 | 8.05E+02 | 3.94E+02 | 9.10E+01 | 4.14E+01 | **1.09E+02** | **3.96E+01** | 3.62E+02 | 6.98E+01 | 8.46E+03 | 3.75E+03 | 1.41E+02 | 3.26E+02 |
| F7 | 3.37E+02 | 8.11E+02 | 8.79E+01 | 3.88E+02 | 9.42E+00 | 4.25E+01 | 1.27E-01 | 6.94E-01 | 2.76E+05 | 5.42E+05 | **0.00E+00** | **0.00E+00** |
| F8 | **2.10E+01** | **5.09E-02** | **2.10E+01** | **5.59E-02** | **2.10E+01** | **5.22E-02** | **2.10E+01** | **4.36E-02** | **2.10E+01** | **5.81E-02** | 2.13E+01 | 8.11E-02 |
| F9 | 1.90E+01 | 1.24E+01 | 2.11E+01 | 1.48E+01 | 6.36E+00 | 1.12E+01 | 9.33E-01 | 5.11E+00 | 3.42E+01 | 4.04E+00 | **0.00E+00** | **0.00E+00** |
| F10 | 1.30E+03 | 4.13E+02 | 3.95E+01 | 1.56E+01 | 2.69E+02 | 1.20E+02 | 7.50E+02 | 1.79E+02 | 6.24E+03 | 1.81E+03 | **1.41E+00** | **4.36E+00** |
| F11 | 9.07E+01 | 6.33E+01 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** | 1.63E+02 | 7.92E+01 | **0.00E+00** | **0.00E+00** |
| F12 | 1.96E+02 | 6.06E+01 | 1.86E+02 | 6.59E+01 | 9.45E+01 | 8.04E+01 | 1.28E+02 | 7.71E+01 | 3.44E+02 | 9.36E+01 | **0.00E+00** | **0.00E+00** |
| F13 | 1.97E+02 | 5.77E+01 | 1.88E+02 | 4.80E+01 | 1.00E+02 | 8.57E+01 | 1.25E+02 | 9.03E+01 | 3.37E+02 | 8.19E+01 | **0.00E+00** | **0.00E+00** |
| F14 | 3.41E+03 | 4.66E+02 | 4.34E+03 | 6.64E+02 | 4.55E+03 | 9.46E+02 | 6.89E+03 | 4.58E+02 | 7.62E+03 | 2.96E+02 | **2.71E+03** | **6.03E+02** |
| F15 | 7.37E+03 | 3.64E+02 | **5.41E+03** | **8.62E+02** | 6.83E+03 | 9.35E+02 | 6.37E+03 | 4.42E+02 | 7.39E+03 | 2.94E+02 | 6.07E+03 | 1.47E+03 |
| F16 | 2.50E+00 | 3.88E-01 | **1.73E+00** | **3.87E-01** | 2.08E+00 | 6.85E-01 | 2.60E+00 | 2.94E-01 | 2.62E+00 | 3.58E-01 | 5.98E+00 | 1.17E+00 |
| F17 | 1.06E+03 | 2.60E+02 | 4.73E+02 | 9.98E+01 | **2.62E+02** | **5.83E+01** | 4.56E+02 | 4.89E+01 | 2.31E+03 | 3.03E+02 | 3.91E+02 | 1.66E+02 |
| F18 | 1.11E+03 | 2.55E+02 | 4.61E+02 | 1.12E+02 | **3.18E+02** | **3.58E+01** | 4.49E+02 | 4.19E+01 | 2.10E+03 | 3.37E+02 | 5.97E+02 | 4.09E+02 |
| F19 | 6.95E+03 | 5.01E+03 | **3.61E+01** | **9.91E+00** | 4.25E+01 | 9.13E+01 | 2.47E+02 | 1.67E+02 | 9.96E+05 | 1.17E+06 | 7.05E+01 | 2.06E+02 |
| F20 | 1.49E+01 | 1.76E-01 | 1.46E+01 | 2.02E-01 | 9.91E+00 | 4.49E+00 | 9.95E+00 | 3.90E+00 | 1.49E+01 | 1.86E-01 | **0.00E+00** | **0.00E+00** |
| F21 | 9.13E+02 | 1.61E+02 | 4.34E+02 | 1.47E+02 | 5.73E+02 | 4.38E+02 | 7.67E+02 | 7.09E+01 | 2.06E+03 | 1.91E+02 | **4.24E+02** | **1.11E+02** |
| F22 | 4.28E+03 | 6.38E+02 | 5.74E+03 | 1.26E+03 | 4.62E+03 | 9.14E+02 | 7.57E+03 | 3.14E+02 | 8.78E+03 | 5.04E+02 | **3.85E+03** | **7.57E+02** |
| F23 | 7.91E+03 | 3.79E+02 | 6.30E+03 | 1.01E+03 | 6.49E+03 | 1.03E+03 | 7.62E+03 | 3.68E+02 | 8.43E+03 | 4.73E+02 | **5.95E+03** | **1.61E+03** |
| F24 | 2.77E+02 | 4.52E+01 | 2.50E+02 | 4.36E+01 | 2.34E+02 | 3.42E+01 | **2.08E+02** | **1.57E+01** | 4.00E+02 | 1.09E+02 | 2.19E+02 | 2.74E+01 |
| F25 | 3.23E+02 | 3.12E+01 | 3.27E+02 | 2.04E+01 | 2.99E+02 | 2.00E+01 | 3.06E+02 | 1.81E+01 | 3.39E+02 | 3.01E+01 | **2.64E+02** | **3.30E+01** |
| F26 | 3.61E+02 | 5.58E+01 | 3.65E+02 | 4.63E+01 | 3.08E+02 | 8.97E+01 | **2.31E+02** | **3.90E+01** | 3.81E+02 | 3.26E+01 | 3.03E+02 | 4.65E+01 |
| F27 | 2.53E+03 | 1.49E+02 | 2.36E+03 | 2.93E+02 | 2.27E+03 | 2.88E+02 | 1.93E+03 | 2.24E+02 | 2.95E+03 | 4.78E+02 | **9.24E+02** | **2.88E+02** |
| F28 | 3.89E+03 | 5.08E+02 | 1.98E+03 | 1.11E+03 | **1.17E+03** | **3.65E+02** | 2.22E+03 | 2.65E+02 | 5.71E+03 | 1.28E+03 | 1.28E+03 | 3.48E+02 |

