mathematics

MDPI

Article

Enhanced Remora Optimization Algorithm for Solving Constrained Engineering Optimization Problems

Shuang Wang [1,*], Abdelazim G. Hussien [2,3,*], Heming Jia [1], Laith Abualigah [4] and Rong Zheng [1]

[1] School of Information Engineering, Sanming University, Sanming 365004, China; jiaheming@fjsmu.edu.cn (H.J.); zhengr@fjsmu.edu.cn (R.Z.)
[2] Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
[3] Faculty of Science, Fayoum University, Fayoum 63514, Egypt
[4] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan; ligah.2020@gmail.com
* Correspondence: shuang.wang@fjsmu.edu.cn (S.W.); abdelazim.hussien@liu.se (A.G.H.)

Abstract: Remora Optimization Algorithm (ROA) is a recent population-based algorithm that mimics the intelligent traveler behavior of Remora. However, the performance of ROA is barely satisfactory; it may be stuck in local optimal regions or has a slow convergence, especially in high dimensional complicated problems. To overcome these limitations, this paper develops an improved version of ROA called Enhanced ROA (EROA) using three different techniques: adaptive dynamic probability, SFO with Levy flight, and restart strategy. The performance of EROA is tested using two different benchmarks and seven real-world engineering problems. The statistical analysis and experimental results show the efficiency of EROA.

Keywords: remora optimization algorithm; adaptive dynamic probability; restart strategy; metaheuristic algorithm; constrained engineering problems

MSC: 49K35

Citation: Wang, S.; Hussien, A.G.; Jia, H.; Abualigah, L.; Zheng, R. Enhanced Remora Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics* **2022**, *10*, 1696. https://doi.org/10.3390/math10101696

Academic Editors: Shi Qiang Liu, Erhan Kozan, Felix T. S. Chan and Weidong Li

Received: 21 April 2022
Accepted: 11 May 2022
Published: 16 May 2022

1. Introduction

The process of determining the best values of design variables to minimize/maximize fitness function while fulfilling the requirements of the whole system is known as optimization [1,2]. Optimization problems exist in almost everything and every field, such as engineering, business, science, etc. Optimization methods can be classified into two large categories: (1) exact and (2) heuristic & metaheuristic algorithms [3–5]. The former category can be considered less applicable and practical as it needs less complicated calculations, which will take much time. In contrast, the latter class (metaheuristic algorithms) shows some randomized/stochastic behavior and performs an "educated search decision" for some "wise regions" [6,7].

These days optimization field has gained a huge interest by many scholars as it become one of the hot topics in computer science since it appears in many domains such as cloud computing tasks [8], face detection [9], power [10,11], and engineering problems [12]. In the literature, there exist an enormous number of optimization algorithms since there is no algorithm that is able to find the optimal solution in all problems as stated by the No Free Lunch (NFL) theory [13]. In other words, if an algorithm is able to find the optimal solution in one type of problem, it will fail in other types. The above theorem encourages researchers to introduce novel algorithms and enhance already existed ones.

Metaheuristics algorithms can be classified based on their source of inspiration into three sub-classes: (1) Swarm-based, (2) Physics-based, and (3) Evolution-based [14].

Swarm-inspired metaheurstics contain algorithms that mimic species social & biological traits such as mating, labor division, foraging, navigation or self-organize [15,16].

Examples of such algorithms includes Particle Swarm Optimization (PSO) [17], Ant Colony Optimization [18], Grey Wolf Optimization (GWO) [19], Ant Lion Optimizer (ALO) [20], Whale Optimization Algorithm (WOA) [21], Marine Predator Algorithm (MPA) [22], Salp Swarm Algorithm (SSA) [23], Remora Optimization Algorithm (ROA) [24], Harris Hawks Optimization (HHO) [25], COOT bird [26], Moth-Flame Optimization (MFO) [27], Social Spider Optimization [28], Snake Optimizer [29], Crow Search Algorithm [30], Emperor Penguin Optimizer [31], Virus Colony Search [32], Spotted Hyena Optimizer [33], Aquila Optimizer (AO) [34], and Tunicate Swarm Algorithm [35].

Physics-inspired meta-heuristic contains algorithms inspired by physical laws or phenomena. Examples of such algorithms, contain Simulated Annealing [36], Big-Bang Big-Crunch [37], Gravitational Search Algorithm (GSA) [38], Lightning Search Algorithm [39], Black Hole Algorithm [40], Sine Cosine Algorithm (SCA) [41], Ray Optimization Algorithm [42], Artificial Electric Field Algorithm [43], Arithmetic Optimization Algorithm (AOA) [44], Multi-Verse Optimizer (MVO) [45], and Henry Gas Solubility Optimization [46].

Evolution-based algorithms are driven from biological evolution ideas. Examples of such algorithms are Genetic Algorithm (GA) [47], Evolutionary Programming [48], Biogeography Based Optimization [49], Memetic Algorithm [50], Bacterial Foraging Optimization [51], Artificial Algae Algorithm [52], and Monkey King Evolutionary [53].

ROA [24] is a recent metaheuristic population-based algorithm inspired by Remora foraging parasitic behavior in the oceans. ROA handles many updating position rules based on many hosts. Zheng et al. [54] developed an improved version of ROA called IROA by using an autonomous foraging mechanism (AFM). Furthermore, Liu et al. [55] developed a modified version of ROA based on Brownian motion and lens opposition-based. Vinayaki et al. [56] applied ROA in the multilevel image segmentation in the detection of Retinopathy from fundus images. In this study, an enhanced ROA version is proposed, which is called EROA, to improve the performance of the original ROA by using three techniques, (1) Adaptive dynamic probability, (2) Sailfish Optimizer (SFO) with Levy flight, and (3) Restart strategy (RS).

The main contribution of this paper can be summarized as follows:

- Enhanced version of ROA is proposed based on 3 strategies: Adaptive dynamic probability, SFO with Levy flight, and Restart strategy (RS);
- EROA has been tested on 23 different functions (CEC2005), 29 functions from (CEC2017), and 7 real-world engineering problems;
- EROA has been tested using 3 dimensions (D = 30, 100, & 500);
- EROA has been compared with original algorithm and other 6 different algorithms.

This paper is organized as follows: Section 2 gives a brief description of the ROA, whereas Section 3 illustrates the using operators: adaptive dynamic probability, Levy flight, and restart strategy, and gives the framework of the proposed algorithm. Sections 4 and 5 show the experiments results of the proposed algorithm in solving benchmark and constrained engineering problems, whereas Section 6 concludes the paper.

2. Remora Optimization Algorithm (ROA)

ROA is a new metaheuristic optimization algorithm inspired by Remora, the "intelligent traveler" in the ocean (Figure 1). It mimics the concept of parasitism and random host replacement of Remora. Remora can attach to whales and swordfishes to learn the effective characteristics of the hosts, so the ROA borrows two strategies from WOA and SFO [57]. ROA contains "Free travel" and "Eat thoughtfully" phases, corresponding to the exploration and exploitation stages. The algorithm switches between exploration and exploitation phases through a "one small step try".

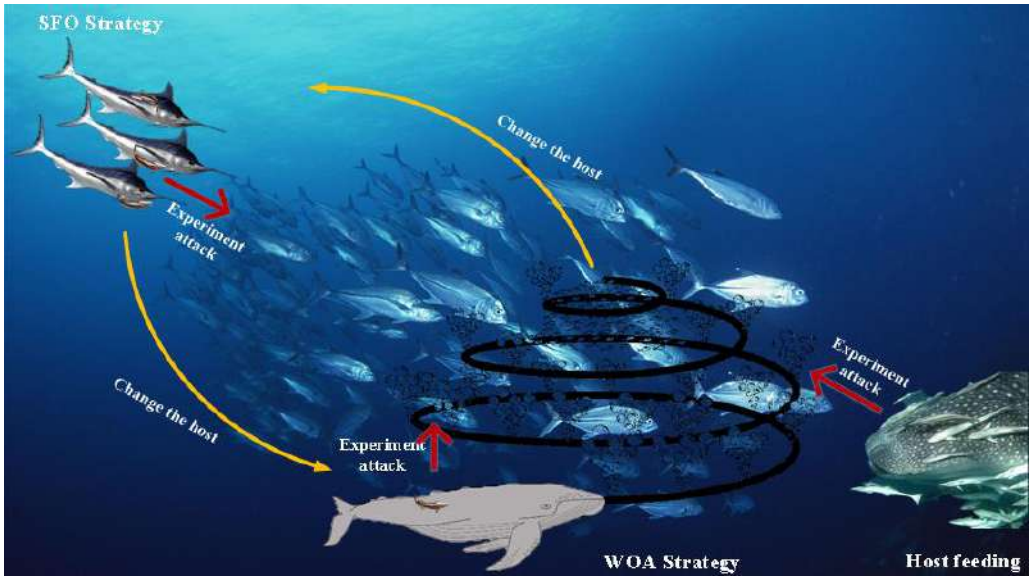

Figure 1. The different states of ROA.

ROA has many advantages such as:

- Easy-to-implement;
- Few number of parameters;
- Good balance between exploration and exploitation.

Moreover, like all other metaheurstics algorithms, it may stuck in local optima or have a slow convergence curve. A brief description of ROA's mathematical model can be described as follows.

2.1. Free Travel (Exploration)

2.1.1. Sailfish Optimization (SFO) Strategy

When Remora attach to the swordfish, its position can be considered as the swordfish's position. ROA improves the location update formula based on the elite idea and obtains the following formula:

$$X(t+1) = X_{Best}(t) - \left(rand \cdot \left(\frac{X_{Best}(t) + X_{rand}(t)}{2} \right) - X_{rand}(t) \right) \quad (1)$$

where t is the number of current iterations; $X_{Best}(t)$ represents the best solution obtained so far and $X_{rand}(t)$ indicates a random location; and *rand* is a uniformly distributed random number between 0 and 1.

2.1.2. Experience Attempt

At the same time, Remora continuously takes a small step around the host to accumulate experience to determine whether the host needs to be replaced. The mathematical formula is as follows.

$$X_{att}(t+1) = X(t) + \left(X(t) - X_{pre}(t) \right) \cdot randn \quad (2)$$

where $X_{att}(t+1)$ represents a tentative step; $X_{pre}(t)$ is the position of the previous generation and $X(t)$ indicates the current position; and *randn* is a normally distributed random number between 0 and 1.

After this "small global" movement, Remora will compare fitness values of SFO Strategy $f(X)$ and experience attempt $f(X_{att})$ to choose whether to change the host. The position with a smaller fitness value is retained.

2.2. Eat Thoughtfully (Exploitation)

2.2.1. Whale Optimization Algorithm (WOA) Strategy

When Remora attaches to the whale, the position update formula of remora is described as follows.

$$X(t+1) = Dist \cdot e^{\alpha} \cdot \cos(2\pi\alpha) + X(t) \tag{3}$$

$$\alpha = rand \cdot (a-1) + 1 \tag{4}$$

$$a = -\left(1 + \frac{t}{T}\right) \tag{5}$$

$$Dist = |X_{Best}(t) - X(t)| \tag{6}$$

where T is the maximum number of iterations, $Dist$ indicates the distance between the best position and the current position, α is a random number in $[-1, 1]$, and a linearly decreases from -1 to -2.