Table 2 and Table 3 present the average and standard deviation of 30 separate trials conducted on each algorithm using the CEC2013 test set in dimensions of 30 and 50, respectively. Additionally, the algorithm that yielded the most optimal results for the given function is highlighted in black. The results of the Friedman test and the Wilcoxon rank sum test for each algorithm compared to the IRSA, with a significance threshold of 5%, are presented in Table 4.

As shown from Table 2, the 30-dimensional optimization problem has been subjected to IRSA on nine test functions, resulting in the attainment of global optimal values for F1, F3, F5, F7, F9, F11, F12, F13, and F20. RSA did not attain the globally optimal value for any of the test functions. RSA is not on any one test function that has reached the global optimal value, and only in F8 and F16 are the results of RSA better than those of IRSA, but in the rest of the 26 test functions, IRSA has achieved a better average. ATOA failed to achieve the theoretical ideal value for any given test function. CLA obtained theoretically optimal results only on F3 and F11. Both IDBO and ICSA achieve the theoretical

Table 3. Results of IRSA and other comparison algorithms on the 50-dimensional CEC2013 test set

| | ATOA | | CLA | | IDBO | | ICSA | | RSA | | IRSA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| F1 | 3.60E+04 | 6.79E+03 | 1.79E+01 | 3.96E+00 | 1.33E+03 | 1.06E+03 | 1.67E+04 | 2.46E+03 | 7.23E+04 | 5.53E+03 | **4.33E-24** | **1.25E-23** |
| F2 | 2.63E+08 | 7.83E+07 | 7.52E+07 | 2.34E+07 | 2.13E+08 | 1.15E+08 | 2.37E+08 | 5.31E+07 | 1.46E+09 | 9.05E+08 | **1.52E+07** | **1.44E+07** |
| F3 | 2.96E+09 | 6.04E+09 | **0.00E+00** | **0.00E+00** | 6.34E+08 | 3.47E+09 | 1.85E+07 | 5.90E+07 | 3.37E+14 | 5.05E+14 | **0.00E+00** | **0.00E+00** |
| F4 | 9.92E+04 | 1.91E+04 | 5.79E+04 | 1.35E+04 | 8.78E+04 | 6.52E+03 | 7.69E+04 | 3.98E+03 | 8.67E+04 | 4.83E+03 | **4.07E+04** | **5.86E+03** |
| F5 | 3.58E+03 | 6.97E+02 | 2.11E+02 | 6.20E+01 | 6.91E+02 | 1.95E+02 | 6.87E+03 | 1.17E+03 | 4.94E+04 | 1.19E+04 | **3.98E-16** | **1.40E-15** |
| F6 | 1.89E+03 | 6.05E+02 | 1.75E+02 | 6.12E+01 | 7.39E+02 | 5.82E+02 | 1.07E+03 | 1.92E+02 | 7.51E+03 | 1.87E+03 | **7.11E+01** | **4.95E+01** |
| F7 | 5.56E+01 | 1.07E+02 | 2.10E+02 | 4.07E+02 | 2.86E+00 | 1.12E+01 | 3.25E+00 | 5.06E+00 | 7.93E+03 | 1.07E+04 | **0.00E+00** | **0.00E+00** |
| F8 | **2.12E+01** | **4.33E-02** | **2.12E+01** | **3.45E-02** | **2.12E+01** | **3.92E-02** | **2.12E+01** | **3.60E-02** | **2.12E+01** | **4.61E-02** | 2.14E+01 | 6.01E-02 |
| F9 | 2.27E+01 | 2.31E+01 | 2.46E+01 | 2.57E+01 | 9.77E-01 | 5.35E+00 | **0.00E+00** | **0.00E+00** | 4.48E+01 | 7.02E+00 | **0.00E+00** | **0.00E+00** |
| F10 | 3.39E+03 | 7.84E+02 | 2.01E+02 | 5.12E+01 | 3.26E+03 | 1.21E+03 | 2.26E+03 | 3.18E+02 | 1.04E+04 | 2.30E+03 | **3.03E+01** | **9.86E+01** |
| F11 | 1.82E+02 | 8.31E+01 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** | 4.66E+00 | 9.02E+00 | 3.50E+02 | 1.15E+02 | **0.00E+00** | **0.00E+00** |
| F12 | 3.70E+02 | 1.04E+02 | 4.00E+02 | 8.06E+01 | 8.70E+01 | 1.36E+02 | 3.65E+02 | 3.67E+01 | 4.70E+02 | 4.13E+01 | **0.00E+00** | **0.00E+00** |
| F13 | 3.80E+02 | 7.84E+01 | 3.90E+02 | 6.12E+01 | 5.86E+01 | 1.29E+02 | 3.12E+02 | 1.26E+02 | 4.75E+02 | 4.14E+01 | **0.00E+00** | **0.00E+00** |
| F14 | 8.52E+03 | 7.86E+02 | **1.77E+03** | **3.61E+03** | 8.83E+03 | 1.54E+03 | 1.32E+04 | 5.66E+02 | 1.41E+04 | 5.91E+02 | 3.90E+03 | 7.32E+02 |
| F15 | 1.44E+04 | 4.42E+02 | **1.03E+04** | **1.27E+03** | 1.39E+04 | 1.31E+03 | 1.38E+04 | 5.26E+02 | 1.48E+04 | 2.82E+02 | 1.13E+04 | 2.07E+03 |
| F16 | 3.56E+00 | 3.49E-01 | **2.47E+00** | **3.61E-01** | 3.07E+00 | 6.87E-01 | 3.72E+00 | 3.19E-01 | 3.72E+00 | 2.80E-01 | 7.09E+00 | 9.90E-01 |
| F17 | 1.87E+03 | 2.78E+02 | 1.12E+03 | 2.15E+02 | **7.96E+02** | **4.10E+02** | 1.17E+03 | 1.18E+02 | 3.33E+03 | 2.25E+02 | 9.69E+02 | 2.90E+02 |
| F18 | 2.15E+03 | 3.22E+02 | 1.12E+03 | 2.45E+02 | **8.08E+02** | **3.73E+02** | 1.15E+03 | 1.09E+02 | 3.32E+03 | 2.44E+02 | 1.01E+03 | 2.30E+02 |
| F19 | 1.73E+04 | 1.14E+04 | **1.05E+02** | **3.21E+01** | 6.85E+03 | 1.02E+04 | 4.08E+03 | 2.73E+03 | 4.69E+05 | 1.88E+05 | 4.28E+02 | 8.98E+02 |
| F20 | 2.47E+01 | 5.74E-01 | 2.46E+01 | 2.09E-01 | 1.77E+01 | 7.56E+00 | 2.35E+01 | 1.08E+00 | 2.49E+01 | 1.45E-01 | **8.16E+00** | **1.17E+01** |
| F21 | 2.46E+03 | 2.48E+02 | **4.02E+02** | **5.66E+00** | 5.21E+02 | 4.78E+02 | 1.75E+03 | 1.29E+02 | 3.30E+03 | 1.38E+02 | 7.20E+02 | 1.39E+03 |
| F22 | 1.02E+04 | 9.44E+02 | 1.11E+04 | 2.18E+03 | 9.95E+03 | 1.02E+03 | 1.45E+04 | 4.29E+02 | 1.58E+04 | 6.82E+02 | **8.42E+03** | **1.67E+03** |
| F23 | 1.51E+04 | 6.32E+02 | **1.23E+04** | **1.38E+03** | 1.35E+04 | 1.79E+03 | 1.47E+04 | 5.25E+02 | 1.59E+04 | 3.60E+02 | 1.31E+04 | 2.28E+03 |
| F24 | 3.84E+02 | 3.85E+01 | 3.89E+02 | 7.19E+01 | 3.31E+02 | 4.02E+01 | 2.55E+02 | 3.21E+01 | 6.74E+02 | 2.59E+02 | **2.41E+02** | **4.17E+01** |
| F25 | 4.07E+02 | 1.39E+01 | 4.14E+02 | 1.13E+01 | 3.70E+02 | 1.41E+01 | 3.84E+02 | 2.31E+01 | 4.78E+02 | 5.65E+01 | **3.07E+02** | **6.41E+01** |
| F26 | 4.50E+02 | 7.15E+01 | 4.51E+02 | 7.05E+01 | 4.25E+02 | 4.72E+01 | 3.81E+02 | 3.88E+01 | 4.89E+02 | 1.32E+01 | **3.26E+02** | **5.23E+01** |
| F27 | 3.47E+03 | 2.14E+02 | 3.30E+03 | 2.81E+02 | 3.13E+03 | 2.29E+02 | 2.58E+03 | 3.09E+02 | 4.73E+03 | 1.01E+03 | **1.67E+03** | **2.77E+02** |
| F28 | 7.05E+03 | 9.61E+02 | 7.43E+03 | 1.72E+03 | **4.61E+03** | **1.40E+03** | 5.42E+03 | 3.58E+02 | 8.32E+03 | 5.85E+02 | 4.65E+03 | 7.24E+02 |