2.2.2. Host Feeding

"Host feeding" is a small step in the exploitation process, which creates a solution space that converges gradually around the host, refining and enhancing the ability of local optimization. This stage can be mathematically modeled as:

$$X(t+1) = X(t) + A \tag{7}$$

$$A = B \cdot (X(t) - C \cdot X_{Best}(t)) \tag{8}$$

$$B = 2V \cdot rand - V \tag{9}$$

$$V = 2\left(1 - \frac{t}{T}\right) \tag{10}$$

where A denotes a small step movement related to the volume space of the host and Remora, and factor C is a constant number equal to 0.1, used to narrow the position of remora.

It is worth noting that a random integer argument H (0 or 1) is used to decide whether to choose the WOA Strategy or the SFO Strategy. The pseudo-code of ROA is shown in Algorithm 1.

Algorithm 1 Pseudo-code of ROA

```
Set initial values of the population size N and the maximum number of iterations T
Initialize positions of the population X_i (i = 1, 2, 3, ..., N)
Initialize the best solution X_best and corresponding best fitness f(X_best)
While t < T do
    Calculate the fitness value of each Remora
    Check if any search agent goes beyond the search space and amend it
    Update a, α, V and H
    For each Remora indexed by i do
        If H(i) = 0 then
            Update the position using Equation (3)
        Elseif H(i) = 1 then
            Update the position using Equation (1)
        Endif
        Make a one-step prediction by Equation (2)
        Compare fitness values to judge whether host replacement is necessary
        If the host is not replaced, Equation (7) is used as the host feeding mode for Remora
    End for
End while
Return X_best
```

3. The Proposed Approach

As a newly proposed algorithm, ROA has achieved good results on some test functions. However, experiment results show that it still has the defects of insufficient global exploration and local optimum stagnation. The global exploration is implemented by the SFO Strategy and the "small global" movement experience attempt. The lack of global exploration capacity can be attributed to the deficient SFO Strategy. Thus, adaptive dynamic probability and Levy flight are utilized to improve the global search ability in this work. Meanwhile, a restart strategy is added to help the algorithm escape from local optima.

To best of our knowledge, it is the first time the following there operators have been combined with ROA.

3.1. Adaptive Dynamic Probability

As mentioned above, H is used to decide whether to choose the WOA Strategy or the SFO Strategy; that is, H determines whether to explore or exploit the search space. However, H is a random integer number, which means that the probability of exploration and exploitation is the same, whether during early or late iterations. This is not in line with our desire to focus on exploration in the early stage and on exploitation in the later stage for the optimization algorithm. Thus, the adaptive dynamic probability of H is designed as follows:

$$\begin{cases} p(H=0) = \frac{t}{T} \\ p(H=1) = 1 - \frac{t}{T} \end{cases} \tag{11}$$

where p denotes the probability that H takes 0 or 1; obviously, with the increase of the number of iterations, the probability of H taking 0 increases, while the probability of 1 decreases. The possibility of individuals for exploited increases and the possibility of exploration decreases.

3.2. Sailfish Optimization (SFO) Strategy with Levy Flight

Levy flight is a stochastic strategy widely used in optimization algorithms. It has a relatively high probability of large strides in random walking, which can effectively improve the randomness of the algorithm. To further enhance the exploration ability of the method, Levy flight is integrated into the formula of the SFO Strategy, which is described as follows:

$$X(t+1) = X_{Best}(t) - \left(rand \cdot \left(\frac{X_{Best}(t) + X_{rand}(t)}{2} \right) - X_{rand}(t) \right) \cdot Levy(D) \tag{12}$$

$$Levy(D) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{13}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \tag{14}$$

where *Levy* represents the Levy flight function, and D is the dimension size of the problem. u and v are random values between 0 and 1, and β is a constant number equal to 1.5.

3.3. Restart Strategy (RS)

Restart schemes can help worse individuals jump out of the local optimum, so they are used to prevent the population from stagnating. Zhang et al. [58] proposed a RS with a trial vector recording the times the position of individuals has not been improved. If the position of the ith individual has not been improved in this search, the trial value of this individual is increased by 1. Otherwise, the trial value is reset to zero. If the trail value is

not less than the predefined *Limit*, the position will be replaced by choosing the location with a better fitness value from Equations (15) and (16).

$$X(t+1) = lb + rand \cdot (ub - lb) \tag{15}$$

$$X(t+1) = rand \cdot (ub + lb) - X(t) \tag{16}$$

where lb and ub are the lower and upper bound of the problem, respectively; in this paper, we replace Equation (16) with Equation (17) from the random opposition-based learning (ROL) strategy [59] to obtain an opposite position. A better solution generated from Equations (15) and (17) is adopted if the trail value is not less than the Limit.

$$X(t+1) = (ub + lb) - rand \cdot X(t) \tag{17}$$

3.4. The Proposed EROA

EROA is proposed to combine the above three strategies. The overall process of the EROA is similar to ROA, except that the update method of H is replaced by adaptive dynamic probability, the SFO Strategy integrates Levy flight, and the RS is added at the end. The pseudo-code of EROA is given in Algorithm 2, and the summarized flowchart is illustrated in Figure 2.

Algorithm 2 Pseudo-code of EROA

```
Set initial values of the population size N and the maximum number of iterations T
Initialize positions of the population X_i (i = 1, 2, 3, ..., N)
Initialize the best solution X_best and corresponding best fitness f(X_best)
While t < T do
    Calculate the fitness value of each Remora
    Check if any search agent goes beyond the search space and amend it
    Update a, α, and V
    Update H based on Equation (11)
    For each Remora indexed by i do
        If H(i) = 0 then
            Update the position using Equation (3)
        Elseif H(i) = 1 then
            Update the position using Equation (12)
        End if
        Make a one-step prediction by Equation (2)
        Compare fitness values to judge whether host replacement is necessary
        If the host is not replaced, Equation (7) is used as the host feeding mode for Remora
        Update trial(i) for remora
        If trial(i) >= Limit
            Generate positions using Equations (15) and (17), respectively
            Compare fitness values to choose the position with better fitness value
        End if
    End for
End while
Return X_best
```

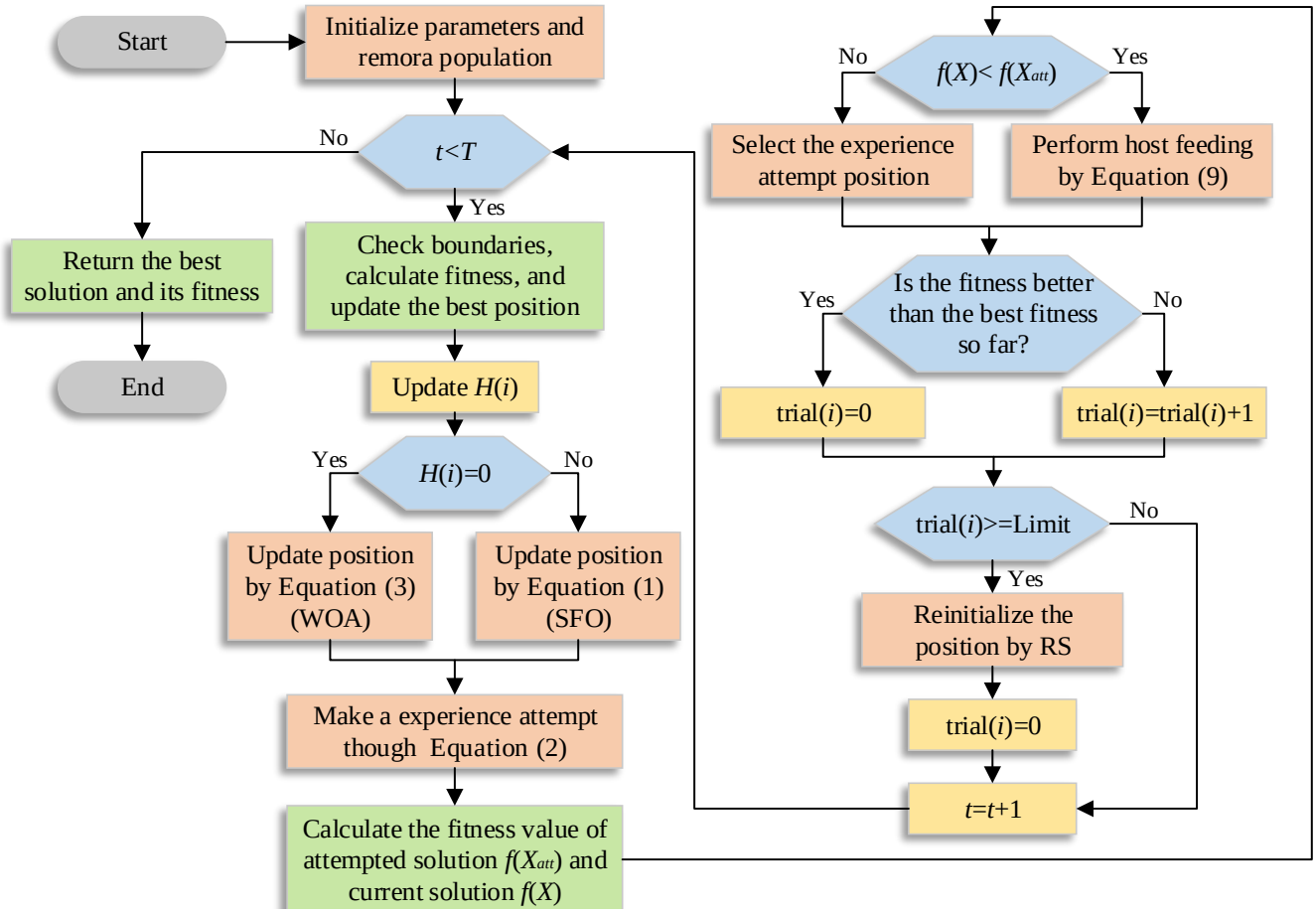

Figure 2. The flowchart of the EROA.

4. Numerical Experiment Results

In this section, two different types of benchmark functions are used to evaluate the performance of the EROA. First, experiments on 23 standard benchmark functions are carried out to evaluate the performance of EROA in solving simple numerical optimization problems. Then, the CEC2017 test suite, including 29 benchmark functions, is utilized to evaluate the performance of EROA in solving complex numerical problems. The EROA is compared with seven well-known metaheuristic methods, including ROA, AO, AOA, HHO, WOA, SCA, and STOA [60]. We set the population size N = 30, dimension size D = 30/100/500, the maximum number of iterations T = 500, and run 30 times independently for all algorithms. The parameter settings of each algorithm are shown in Table 1. All experiments are carried out in MATLAB R2016a on a PC with Intel (R) Core (TM) i7-9700 CPU @ 3.00 GHz and RAM 8 GB memory on OS Windows 10.