optimum only when applied to the F11 function. As Table 4 shows, compared with IRSA, RSA only performed significantly better on F8 and F16, but performance on the other 26 test functions is significantly worse. ATOA performed better on the F8 and F16 test functions but worse on the other 26 functions. CLA exhibits comparable performance to IRSA across seven test functions and demonstrates notably superior performance on the F8, F15, and F16 test functions, but performs poorly on the remaining 18 test functions. IDBO is comparable to IRSA on 9 test functions, and IDBO on F8, F16, F17, and F18 test functions exhibits much superior performance, but the IDBO demonstrates significantly poorer performance on the other 15 test functions. ICSA is comparable to IRSA on 5 test functions, exhibiting similar levels of effectiveness and demonstrating notably superior performance on four test functions, surpassing the performance of other algorithms, but ICSA performs weakly on the remaining nineteen test functions. In conclusion, with RSA and the other four improved optimization techniques, IRSA shows clear convergence accuracy gains.

The dimensions of each algorithm in the CEC2013 random run on test set are shown in Figure 1. These are the test function curve of convergence, the abscissa number for computing function evaluation, and the y coordinate for the number of computational evaluations under the fitness value of logarithmic. This makes it easier to compare the speeds at which different algorithms converge.

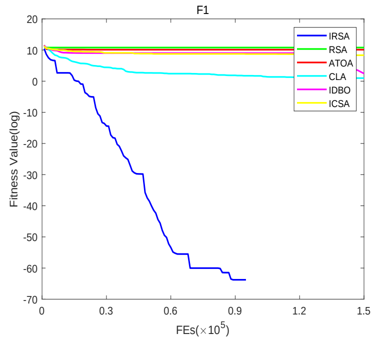
The analysis of Figure 1 reveals that IRSA demonstrates superior performance in achieving the global optimal value for single modular functions F1–F5. It exhibits optimal convergence accuracy and convergence speed when compared to an alternative algorithm. When employed in F2 and F4, IRSA offers the highest convergence accuracy and speed of the five algorithms. IRSA has been shown to possess the capability to acquire global optimal solutions for multimodal functions F6–F20. The algorithm attains the theoretical maximum performance on the test functions F7, F9, F11, F12, F13, and F20. The findings derived from the IRSA exhibit greater levels of convergence accuracy and the

TABLE 4. Wilcoxon rank sum test and Friedman test between IRSA and other algorithms

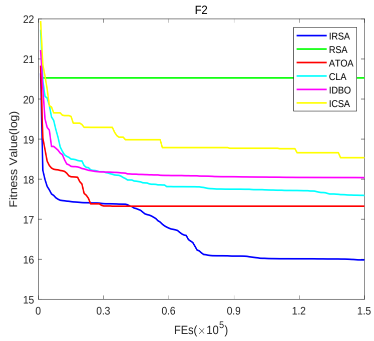
| | D=30 | | | | | D=50 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ATOA | CLA | IDBO | ICSA | RSA | ATOA | CLA | IDBO | ICSA | RSA |
| F1 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F2 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F3 | 0.000(−) | 1.000(=) | 0.082(=) | 0.334(=) | 0.000(−) | 0.000(−) | 1.000(=) | 0.042(−) | 0.001(−) | 0.000(−) |
| F4 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F5 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F6 | 0.000(−) | 0.096(=) | 0.005(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F7 | 0.000(−) | 0.006(−) | 0.042(−) | 0.334(=) | 0.000(−) | 0.000(−) | 0.000(−) | 0.082(=) | 0.000(−) | 0.000(−) |
| F8 | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) |
| F9 | 0.000(−) | 0.000(−) | 0.001(−) | 0.334(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.334(=) | 1.000(=) | 0.000(−) |
| F10 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F11 | 0.000(−) | 1.000(=) | 1.000(=) | 1.000(=) | 0.000(−) | 0.000(−) | 1.000(=) | 1.000(=) | 0.000(−) | 0.000(−) |
| F12 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.001(−) | 0.000(−) | 0.000(−) |
| F13 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.003(−) | 0.000(−) | 0.000(−) |
| F14 | 0.001(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F15 | 0.000(−) | 0.019(+) | 0.075(=) | 0.483(=) | 0.000(−) | 0.000(+) | 0.000(+) | 0.145(=) | 0.610(=) | 0.000(+) |
| F16 | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) | 0.000(+) |
| F17 | 0.000(−) | 0.004(−) | 0.001(+) | 0.006(−) | 0.000(−) | 0.000(−) | 0.002(−) | 0.001(+) | 0.000(−) | 0.000(−) |
| F18 | 0.000(−) | 0.340(=) | 0.000(+) | 0.234(=) | 0.000(−) | 0.000(−) | 0.048(−) | 0.000(+) | 0.001(−) | 0.000(−) |
| F19 | 0.000(−) | 0.530(=) | 0.196(=) | 0.000(−) | 0.000(−) | 0.000(−) | 0.001(−) | 0.010(−) | 0.000(−) | 0.000(−) |
| F20 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.007(−) | 0.000(−) | 0.019(-) | 0.000(−) | 0.007(−) |
| F21 | 0.000(−) | 0.031(−) | 0.807(=) | 0.000(−) | 0.000(−) | 0.000(−) | 0.009(+) | 0.620(=) | 0.000(−) | 0.000(−) |
| F22 | 0.015(−) | 0.000(−) | 0.001(−) | 0.000(−) | 0.000(−) | 0.042(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.042(−) |
| F23 | 0.000(−) | 0.559(=) | 0.340(=) | 0.000(−) | 0.000(−) | 0.000(+) | 0.009(+) | 0.464(=) | 0.001(−) | 0.000(+) |
| F24 | 0.000(−) | 0.387(=) | 0.935(=) | 0.000(+) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.099(=) | 0.000(−) |
| F25 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.042(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.042(−) |
| F26 | 0.000(−) | 0.000(−) | 0.119(=) | 0.000(+) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F27 | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) | 0.000(−) |
| F28 | 0.000(−) | 0.003(−) | 0.149(=) | 0.000(−) | 0.000(−) | 0.001(−) | 0.000(−) | 0.589(=) | 0.000(−) | 0.001(−) |
| +/=/− | 2/0/26 | 3/7/18 | 4/9/15 | 4/5/19 | 2/0/26 | 4/0/24 | 5/2/21 | 4/7/17 | 2/3/23 | 4/0/24 |
| **Avg.rank** | 4.48 | 2.89 | 2.70 | 3.23 | 5.84 | 4.43 | 2.95 | 2.71 | 3.39 | 5.77 |
| **sort** | 5 | 3 | 2 | 4 | 6 | 5 | 3 | 2 | 4 | 6 |

highest rate of convergence when compared to the other five algorithms across the F6, F10, F14, F17, F18, and F19 test functions. The results obtained from iterative IRSA indicate a slightly lower level of convergence accuracy compared to other algorithms, such as F8, F15, and F16 test functions, but the convergence time of IRSA remains comparable to these algorithms. While the initial convergence rate of IRSA may be slightly slower compared to other algorithms, it demonstrates a distinct advantage in the latter stages of evolution. Unlike other algorithms, IRSA maintains its search performance and continues to explore and identify superior solutions. The composite functions F21–F28 have been subject to analysis, revealing that the IRSA algorithm demonstrates greater convergence accuracy and the highest convergence speed in comparison, specifically when applied to the F21, F22, and F27 test functions. The results obtained from the IRSA demonstrate superior convergence accuracy and faster convergence speed when applied to problems F23 and F25. According to the findings, the convergence time of the IRSA on F24 is comparable to that of the IDBO and the ICSA. The convergence accuracy of the IRSA is ranked second, behind the ICSA. The convergence speed and accuracy of the IRSA exhibit similarities to those of the ICSA on the F26. The convergence accuracy of the

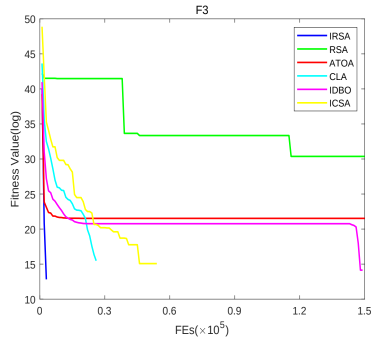
IRSA is ranked second only to that of the ICSA. IRSA demonstrates superior convergence speed on F28, and in terms of convergence accuracy, it ranks second only to IDBO. In summary, when compared to RSA and the other four notable enhanced optimization methods, IRSA has specific advantages in terms of convergence speed.