Table 1. Parameter settings for the comparative algorithms.

Algorithm	Parameters
EROA	$C = 0.1$; Limit = log(t)
ROA	$C = 0.1$
AO	$U = 0.00565$; $r_1 = 10$; $\omega = 0.005$; $\alpha = 0.1$; $\delta = 0.1$; $G_1 \in [-1, 1]$; $G_2 = [2, 0]$
AOA	$\alpha = 5$; $\mu = 0.5$;
HHO	$q \in [0, 1]$; $r \in [0, 1]$; $E_0 \in [-1, 1]$; $E_1 = [2, 0]$; $E \in [-2, 2]$;
WOA	$a_1 = [2, 0]$; $a_2 = [-1, -2]$; $b = 1$
STOA	$C_f = 2$; $u = 1$; $v = 1$

4.1. Experiments on Standard Benchmark Functions

Here, the EROA performance is tested using 23 mathematical benchmark functions. This benchmark contains seven unimodal, six multimodal, and ten fixed-dimension multimodal functions. The mathematical description of each type is given in Tables 2–4 where Fun refers to a mathematical function, D refers to the number of dimensions, Range shows the interval of search space, $f_{\min}$ refers to the optimal value that the corresponding functions can achieve.

Table 5 shows the results of the introduced algorithm with its competitors. The parameter settings of each algorithm are illustrated in Table 1. From Table 5, it can be seen that EROA ranked first in 19 functions out of 23 ones. In unimodal functions, it ranked first in 5 out of 7, whereas in multimodal, it achieves the best results in 4 out of 6. On the other hand, it achieves the best in all functions that belong to fixed-dimension multimodal functions. Figure 3 shows the convergence curve for all functions. From this table, it can be noticed that EROA has a faster convergence than other competitors.

To test the scalability of the proposed algorithm, we carry out the experiments from F1–F13 using two other dimensions $D = 100$ and $D = 500$ as shown in Table 5. Moreover, the convergence curve for these dimensions are shown in Figures 4 and 5.

Furthermore, a non-parametric test called Wilcoxon rank-sum is used at a 5% level of significance to make a fair comparison between EROA and other algorithm results in each independent run. Table 6 shows the results of such parameters. From this table, it can be seen that p-values for almost functions are less than 0.05.

Table 2. Unimodal benchmark functions.

Fun.	D	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/100/500	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30/100/500	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30/100/500	$[-100, 100]$	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30/100/500	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30/100/500	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{n} (x_i + 5)^2$	30/100/500	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	30/100/500	$[-1.28, 1.28]$	0

Table 3. Multimodal benchmark functions.

Fun.	D	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30/100/500	$[-500, 500]$	$-418.9829 \times D$
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30/100/500	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30/100/500	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30/100/500	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+\sum_{i=1}^{n} u(x_i, 10, 100, 4)$, where $y_i = 1 + \frac{x_i+1}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30/100/500	$[-50, 50]$	0
$F_{13}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)]) + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30/100/500	$[-50, 50]$	0

Table 4. Fixed-dimension multimodal benchmark functions.

Fun.	D	Range	$f_{\min}$
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	$[-65, 65]$	1
$F_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	$[-5, 5]$	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + x_2^4$	2	$[-5, 5]$	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_2 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	$[-1, 2]$	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	$[0, 1]$	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5363

Table 5. Results of algorithms on 23 benchmark functions.

F	*D*		EROA	ROA	AO	AOA	HHO	WOA	SCA	STOA
F1	30	Avg	0	7.33×10^{-314}	2.73×10^{-102}	4.83×10^{-6}	2.34×10^{-95}	4.15×10^{-72}	9.65	5.81×10^{-7}
		Std	0	0	1.49×10^{-101}	1.81×10^{-6}	1.26×10^{-94}	2.25×10^{-71}	1.71×10^{1}	1.24×10^{-6}
	100	Avg	0	0	1.47×10^{-98}	9.93×10^{-4}	5.55×10^{-93}	9.06×10^{-72}	1.08×10^{4}	9.12×10^{-3}
		Std	0	0	8.02×10^{-98}	2.80×10^{-4}	2.05×10^{-92}	2.16×10^{-71}	7.78×10^{3}	1.22×10^{-2}
	500	Avg	0	0	7.28×10^{-102}	5.24×10^{-1}	9.52×10^{-97}	4.70×10^{-70}	2.17×10^{5}	9.52
		Std	0	0	3.86×10^{-101}	3.36×10^{-2}	3.92×10^{-96}	1.35×10^{-69}	7.89×10^{4}	9.27
F2	30	Avg	0	1.16×10^{-165}	4.63×10^{-64}	1.50×10^{-3}	9.36×10^{-51}	6.83×10^{-50}	1.69×10^{-2}	1.01×10^{-5}
		Std	0	0	2.32×10^{-63}	1.92×10^{-3}	3.73×10^{-50}	3.45×10^{-49}	2.25×10^{-2}	1.06×10^{-5}
	100	Avg	0	7.40×10^{-162}	5.90×10^{-55}	1.85×10^{-2}	2.07×10^{-50}	1.24×10^{-49}	$1.27\text{E} \times 10^{1}$	2.65×10^{-3}
		Std	0	4.04×10^{-161}	2.61×10^{-54}	2.34×10^{-3}	8.48×10^{-50}	5.08×10^{-49}	1.03×10^{1}	2.05×10^{-3}
	500	Avg	0	9.24×10^{-160}	1.46×10^{-57}	5.24×10^{-1}	1.17×10^{-48}	1.22×10^{-47}	1.11×10^{2}	9.34×10^{-2}
		Std	0	3.95×10^{-159}	8.01×10^{-57}	9.93×10^{-2}	5.95×10^{-48}	6.19×10^{-47}	7.32×10^{1}	6.76×10^{-2}
F3	30	Avg	0	1.68×10^{-289}	9.44×10^{-111}	9.51×10^{-4}	2.69×10^{-67}	4.09×10^{4}	1.11×10^{4}	7.91×10^{-2}
		Std	0	0	3.78×10^{-110}	7.68×10^{-4}	1.47×10^{-66}	1.39×10^{4}	7.68×10^{3}	9.41×10^{-2}
	100	Avg	0	3.32×10^{-276}	5.31×10^{-100}	1.30×10^{-1}	3.89×10^{-62}	1.09×10^{6}	2.49×10^{5}	2.12×10^{3}
		Std	0	0	2.91×10^{-99}	3.20×10^{-2}	2.11×10^{-61}	3.07×10^{5}	4.85×10^{4}	3.24×10^{3}
	500	Avg	0	1.09×10^{-253}	3.30×10^{-104}	6.91	4.61×10^{-39}	3.38×10^{7}	7.15×10^{6}	5.89×10^{5}
		Std	0	0	1.02×10^{-103}	1.24	2.53×10^{-38}	1.05×10^{7}	1.79×10^{6}	2.47×10^{5}
F4	30	Avg	0	3.71×10^{-156}	6.47×10^{-53}	1.67×10^{-2}	3.18×10^{-48}	5.54×10^{1}	3.20×10^{1}	5.18×10^{-2}
		Std	0	2.03×10^{-155}	2.57×10^{-52}	1.24×10^{-2}	1.71×10^{-47}	2.37×10^{1}	1.33×10^{1}	5.14×10^{-2}
	100	Avg	0	5.19×10^{-156}	1.20×10^{-55}	5.57×10^{-3}	5.67×10^{-48}	7.59×10^{1}	8.97×10^{1}	7.04×10^{1}
		Std	0	2.83×10^{-155}	6.59×10^{-55}	5.81×10^{-3}	2.84×10^{-47}	2.24×10^{1}	3.28	1.61×10^{1}
	500	Avg	0	3.95×10^{-152}	5.72×10^{-54}	1.21×10^{-1}	4.10×10^{-49}	8.22×10^{1}	9.91×10^{1}	9.87×10^{1}
		Std	0	2.16×10^{-151}	2.85×10^{-53}	9.18×10^{-3}	1.93×10^{-48}	2.13×10^{1}	3.52×10^{-1}	5.42×10^{-1}
F5	30	Avg	6.52×10^{-2}	2.71×10^{1}	9.40×10^{-3}	2.79×10^{1}	1.28×10^{-2}	2.79×10^{1}	7.71×10^{4}	2.84×10^{1}
		Std	1.58×10^{-2}	4.46×10^{-1}	2.72×10^{-2}	2.37×10^{-1}	1.39×10^{-2}	4.94×10^{-1}	2.34×10^{5}	5.01×10^{-1}
	100	Avg	2.73×10^{-1}	9.76×10^{1}	2.29×10^{-2}	9.82×10^{1}	3.42×10^{-2}	9.81×10^{1}	1.08×10^{8}	1.06×10^{2}
		Std	5.56×10^{-1}	4.54×10^{-1}	3.13×10^{-2}	5.63×10^{-2}	4.13×10^{-2}	2.42×10^{-1}	3.96×10^{7}	6.54
	500	Avg	8.84×10^{-1}	4.95×10^{2}	1.00×10^{-1}	4.99×10^{2}	2.39×10^{-1}	4.96×10^{2}	2.06×10^{9}	1.23×10^{4}
		Std	1.97	3.06×10^{-1}	1.26×10^{-1}	1.37×10^{-1}	4.50×10^{-1}	4.25×10^{-1}	4.56×10^{8}	1.07×10^{4}