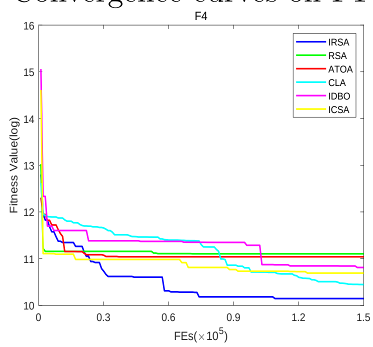

Convergence curves on F1



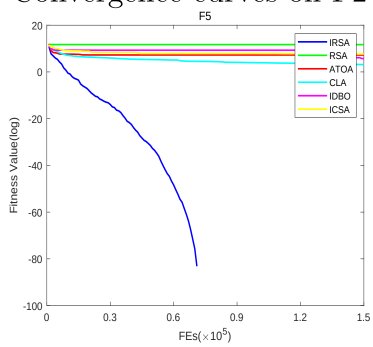

Convergence curves on F2



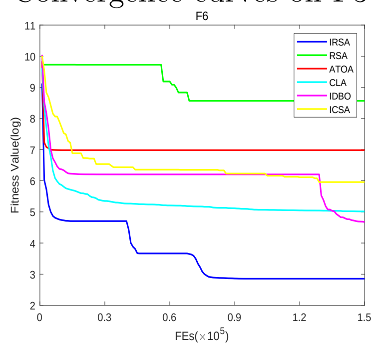

Convergence curves on F3



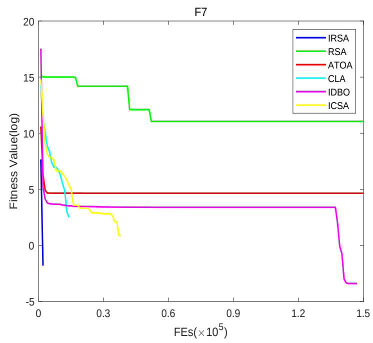

Convergence curves on F4



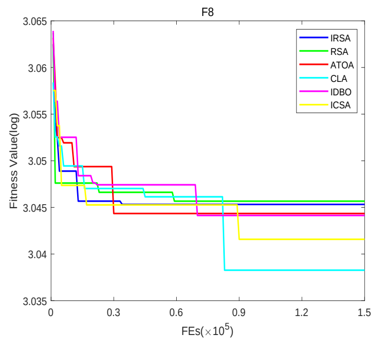

Convergence curves on F5



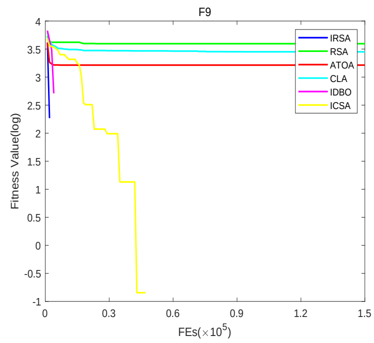

Convergence curves on F6



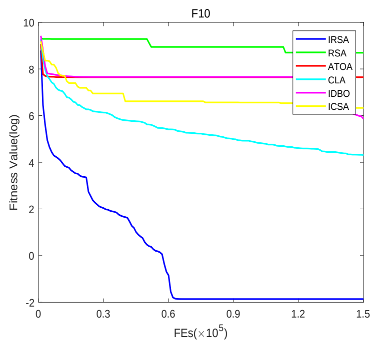

Convergence curves on F7



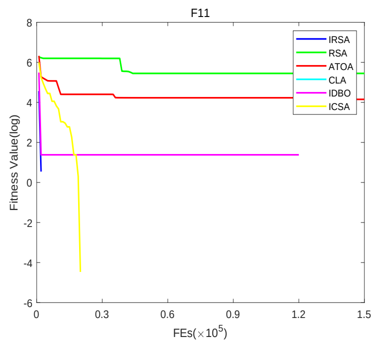

Convergence curves on F8



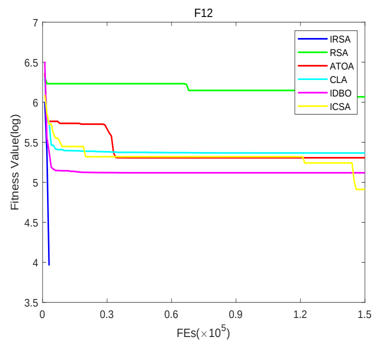

Convergence curves on F9




Convergence curves on F10




Convergence curves on F11

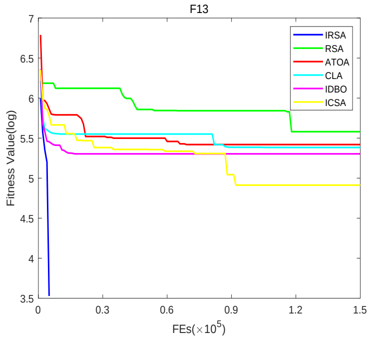



Convergence curves on F12

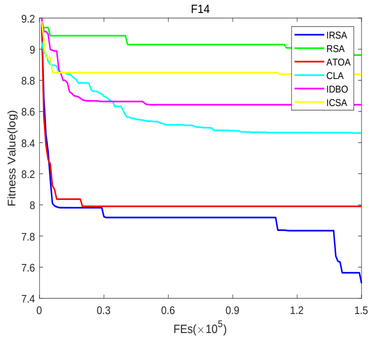
Figure 1. Convergence curve of each function