Table 5. *Cont.*

F	*D*		EROA	ROA	AO	AOA	HHO	WOA	SCA	STOA
F6	30	Avg	4.53×10^{-4}	1.05×10^{-1}	1.52×10^{-4}	3.02	9.07×10^{-5}	4.33×10^{-1}	2.06×10^{1}	2.59
		Std	2.66×10^{-4}	9.54×10^{-2}	4.51×10^{-4}	2.46×10^{-1}	1.74×10^{-4}	2.21×10^{-1}	3.95×10^{1}	5.09×10^{-1}
	100	Avg	1.00×10^{-2}	1.84	9.23×10^{-4}	1.59×10^{1}	4.92×10^{-4}	4.33	1.37×10^{4}	1.77×10^{1}
		Std	1.50×10^{-2}	5.34×10^{-1}	2.94×10^{-3}	7.25×10^{-1}	5.53×10^{-4}	1.21	8.42×10^{3}	7.95×10^{-1}
	500	Avg	9.47×10^{-2}	1.56×10^{1}	6.07×10^{-4}	1.12×10^{2}	1.82×10^{-3}	3.31×10^{1}	2.11×10^{5}	1.24×10^{2}
		Std	1.28×10^{-1}	4.43	8.88×10^{-4}	1.64	2.10×10^{-3}	9.98	6.42×10^{4}	6.66
F7	30	Avg	8.37×10^{-5}	1.36×10^{-4}	1.01×10^{-4}	9.06×10^{-5}	1.57×10^{-4}	4.18×10^{-3}	9.69×10^{-2}	5.85×10^{-3}
		Std	5.99×10^{-5}	1.20×10^{-4}	6.70×10^{-5}	8.11×10^{-5}	1.32×10^{-4}	4.43×10^{-3}	1.13×10^{-1}	2.83×10^{-3}
	100	Avg	1.11×10^{-4}	1.57×10^{-4}	1.22×10^{-4}	7.75×10^{-5}	1.57×10^{-4}	4.78×10^{-3}	1.66×10^{2}	2.58×10^{-2}
		Std	1.02×10^{-4}	2.34×10^{-4}	1.30×10^{-4}	7.26×10^{-5}	1.61×10^{-4}	4.32×10^{-3}	9.84×10^{1}	1.10×10^{-2}
	500	Avg	7.39×10^{-5}	1.47×10^{-4}	6.99×10^{-5}	6.12×10^{-5}	1.77×10^{-4}	4.38×10^{-3}	1.44×10^{4}	4.94×10^{-1}
		Std	7.05×10^{-5}	1.24×10^{-4}	5.04×10^{-5}	6.66×10^{-5}	1.94×10^{-4}	5.40×10^{-3}	4.06×10^{3}	2.40×10^{-1}
F8	30	Avg	-1.26×10^{4}	-1.23×10^{4}	-7.48×10^{3}	-5.34×10^{3}	-1.26×10^{4}	-1.03×10^{4}	-3.91×10^{3}	-5.12×10^{3}
		Std	2.20×10^{-2}	7.21×10^{2}	3.74×10^{3}	3.69×10^{2}	9.54×10^{1}	1.74×10^{3}	3.78×10^{2}	4.44×10^{2}
	100	Avg	-4.19×10^{4}	-4.14×10^{4}	-1.09×10^{4}	-1.40×10^{4}	-4.19×10^{4}	-3.52×10^{4}	-6.83×10^{3}	-1.10×10^{4}
		Std	1.79×10^{-1}	1.45×10^{3}	6.20×10^{3}	7.36×10^{2}	3.35	6.14×10^{3}	5.65×10^{2}	1.51×10^{3}
	500	Avg	-2.09×10^{5}	-2.07×10^{5}	-3.90×10^{4}	-3.82×10^{4}	-2.09×10^{5}	-1.66×10^{5}	-1.55×10^{4}	-2.50×10^{4}
		Std	3.46	5.11×10^{3}	1.04×10^{4}	1.54×10^{3}	2.02×10^{3}	2.86×10^{4}	1.28×10^{3}	3.26×10^{3}
F9	30	Avg	0	0	0	1.20×10^{-6}	0	1.89×10^{-15}	4.33×10^{1}	1.26×10^{1}
		Std	0	0	0	1.08×10^{-6}	0	1.04×10^{-14}	3.79×10^{1}	1.67×10^{1}
	100	Avg	0	0	0	1.85×10^{-4}	0	0	3.23×10^{2}	1.26×10^{1}
		Std	0	0	0	3.93×10^{-5}	0	0	1.08×10^{2}	8.97
	500	Avg	0	0	3.03×10^{-14}	1.11×10^{-2}	0	0	1.44×10^{3}	2.76×10^{1}
		Std	0	0	1.66×10^{-13}	7.83×10^{-4}	0	0	5.64×10^{2}	1.88×10^{1}
F10	30	Avg	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	4.15×10^{-4}	8.88×10^{-16}	4.91×10^{-15}	1.21×10^{1}	1.99×10^{1}
		Std	0	0	0	1.78×10^{-4}	0	2.23×10^{-15}	9.25	1.49×10^{-3}
	100	Avg	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	3.42×10^{-3}	8.88×10^{-16}	4.91×10^{-15}	1.81×10^{1}	2.00×10^{1}
		Std	0	0	0	3.76×10^{-4}	0	3.06×10^{-15}	4.95	3.43×10^{-4}
	500	Avg	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	2.67×10^{-2}	8.88×10^{-16}	3.26×10^{-15}	1.93E $\times 10^{1}$	2.00×10^{1}
		Std	0	0	0	9.11×10^{-4}	0	2.35×10^{-15}	3.32	6.17×10^{-5}
F11	30	Avg	0	0	0	1.09×10^{-3}	0	2.49×10^{-2}	8.85×10^{-1}	1.81×10^{-2}
		Std	0	0	0	4.17×10^{-3}	0	7.26×10^{-2}	3.10×10^{-1}	2.16×10^{-2}
	100	Avg	0	0	0	1.42×10^{-1}	0	0	8.53×10^{1}	4.58×10^{-2}
		Std	0	0	0	1.61×10^{-1}	0	0	6.15×10^{1}	5.97×10^{-2}
	500	Avg	0	0	0	1.35×10^{3}	0	0	1.85×10^{3}	6.45×10^{-1}
		Std	0	0	0	5.15×10^{2}	0	0	7.02×10^{2}	2.99×10^{-1}

Table 5. *Cont.*

F	*D*		EROA	ROA	AO	AOA	HHO	WOA	SCA	STOA
F12	30	Avg	1.35×10^{-5}	9.35×10^{-3}	6.63×10^{-6}	7.42×10^{-1}	1.55×10^{-5}	1.66×10^{-1}	1.32×10^{5}	2.62×10^{-1}
		Std	1.23×10^{-5}	6.03×10^{-3}	1.26×10^{-5}	2.08×10^{-2}	1.93×10^{-5}	7.90×10^{-1}	5.71×10^{5}	1.71×10^{-1}
	100	Avg	6.55×10^{-6}	2.20×10^{-2}	1.07×10^{-6}	9.10×10^{-1}	2.57×10^{-5}	5.52×10^{-2}	3.21×10^{8}	7.79×10^{-1}
		Std	9.69×10^{-6}	1.10×10^{-2}	2.81×10^{-6}	6.55×10^{-2}	3.17×10^{-5}	2.34×10^{-2}	1.69×10^{8}	1.20×10^{-1}
	500	Avg	2.18×10^{-5}	4.56×10^{-2}	9.43×10^{-7}	9.34×10^{-1}	2.24×10^{-6}	9.73×10^{-2}	6.35×10^{9}	4.73
		Std	2.79×10^{-5}	2.84×10^{-2}	1.11×10^{-6}	2.56×10^{-2}	3.24×10^{-6}	4.54×10^{-2}	1.21×10^{9}	2.46
F13	30	Avg	2.76×10^{-4}	1.85×10^{-1}	4.18×10^{-5}	2.96	9.08×10^{-5}	5.31×10^{-1}	3.24×10^{5}	1.93
		Std	2.19×10^{-4}	1.13×10^{-1}	9.53×10^{-5}	1.77×10^{-2}	1.09×10^{-4}	3.47×10^{-1}	1.51×10^{5}	2.57×10^{-1}
	100	Avg	1.79×10^{-3}	1.32	8.10×10^{-5}	9.92	1.33×10^{-4}	2.73	5.67×10^{8}	1.03×10^{1}
		Std	3.51×10^{-3}	6.70×10^{-1}	9.74×10^{-5}	8.17×10^{-3}	1.95×10^{-4}	8.22×10^{-1}	$2..41 \times 10^{8}$	6.93×10^{-1}
	500	Avg	6.08×10^{-3}	7.51	3.43×10^{-4}	4.93×10^{1}	5.18×10^{-4}	2.00	9.35×10^{9}	1.62×10^{2}
		Std	1.09×10^{-2}	3.92	8.03×10^{-4}	2.68×10^{-1}	6.87×10^{-4}	7.64	1.98×10^{9}	5.41×10^{1}
F14	2	Avg	9.98×10^{-1}	4.91	3.47	9.15	1.16	3.41	1.59	2.18
		Std	1.47×10^{-11}	4.57	4.09	4.42	3.77×10^{-1}	3.82	9.22×10^{-1}	2.49
F15	4	Avg	3.13×10^{-4}	5.56×10^{-4}	4.90×10^{-4}	4.97×10^{-3}	3.43×10^{-4}	6.84×10^{-4}	9.41×10^{-4}	3.61×10^{-3}
		Std	1.57×10^{-5}	3.12×10^{-4}	2.87×10^{-4}	9.79×10^{-3}	2.89×10^{-5}	3.35×10^{-4}	3.13×10^{-4}	6.69×10^{-3}
F16	2	Avg	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03
		Std	7.94×10^{-9}	4.88×10^{-8}	7.48×10^{-4}	1.68×10^{-11}	2.81×10^{-9}	3.75×10^{-9}	5.08×10^{-5}	2.19×10^{-6}
F17	2	Avg	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.99×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.99×10^{-1}	3.98×10^{-1}
		Std	2.46×10^{-7}	9.09×10^{-6}	1.65×10^{-4}	4.72×10^{-3}	1.36×10^{-5}	5.86×10^{-6}	2.04×10^{-3}	1.01×10^{-4}
F18	2	Avg	3.00	3.00	3.04	1.74×10^{1}	3.00	3.00	3.00	3.00
		Std	1.75×10^{-5}	1.03×10^{-4}	4.20×10^{-2}	2.53×10^{1}	4.71×10^{-7}	6.68×10^{-5}	1.90×10^{-4}	1.37×10^{-4}
F19	3	Avg	−3.86	−3.86	−3.86	−3.77	−3.86	−3.86	−3.85	−3.86
		Std	3.06×10^{-6}	1.66×10^{-3}	6.05×10^{-3}	5.22×10^{-1}	2.90×10^{-3}	6.04×10^{-3}	1.04×10^{-2}	4.95×10^{-3}
F20	6	Avg	−3.26	−3.25	−3.13	−3.27	−3.12	−3.20	−2.82	−2.89
		Std	7.76×10^{-2}	8.86×10^{-2}	1.05×10^{-1}	5.93×10^{-2}	8.94×10^{-2}	1.19×10^{-1}	4.77×10^{-1}	5.65×10^{-1}
F21	4	Avg	-1.02×10^{1}	-1.01×10^{1}	-1.01×10^{1}	−8.05	−5.37	−8.28	−2.29	−3.82
		Std	1.85×10^{-4}	2.19×10^{-2}	4.20×10^{-2}	2.66	1.24	2.73	1.87	4.31
F22	4	Avg	-1.04×10^{1}	1.04×10^{1}	-1.04×10^{1}	−6.83	−5.25	−7.79	−3.07	−5.96
		Std	1.59×10^{-4}	1.74×10^{-2}	1.62×10^{-2}	3.72	9.14×10^{-1}	3.07	1.60	4.34
F23	4	Avg	-1.05×10^{1}	1.05×10^{1}	-1.05×10^{1}	−8.13	−5.62	−7.35	−3.37	−6.96
		Std	1.76×10^{-4}	1.95×10^{-2}	2.68×10^{-2}	3.30	1.52	3.08	1.87	3.94