**Algorithm 2** IRSA

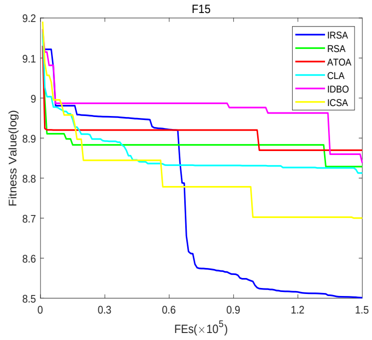
**Input:** $N$, $dim$, $\alpha_{max}$, $\alpha_{min}$, $\beta_{max}$, $\beta_{min}$, $M$, $G$, $C1$, $C2$, $C3$, $C4$, $S$.
**Output:** The optimal solution and its fitness value
Initialize population ($X_1$, $X_2$, ...,$X_i$, ..., $X_N$)
Calculate the fitness value of each individual in the initialized population
Find the global optimum, the global worst, and the individual optimum
**while** $g<G$ **do**
  **if** $fail > S$ **then**
    **for** $i$=1 to $N$ **do**
      Update the position using the shock Equation (21) in section 4.3
    **end for**
    Calculate each individual's fitness value for the population
    Find the global optimum, the global worst, and the individual optimum so far
  **else**
    Using Equation (12) to update $ES$
    **for** $i$=1 to $N$ **do**
      Using Equation (4, 5 and 7) to update parameters $\gamma$, $R$, and $P$.
      **if** $g > \frac{G}{4}$ **then**
        $X_i^{new}(g) = X^{Best}(g) - \gamma_i(g) \times \beta(t) - P_i(g) \times rand$
      **else if** $g > \frac{G}{4}$ and $g < 2\frac{G}{4}$ **then**
        $X_i^{new}(g) = X^{Best}(g) + (X^{Best}(g) - X_{rand}(g)) \times ES(g)$
      **else if** $g > 2\frac{G}{4}$ and $g < 3\frac{G}{4}$ **then**
        $X_i^{new}(g) = X^{Best}(g) \times R_i(g) \times rand$
      **else** $g > 3\frac{G}{4}$ and $g < G$
        $X_i^{new}(g) = X^{Best}(g) - \gamma_i(g) \times \epsilon - P_i(g) \times rand$
      **end if**
      Using Equation (15) each individual learns from its individual optimum
      Calculate each individual's fitness value for the population
      Using Equation (16) to make greedy choices
      Find the global optimum, the global worst, and the individual optimum so far
      **if** $f_{X_i(g)} \geq f_{min}$ **then**
        $fail = fail + 1$
      **end if**
    **end for**
  **end if**
  Use Equation (17) to calculate the probability of selecting individuals in the population
  Select $M$ individuals to form an $X^L$ population
  **for** $i$=1 to $M$ **do**
    Using Equations (18, 19, and 20) for mutation operations
    Calculate each individual's fitness value for the population
    Find the global optimum, the global worst, and the individual optimum so far
    **if** $f_{X_i(g+1)} \geq f_{min}$ **then**
      $fail = fail + 1$
    **end if**
  **end for**
  $g = g + 1$
**end while**
Return the best solution



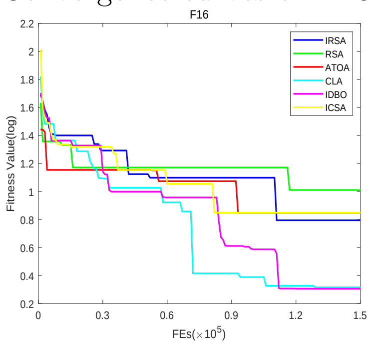

Convergence curves on F13



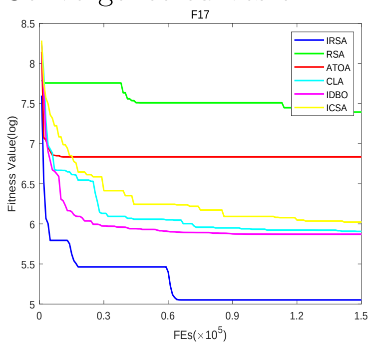

Convergence curves on F14



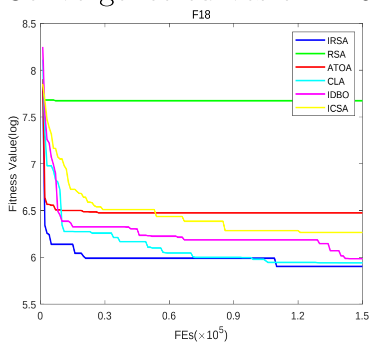

Convergence curves on F15



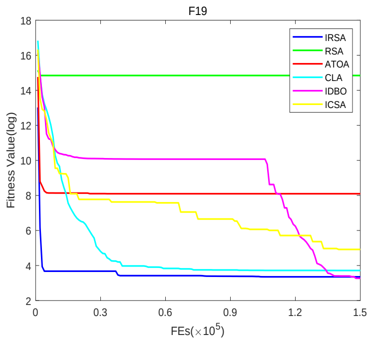

Convergence curves on F16



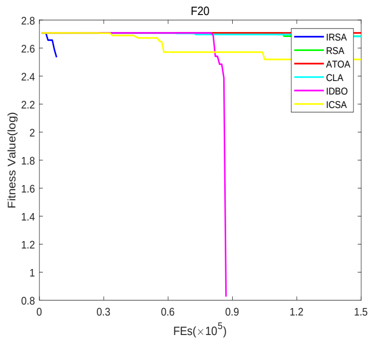

Convergence curves on F17



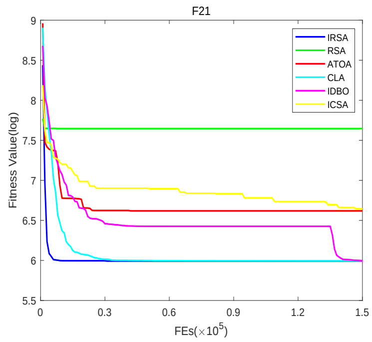

Convergence curves on F18



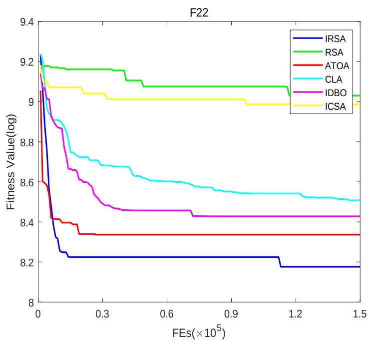

Convergence curves on F19



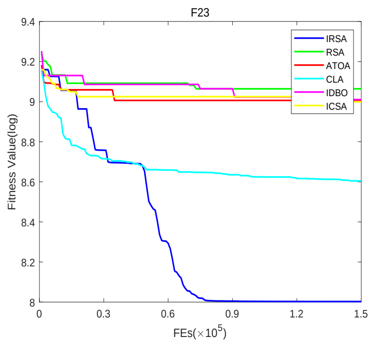

Convergence curves on F20



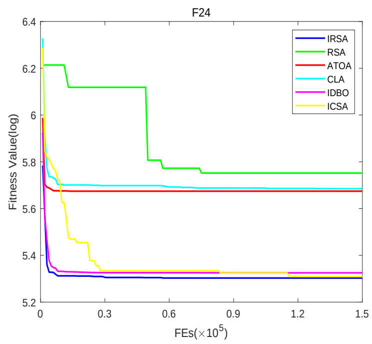

Convergence curves on F21




Convergence curves on F22

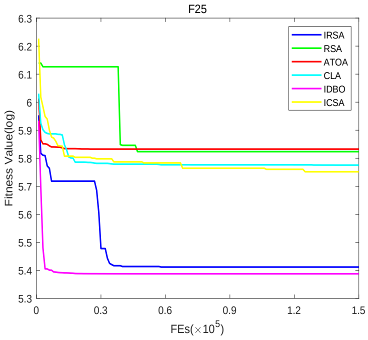



Convergence curves on F23

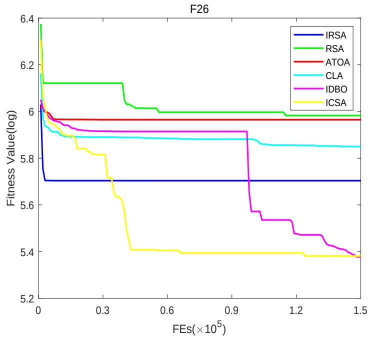



Convergence curves on F24

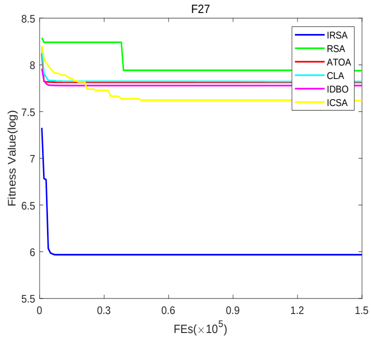
FIGURE 1. Convergence curve of each function



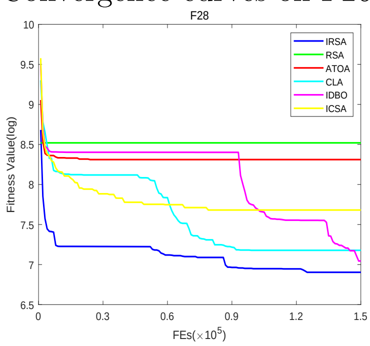

Convergence curves on F25




Convergence curves on F26




Convergence curves on F27




Convergence curves on F28

FIGURE 1. Convergence curve of each function

6. **Conclusion.** To increase the RSA's convergent performance and ability to jump out of the local optimum, this paper proposes an improved reptile search algorithm (IRSA). The update mechanism of reptile belly walking position in the search phase is enhanced, and reptile learning from individual history is introduced to increase crocodile population optimization and early iteration convergence speed. Add mutations and create a new individual position update equation using mixed mutations to improve population diversity and convergence speed. Add a failure restart mechanism to boost population diversity, jump out of local optimal, and search accuracy. The CEC2013 test set showed that the suggested IRSA for unimodal and multi-peak situations is relatively competitive with four different optimization techniques.

## REFERENCES

[1] W.-Y. Sun, and Y.-X. Yuan, "Optimization theory and methods: nonlinear programming," *Springer Science & Business Media*, vol. 1, pp. 1, 2006.
[2] R.-Y. Dong, "Research and Application of Meta-Heuristic Optimization Algorithms," Diss. Jilin University, 2019.
[3] M. Pant, H. Zaheer, and G.-H. Laura, "Differential Evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, pp. 103479, 2020.
[4] W.-A. Guo, M. Chen, L. Wang, Y.-F Mao and Q.-D. Wu, "A survey of biogeography-based optimization," *Neural Computing and Applications*, vol. 28, pp. 1909–1926, 2017.
[5] L. B. Carvalho, and P. P. de Oliveira, "Extending standard evolutionary programming with self-adaptive stable distributions," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 31, no. 2, pp. 114–142, 2016.
[6] S. Nguyen, Y. Mei, and M.-J. Zhang, "Genetic programming for production scheduling: a survey with a unified framework," *Complex & Intelligent Systems*, vol. 3, pp. 41–66, 2017.
[7] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2021.

[8] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of Statistical Physics*, vol. 34, pp. 975–986, 1984.

[9] R. A. Formato, "Central force optimization: A new deterministic gradient-like optimization metaheuristic," *Opsearch*, vol. 46, no. 1, pp. 25–51, 2009.