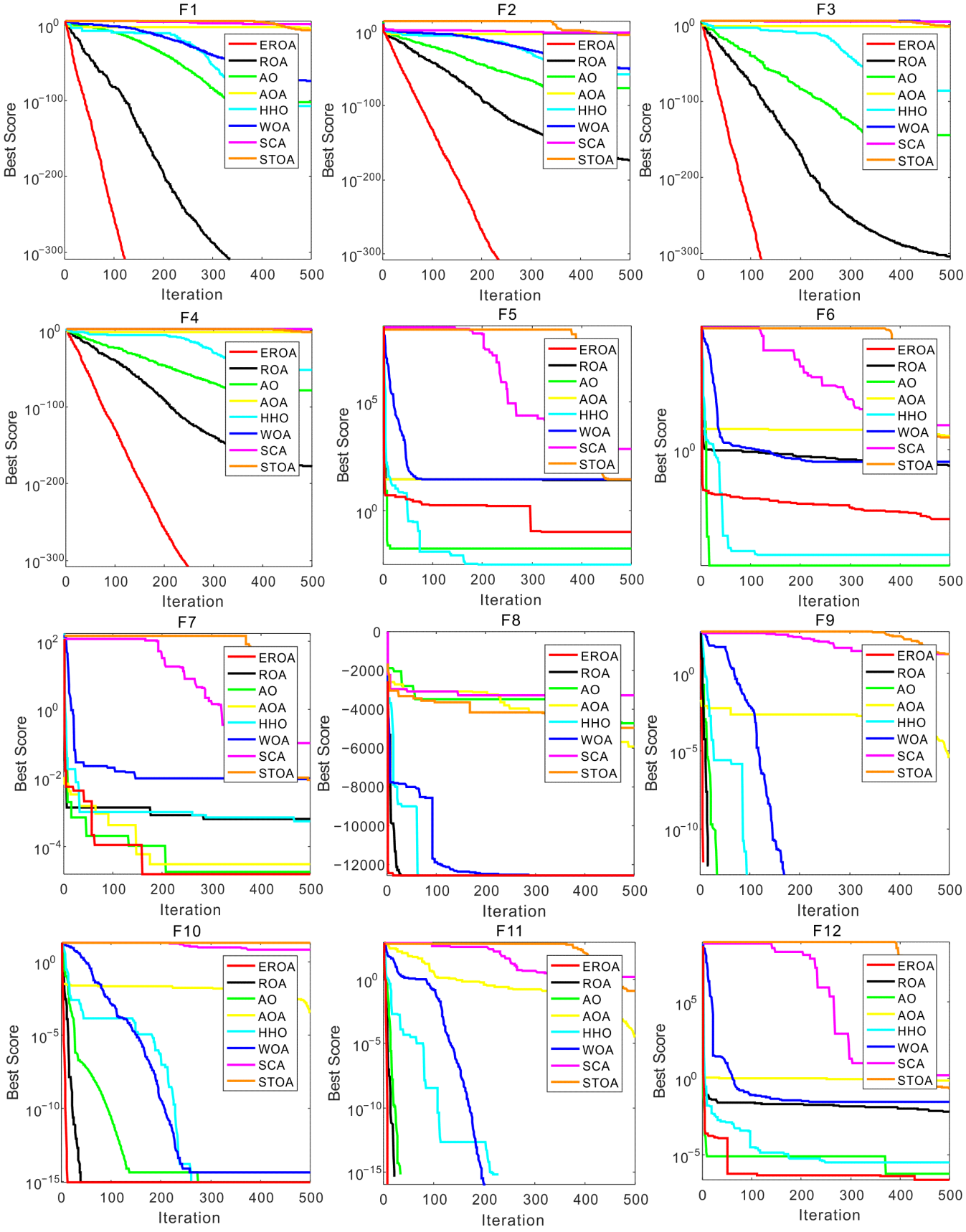

Figure 3. *Cont.*

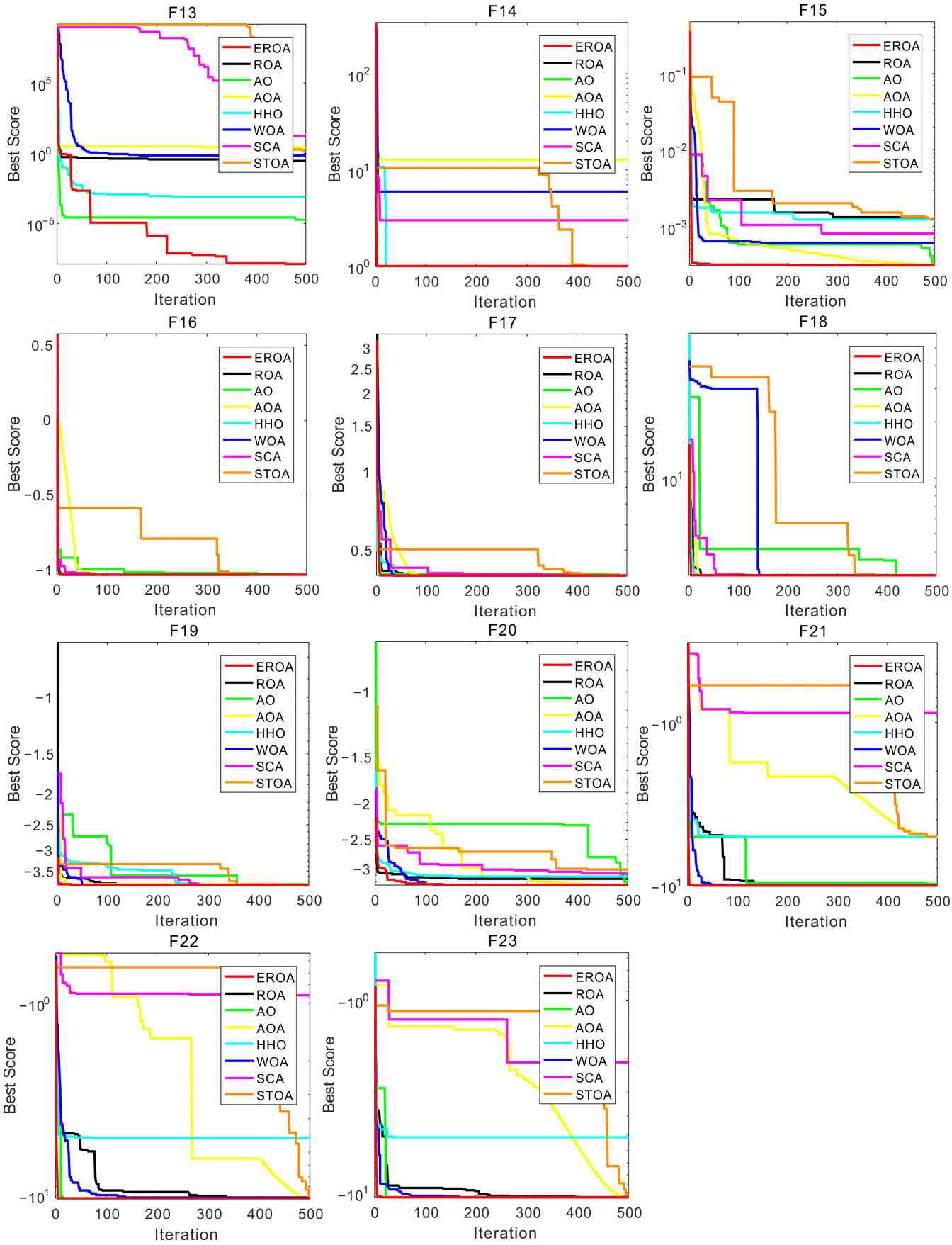

Figure 3. Convergence curves for the optimization algorithms for standard benchmark functions.

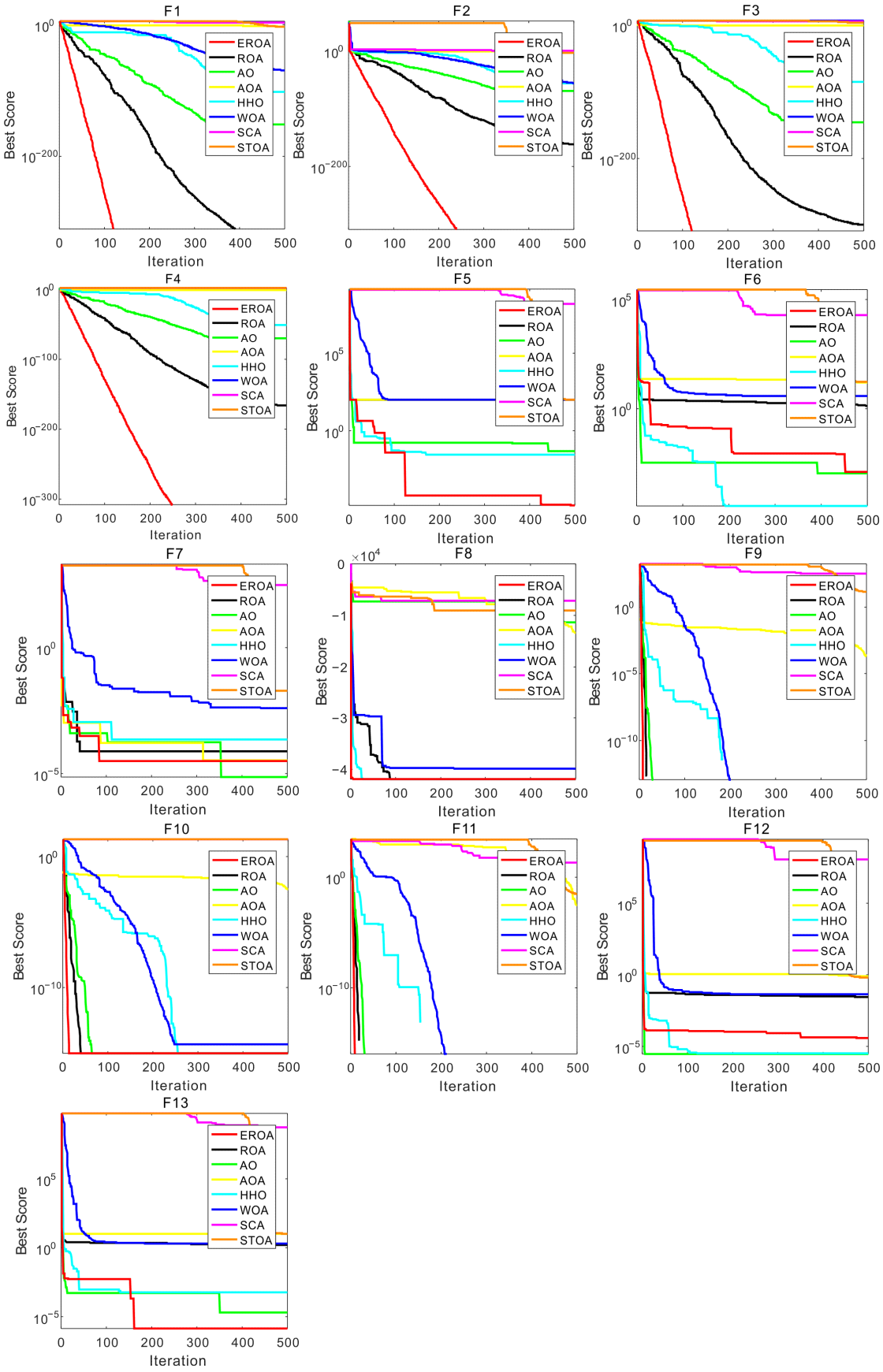

Figure 4. Convergence curves for the optimization algorithms for standard benchmark functions with $D = 100$.

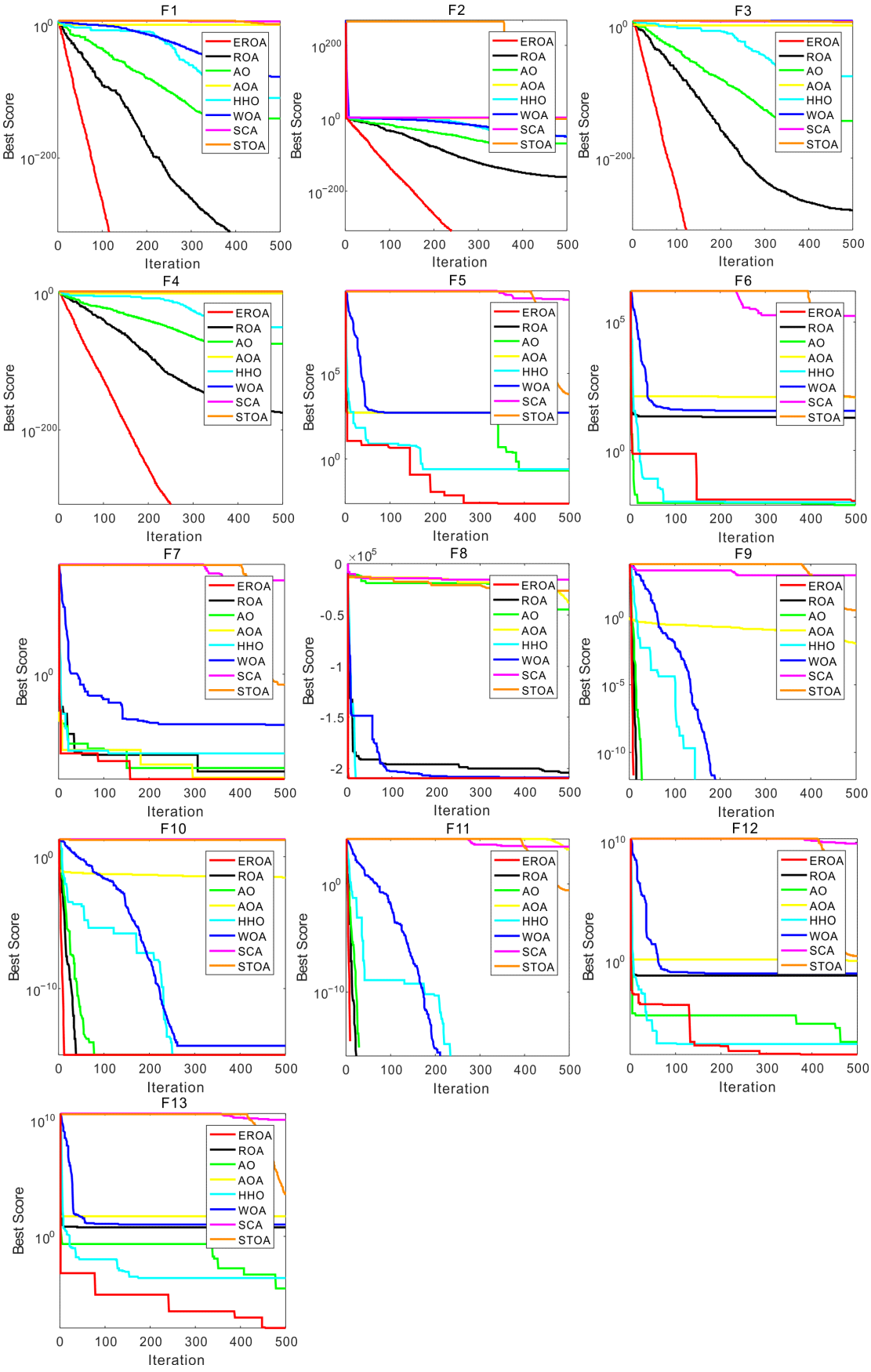

Figure 5. Convergence curves for the optimization algorithms for standard benchmark functions with $D = 500$.

Table 6. *p*-Values from the Wilcoxon rank-sum test for the results in Table 5.

F	*D*	EROA vs. ROA	EROA vs. AO	EROA vs. AOA	EROA vs. HHO	IHAOHHO vs. WOA	EROA vs. SCA	EROA vs. STOA
F1	30	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	N/A	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	N/A	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F2	30	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F3	30	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F4	30	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F5	30	6.1035×10^{-5}	5.5359×10^{-2}	6.1035×10^{-5}	2.5238×10^{-1}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	2.1545×10^{-2}	6.1035×10^{-5}	4.1260×10^{-2}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	2.7686×10^{-1}	6.1035×10^{-5}	5.9949×10^{-1}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F6	30	6.1035×10^{-5}	8.5449×10^{-4}	6.1035×10^{-5}	6.7139×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	8.5449×10^{-4}	6.1035×10^{-5}	1.1597×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	8.5449×10^{-4}	6.1035×10^{-5}	6.7139×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F7	30	2.5238×10^{-2}	9.7797×10^{-2}	9.4606×10^{-3}	9.7797×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.3721×10^{-2}	9.3408×10^{-1}	4.2725×10^{-3}	7.1973×10^{-1}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	1.8762×10^{-1}	8.0396×10^{-1}	8.0396×10^{-1}	5.5359×10^{-2}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F8	30	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	1.1597×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	2.6245×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	4.2725×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	1.2207×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}	8.5449×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F9	30	N/A	N/A	1.2207×10^{-4}	N/A	N/A	6.1035×10^{-5}	6.1035×10^{-5}
	100	N/A	N/A	6.1035×10^{-5}	N/A	N/A	6.1035×10^{-5}	6.1035×10^{-5}
	500	N/A	N/A	6.1035×10^{-5}	N/A	N/A	6.1035×10^{-5}	6.1035×10^{-5}
F10	30	N/A	N/A	6.1035×10^{-5}	N/A	3.9063×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}
	100	N/A	N/A	6.1035×10^{-5}	N/A	4.8828×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}
	500	N/A	N/A	6.1035×10^{-5}	N/A	1.9531×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}
F11	30	N/A	N/A	6.1035×10^{-5}	N/A	N/A	6.1035×10^{-5}	6.1035×10^{-5}
	100	N/A	N/A	6.1035×10^{-5}	N/A	N/A	6.1035×10^{-5}	6.1035×10^{-5}
	500	N/A	N/A	6.1035×10^{-5}	N/A	N/A	6.1035×10^{-5}	6.1035×10^{-5}

Table 6. *Cont.*

F	*D*	EROA vs. ROA	EROA vs. AO	EROA vs. AOA	EROA vs. HHO	IHAOHHO vs. WOA	EROA vs. SCA	EROA vs. STOA
F12	30	6.1035×10^{-5}	3.3569×10^{-3}	6.1035×10^{-5}	1.3538×10^{-1}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	8.5449×10^{-4}	6.1035×10^{-5}	8.3252×10^{-2}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	3.3569×10^{-3}	6.1035×10^{-5}	8.3618×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F13	30	6.1035×10^{-5}	6.1035×10^{-4}	6.1035×10^{-5}	1.8066×10^{-1}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	100	6.1035×10^{-5}	1.5259×10^{-3}	6.1035×10^{-5}	4.7913×10^{-2}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
	500	6.1035×10^{-5}	4.2120×10^{-1}	6.1035×10^{-5}	4.5428×10^{-1}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F14	2	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	2.6245×10^{-3}	1.2207×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}
F15	4	5.5359×10^{-2}	8.3618×10^{-3}	5.3711×10^{-3}	3.8940×10^{-2}	2.0142×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}
F16	2	4.2725×10^{-3}	6.1035×10^{-5}	1.2207×10^{-4}	3.3569×10^{-3}	1.5259×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}
F17	2	1.1597×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.3721×10^{-2}	4.2725×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}
F18	2	1.8066×10^{-2}	8.3618×10^{-3}	6.1035×10^{-5}	2.1545×10^{-2}	1.0254×10^{-2}	8.3618×10^{-3}	8.3618×10^{-3}
F19	3	1.2207×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}	3.3569×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F20	6	1.5143×10^{-1}	6.7139×10^{-3}	5.9949×10^{-1}	6.1035×10^{-4}	5.3711×10^{-3}	6.1035×10^{-5}	6.1035×10^{-5}
F21	4	6.1035×10^{-5}	6.1035×10^{-4}	1.1597×10^{-3}	6.1035×10^{-5}	1.2207×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}
F22	4	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}
F23	4	6.1035×10^{-5}	8.5449×10^{-4}	1.2207×10^{-4}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}	6.1035×10^{-5}

4.2. Experiments on CEC2017 Test Suite

In this subsection, a discussion on the performance of the suggested algorithm named EROA using CEC2017 is provided. The results of EROA with other algorithms are given in Table 7. This table shows that EROA has achieved the best results in 9 functions out of 29, whereas STOA achieved the best results in only eight functions. Moreover, EROA has achieved the second-best in two functions (F20-F21) and the third-best in eight functions (F3-F5-F18-F19-F24-F26-F30). To make a fair comparison in algorithm ranking, we perform a Friedman test, as shown in Table 8. From this table, it can be noticed that EROA has ranked first overall in solving CEC2017. Furthermore, Figure 6 shows some convergence curves to the introduced algorithm compared with the classical ROA and other six functions. EROA achieved a fast convergence.