[10] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.

[11] Y.-J. Zheng, "Water wave optimization: a new nature-inspired metaheuristic," *Computers & Operations Research*, vol. 55, pp. 1–11, 2015.

[12] V. Muthiah-Nakarajan, and M. M. Noel, "Galactic swarm optimization: a new global optimization metaheuristic inspired by galactic motion," *Applied Soft Computing*, vol. 38, pp. 771–787, 2016.

[13] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, pp. 105190, 2020.

[14] E. Atashpaz-Gargari, and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," *2007 IEEE Congress on Evolutionary Computation*, pp. 4661–4667, 2007.

[15] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[16] F. Ramezani, and S. Lotfi, "Social-based algorithm (SBA)," *Applied Soft Computing*, vol. 13, no. 5, pp. 2837–2856, 2013.

[17] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022.

[18] B. C. Mohan, and R. Baskaran, "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618–4627, 2012.

[19] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[20] X.-S. Yang, and X.-S. He, "Bat algorithm: literature review and applications," *International Journal of Bio-inspired Computation*, vol. 5, no. 3, pp. 141–149, 2013.

[21] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm," *Applied Soft Computing*, vol. 61, pp. 1041–1059, 2017.

[22] A. L. Bolaji, M. A. Al-Betar, M. A. K. Awadallah, Tajudin, and L. M. Abualigah, "A comprehensive review: Krill Herd algorithm (KH) and its applications," *Applied Soft Computing*, vol. 49, pp. 437–446, 2016.

[23] V. Kumar, and D. Kumar, "A systematic review on firefly algorithm: past, present, and future," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3269–3291, 2021.

[24] H. Faris, I. Aljarah, M. A. Al-Betar, S.Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, pp. 413–435, 2018.

[25] F. S. Gharehchopogh, and H. Gholizadeh, "A comprehensive survey: Whale Optimization Algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1–24, 2019.

[26] M. Jain, Mohit, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 148–175, 2019.

[27] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, pp. 116158, 2022.

[28] G.-G. Wang, A.H. Gandomi, and A. H. Alavi, "An effective krill herd algorithm with migration operator in biogeography-based optimization," *Applied Mathematical Modelling*, vol. 38, no. 9–10, pp. 2454–2462, 2014.

[29] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Information Sciences*, vol. 300, pp. 140–157, 2015.

[30] G.-G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerospace Science and Technology*, vol. 49, pp. 231–238, 2016.

[31] H. Wang, W.-J. Wang, X.-Y Zhou, H. Sun, J. Zhao, X. Yu, and Z.-H. Cui, "Firefly algorithm with neighborhood attraction," *Information Sciences*, vol. 382, pp. 374–387, 2017.

[32] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowledge-Based Systems*, vol. 195, pp. 105746, 2020.

[33] A. L. H. P. Shaik, M. K. Manoharan, A. K. Pani, R. R. Avala, and C. M. Chen,, "Gaussian Mutation–Spider Monkey Optimization (GM-SMO) Model for Remote Sensing Scene Classification," *Remote Sensing*, vol. 14, no. 24 , pp. 6279, 2022.

[34] H.-L. Zhang, T. Liu, X.-J. Ye, A. A. Heidari, G.-X. Liang, H.-L. Chen, and Z.-F. Pan, "Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems," *Engineering with Computers*, vol. 39, no. 3, pp. 1735–1769, 2023.

[35] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, no. 1, pp. 124, 2023.

[36] R. Almodfer, M. Mudhsh, S. Chelloug, M. Shehab, L. Abualigah, and M. Abd Elaziz, "Quantum mutation reptile search algorithm for global optimization and data clustering," *Human-centric Computing and Information Sciences*, vol. 12, 116158, 2022.

[37] Z. Elgamal, A. Q. Sabri, M. Tubishat, D. Tbaishat, S. N. Makhadmeh, O. A. Alomar, "Improved reptile search optimization algorithm using chaotic map and simulated annealing for feature selection in medical field," *IEEE Access*, vol. 10, pp. 51428–51446, 2022.

[38] Q.-H. Yuan, Y.-D. Zhang, X.-S. Dai, S. Zhang, and others, "A Modified Reptile Search Algorithm for Numerical Optimization Problems," *Computational Intelligence and Neuroscience*, vol. 2022, 9752003, 2022.

[39] L.-Q. Huang, Y.-Y. Wang, Y.-X. Guo, and G. Hu, "An improved reptile search algorithm based on lévy flight and interactive crossover strategy to engineering application," *Mathematics*, vol. 10, no. 13, pp. 2329, 2022.

[40] T. Harris, and J. W. Hardin, "Exact Wilcoxon signed-rank and Wilcoxon Mann–Whitney ranksum tests," *The Stata Journal*, vol. 13, no. 2 , pp. 337–343, 2013.

[41] P. A. M. Devan, F. A. Hussin, R. B. Ibrahim, K. Bingi, M. Nagarajapandian, and M. Assaad, "An arithmetic-trigonometric optimization algorithm with application for control of real-time pressure process plant," *Sensors*, vol. 22, no. 2, pp. 617, 2022.

[42] J. L. J. Pereira, M. B. Francisco, F. A. de Almeida, B. J. Ma, S. S. Cunha Jr, and G. F. Gomes, "Enhanced Lichtenberg algorithm: a discussion on improving meta-heuristics," *Soft Computing*, vol. 27, no. 21, pp. 15619–15647, 2023.

[43] J.-K. Xue, and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *The Journal of Supercomputing*, vol. 79, no. 7, pp. 7305–7336, 2023.

[44] M. A. Hussein, and E. H. Karam, "PI-D controller based on an improved crow search algorithm for cancer growth treatment," *Journal of Engineering and Sustainable Development*, vol. 25, no. 6, pp. 82–90, 2021.