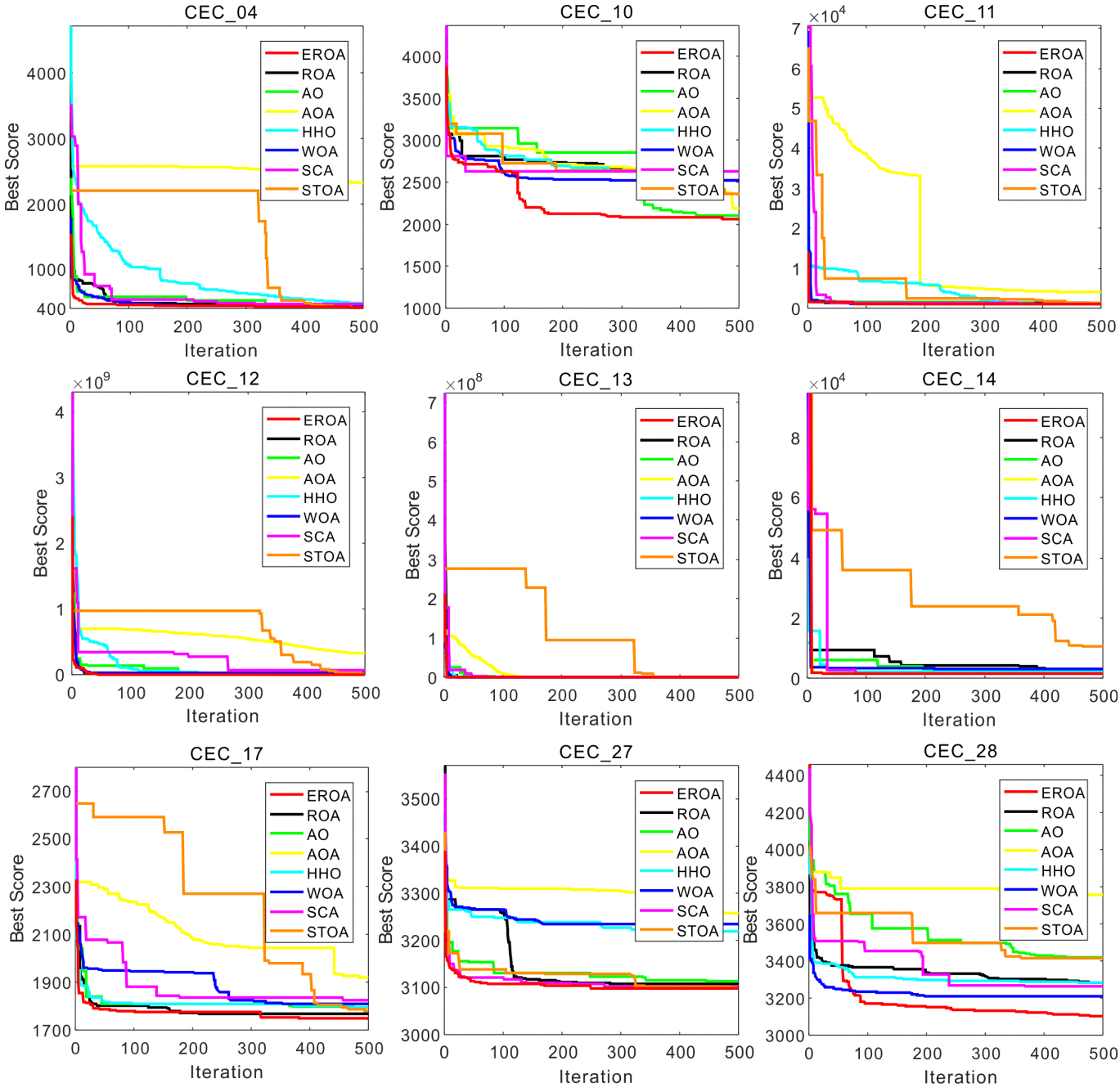

Figure 6. Convergence curves for the optimization algorithms for some test functions on CEC 2017.

Table 7. Results of algorithms on the CEC2017 test suite.

F		EROA	ROA	AO	AOA	HHO	WOA	SCA	STOA
CEC_01	Avg	1.21×10^{8}	1.11×10^{9}	1.26×10^{7}	1.48×10^{10}	1.12×10^{6}	9.98×10^{7}	1.13×10^{9}	2.44×10^{8}
	Std	1.66×10^{8}	1.18×10^{9}	1.20×10^{7}	4.69×10^{9}	1.10×10^{6}	2.92×10^{8}	3.89×10^{8}	1.98×10^{8}
CEC_03	Avg	2.39×10^{3}	4.84×10^{3}	2.56×10^{3}	1.55×10^{4}	7.29×10^{2}	9.65×10^{3}	3.36×10^{3}	2.23×10^{3}
	Std	1.66×10^{3}	3.37×10^{3}	9.78×10^{2}	1.61×10^{3}	3.31×10^{2}	8.33×10^{3}	2.20×10^{3}	1.78×10^{3}
CEC_04	Avg	4.26×10^{2}	4.93×10^{2}	4.30×10^{2}	2.06×10^{3}	4.31×10^{2}	4.52×10^{2}	4.64×10^{2}	4.40×10^{2}
	Std	2.90×10^{1}	1.07×10^{2}	2.73×10^{1}	9.69×10^{2}	3.95×10^{1}	5.08×10^{1}	3.13×10^{1}	2.61×10^{1}
CEC_055	Avg	5.48×10^{2}	5.58×10^{2}	5.36×10^{2}	5.78×10^{2}	5.53×10^{2}	5.63×10^{2}	5.55×10^{2}	5.30×10^{2}
	Std	1.26×10^{1}	2.02×10^{1}	1.11×10^{1}	2.42×10^{1}	1.78×10^{1}	2.42×10^{1}	8.27	8.79
CEC_06	Avg	6.31×10^{2}	6.33×10^{2}	6.21×10^{2}	6.44×10^{2}	6.39×10^{2}	6.38×10^{2}	6.24×10^{2}	6.15×10^{2}
	Std	1.25×10^{1}	1.49×10^{1}	8.32	7.20	9.63	1.49×10^{1}	5.37	5.60
CEC_07	Avg	7.86×10^{2}	7.96×10^{2}	7.58×10^{2}	8.01×10^{2}	7.94×10^{2}	7.96×10^{2}	7.82×10^{2}	7.63×10^{2}
	Std	1.97×10^{1}	2.58×10^{1}	1.43×10^{1}	8.88	1.76×10^{1}	2.43×10^{1}	1.08×10^{1}	1.82×10^{1}
CEC_08	Avg	8.33×10^{2}	8.37×10^{2}	8.26×10^{2}	8.48×10^{2}	8.32×10^{2}	8.42×10^{2}	8.46×10^{2}	8.29×10^{2}
	Std	7.58	8.51×10^{1}	8.30	9.88	7.71	1.59×10^{1}	9.26	1.15×10^{1}
CEC_09	Avg	1.50×10^{3}	1.31×10^{3}	1.05×10^{3}	1.53×10^{3}	1.50×10^{3}	1.48×10^{3}	1.06×10^{3}	1.08×10^{3}
	Std	2.62×10^{2}	2.14×10^{2}	7.39×10^{1}	1.74×10^{2}	2.29×10^{2}	5.67×10^{2}	8.04×10^{1}	1.42×10^{2}
CEC_10	Avg	1.97×10^{3}	2.12×10^{3}	2.06×10^{3}	2.25×10^{3}	2.14×10^{3}	2.32×10^{3}	2.49×10^{3}	2.00×10^{3}
	Std	2.02×10^{2}	3.28×10^{2}	3.32×10^{2}	2.52×10^{2}	3.12×10^{2}	3.27×10^{2}	1.83×10^{2}	2.81×10^{2}
CEC_11	Avg	1.18×10^{3}	1.22×10^{3}	1.29×10^{3}	4.80×10^{3}	1.18×10^{3}	1.25×10^{3}	1.25×10^{3}	1.23×10^{3}
	Std	5.32×10^{1}	8.74×10^{1}	1.12×10^{2}	2.35×10^{3}	6.22×10^{1}	9.25×10^{1}	8.26×10^{1}	1.12×10^{2}
CEC_12	Avg	2.56×10^{6}	4.25×10^{6}	3.40×10^{6}	1.05×10^{9}	4.95×10^{6}	5.48×10^{6}	2.15×10^{7}	3.21×10^{6}
	Std	3.11×10^{6}	5.18×10^{6}	3.54×10^{6}	6.67×10^{8}	5.20×10^{6}	6.12×10^{6}	2.24×10^{7}	3.96×10^{6}
CEC_13	Avg	8.62×10^{3}	1.37×10^{4}	1.63×10^{4}	1.14×10^{6}	1.78×10^{4}	2.29×10^{4}	8.54×10^{4}	2.26×10^{4}
	Std	8.03×10^{3}	9.87×10^{3}	1.03×10^{4}	3.60×10^{6}	1.36×10^{4}	1.66×10^{4}	6.46×10^{4}	1.73×10^{4}
CEC_14	Avg	1.59×10^{3}	2.78×10^{3}	2.71×10^{3}	1.05×10^{4}	2.38×10^{3}	3.23×10^{3}	2.03×10^{3}	3.77×10^{3}
	Std	6.91×10^{2}	1.65×10^{3}	1.64×10^{3}	8.12×10^{3}	1.43×10^{3}	1.85×10^{3}	7.89×10^{2}	2.62×10^{3}
CEC_15	Avg	8.36×10^{3}	7.53×10^{3}	6.43×10^{3}	2.09×10^{4}	7.84×10^{3}	8.99×10^{3}	4.75×10^{3}	6.69×10^{3}
	Std	3.50×10^{3}	5.57×10^{3}	3.06×10^{3}	4.97×10^{3}	3.40×10^{3}	8.37×10^{3}	3.96×10^{3}	4.81×10^{3}
CEC_16	Avg	1.90×10^{3}	1.93×10^{3}	1.89×10^{3}	2.11×10^{3}	1.92×10^{3}	1.94×10^{3}	1.80×10^{3}	1.76×10^{3}
	Std	1.62×10^{2}	2.01×10^{2}	1.49×10^{2}	1.85×10^{2}	1.29×10^{2}	1.50×10^{2}	9.24×10^{1}	1.11×10^{2}
CEC_17	Avg	1.78×10^{3}	1.80×10^{3}	1.78×10^{3}	1.89×10^{3}	1.79×10^{3}	1.82×10^{3}	1.80×10^{3}	1.80×10^{3}
	Std	3.95×10^{1}	3.57×10^{1}	2.98×10^{1}	1.10×10^{2}	5.54×10^{1}	7.80×10^{1}	2.80×10^{1}	5.07×10^{1}
CEC_18	Avg	1.75×10^{4}	2.12×10^{4}	3.67×10^{4}	2.16×10^{8}	1.66×10^{4}	1.53×10^{4}	2.59×10^{5}	4.90×10^{4}
	Std	1.05×10^{4}	1.61×10^{4}	2.92×10^{4}	4.03×10^{8}	1.02×10^{4}	1.13×10^{4}	1.88×10^{5}	2.40×10^{4}
CEC_19	Avg	1.36×10^{4}	2.24×10^{4}	1.91×10^{4}	1.31×10^{5}	2.01×10^{4}	8.22×10^{4}	8.37×10^{3}	1.31×10^{4}
	Std	1.13×10^{4}	5.15×10^{4}	2.21×10^{4}	6.74×10^{4}	2.31×10^{4}	1.28×10^{5}	7.13×10^{3}	1.11×10^{4}
CEC_20	Avg	2.18×10^{3}	2.18×10^{3}	2.13×10^{3}	2.15×10^{3}	2.20×10^{3}	2.22×10^{3}	2.12×10^{3}	2.15×10^{3}
	Std	1.01×10^{2}	6.85×10^{1}	6.48×10^{1}	5.06×10^{1}	6.95×10^{1}	9.70×10^{1}	4.59×10^{1}	6.34×10^{1}
CEC_21	Avg	2.30×10^{3}	2.28×10^{3}	2.31×10^{3}	2.37×10^{3}	2.33×10^{3}	2.33×10^{3}	2.27×10^{3}	2.21×10^{3}
	Std	7.17×10^{1}	5.77×10^{1}	4.48×10^{1}	2.96×10^{1}	6.10×10^{1}	5.87×10^{1}	6.76×10^{1}	2.27×10^{1}
CEC_22	Avg	2.32×10^{3}	2.41×10^{3}	2.31×10^{3}	3.44×10^{3}	2.37×10^{3}	2.40×10^{3}	2.40×10^{3}	2.93×10^{3}
	Std	2.36×10^{1}	1.07×10^{2}	7.90	2.64×10^{2}	2.93×10^{2}	3.15×10^{2}	2.93×10^{1}	6.84×10^{2}
CEC_23	Avg	2.68×10^{3}	2.65×10^{3}	2.65×10^{3}	2.79×10^{3}	2.69×10^{3}	2.66×10^{3}	2.66×10^{3}	2.64×10^{3}
	Std	3.07×10^{1}	2.79×10^{1}	1.70×10^{1}	5.39×10^{1}	3.72×10^{1}	3.04×10^{1}	7.94×10^{1}	1.05×10^{1}
CEC_24	Avg	2.78×10^{3}	2.78×10^{3}	2.76×10^{3}	2.90×10^{3}	2.83×10^{3}	2.80×10^{3}	2.79×10^{3}	2.76×10^{3}
	Std	9.37×10^{1}	6.31×10^{1}	5.05×10^{1}	9.77×10^{1}	7.61×10^{1}	4.91×10^{1}	1.25×10^{1}	1.45×10^{1}
CEC_25	Avg	2.93×10^{3}	3.00×10^{3}	2.94×10^{3}	3.57×10^{3}	2.93×10^{3}	2.96×10^{3}	2.98×10^{3}	2.94×10^{3}
	Std	3.17×10^{1}	9.84×10^{1}	2.22×10^{1}	2.99×10^{2}	2.27×10^{1}	4.02×10^{1}	2.33×10^{1}	2.28×10^{1}
CEC_26	Avg	3.65×10^{3}	3.37×10^{3}	3.05×10^{3}	4.42×10^{3}	3.63×10^{3}	3.64×10^{3}	3.20×10^{3}	3.28×10^{3}
	Std	5.27×10^{2}	3.13×10^{2}	2.23×10^{2}	3.38×10^{2}	2.94×10^{2}	5.69×10^{2}	3.04×10^{2}	4.47×10^{2}
CEC_27	Avg	3.13×10^{3}	3.13×10^{3}	3.11×10^{3}	3.30×10^{3}	3.16×10^{3}	3.15×10^{3}	3.11×10^{3}	3.10×10^{3}
	Std	4.05×10^{1}	4.49×10^{1}	8.58	1.05×10^{2}	4.84×10^{1}	3.85×10^{1}	2.91	2.60
CEC_28	Avg	3.32×10^{3}	3.35×10^{3}	3.44×10^{3}	3.95×10^{3}	3.42×10^{3}	3.46×10^{3}	3.34×10^{3}	3.36×10^{3}
	Std	1.05×10^{2}	1.29×10^{2}	9.77×10^{1}	1.89×10^{2}	1.69×10^{2}	1.57×10^{2}	8.82×10^{1}	1.18×10^{2}
CEC_29	Avg	3.35×10^{3}	3.29×10^{3}	3.25×10^{3}	3.53×10^{3}	3.36×10^{3}	3.39×10^{3}	3.26×10^{3}	3.23×10^{3}
	Std	1.07×10^{2}	7.85×10^{1}	5.58×10^{1}	2.03×10^{2}	1.02×10^{2}	8.82×10^{1}	5.57×10^{1}	5.42×10^{1}
CEC_30	Avg	1.34×10^{6}	1.65×10^{6}	1.26×10^{6}	7.29×10^{7}	3.15×10^{6}	1.58×10^{6}	2.03×10^{6}	4.65×10^{5}
	Std	1.54×10^{6}	1.75×10^{6}	1.33×10^{6}	6.56×10^{7}	3.72×10^{6}	1.82×10^{6}	1.25×10^{6}	2.99×10^{5}

Table 8. Friedman rank test results on CEC2017 test suite.

F	EROA	ROA	AO	AOA	HHO	WOA	SCA	STOA
CEC_01	4	6	2	8	1	3	7	4
CEC_03	3	6	4	8	1	7	5	3
CEC_04	1	7	2	8	3	5	6	1
CEC_05	3	6	2	8	4	7	5	3
CEC_06	4	5	2	8	7	6	3	4
CEC_07	4	6	1	8	5	6	3	4
CEC_08	4	5	1	8	3	6	7	4
CEC_09	6	4	1	8	6	5	2	6
CEC_10	1	4	3	6	5	7	8	1
CEC_11	1	3	7	8	1	5	5	1
CEC_12	1	4	3	8	5	6	7	1
CEC_13	1	2	3	8	4	6	7	1
CEC_14	1	5	4	8	3	6	2	1
CEC_15	2	4	6	8	5	7	1	2
CEC_16	4	6	3	8	5	7	2	4
CEC_17	1	4	1	8	3	7	4	1
CEC_18	3	4	5	8	2	1	7	3
CEC_19	3	6	4	8	5	7	1	3
CEC_20	2	5	5	3	7	8	1	2
CEC_21	4	3	5	8	6	6	2	4
CEC_22	2	6	1	8	3	4	4	2
CEC_23	2	2	6	8	7	4	4	2
CEC_24	3	3	1	8	7	6	5	3
CEC_25	1	7	3	8	1	5	6	1
CEC_26	7	4	1	8	5	6	2	7
CEC_27	4	4	2	8	7	6	2	4
CEC_28	1	3	6	8	5	7	2	1
CEC_29	5	4	2	8	6	7	3	5
CEC_30	3	5	2	8	7	4	6	3
Avg Rank	2.7931	4.5862	3.0344	7.7586	4.4482	5.7586	4.1034	2.8275
Final Rank	1	6	3	8	5	7	4	2

5. Constrained Engineering Design Problems

The previous experiments show the EROA's ability to solve numerical optimization problems. Here, to be able to reveal the powerfulness of EROA in solving real constrained engineering problems, seven different problems are used; namely, pressure vessel design, speed reducer design, tension/compression spring design, three-bar truss, welded beam design, tubular column design, and gear train design problem.

5.1. Pressure Vessel Design Problem

Pressure vessel design is a minimization problem that consists of 4 variables, as shown in Figure 7. The mathematical formulation of this problem is shown below.

Consider

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ \mathrm{R}\ L], \tag{18}$$

Minimize

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \tag{19}$$

Subject to

$$\begin{aligned} g_1(\vec{x}) &= -x_1 + 0.0193x_3 \le 0, \\ g_2(\vec{x}) &= -x_3 + 0.00954x_3 \le 0, \\ g_3(\vec{x}) &= -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \le 0, \\ g_4(\vec{x}) &= x_4 - 240 \le 0, \end{aligned} \tag{20}$$

Variable Range

$$\begin{aligned} &0 \leq x_1 \leq 99, \\ &0 \leq x_2 \leq 99, \\ &10 \leq x_3 \leq 200, \\ &10 \leq x_4 \leq 200, \end{aligned} \tag{21}$$

Results of EROA in solving pressure vessel design are compared with Aquila Optimizer (AO), Harris Hawks Optimization (HHO), Whale Optimization Algorithm (WOA), Slime Mould Algorithm (SMA) [61], Grey Wolf Optimizer (GWO), Multi-Verse Optimizer (MVO), Evolutions Strategy (ES) [62], Gravitational Search Algorithm (GSA), Genetic Algorithm (GA), and Co-evolutionary Particle Swarm Optimization (CPSO) [63] as shown in Table 9. It can be seen from this table, EROA has the smallest cost, 5935,7301 with X = (0.8434295, 0.4007618, 44.786, 145.9578).

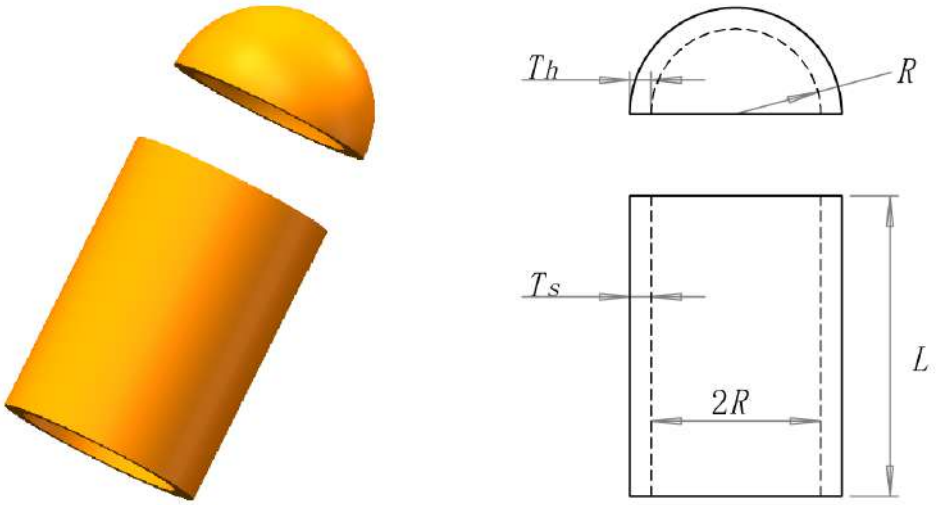

Figure 7. The pressure vessel design problem is the 3D model diagram (**left**) and structural parameters (**right**).

Table 9. Optimization results for the pressure vessel design problem.

Algorithm	Optimum Variables				Best Cost
	T_s	T_h	R	L	
EROA	0.8434295	0.4007618	44.786	145.9578	5935.7301
AO [34]	1.0540	0.182806	59.6219	38.8050	5949.2258
HHO [25]	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
WOA [21]	0.8125	0.4375	42.0982699	176.638998	6059.7410
SMA [61]	0.7931	0.3932	40.6711	196.2178	5994.1857
GWO [19]	0.8125	0.4345	42.0892	176.7587	6051.5639
MVO [45]	0.8125	0.4375	42.090738	176.73869	6060.8066
ES [62]	0.8125	0.4375	42.098087	176.640518	6059.74560
GSA [38]	1.125000	0.625000	55.9886598	84.4542025	8538.8359
GA [47]	0.8125	0.4375	42.097398	176.65405	6059.94634
CPSO [63]	0.8125	0.4375	42.091266	176.7465	6061.0777

5.2. Speed Reducer Design Problem

The second engineering problem is the speed reducer design, which minimizes the reducer weight. It has seven variables, as shown in Figure 8. The mathematical formulation is given below.

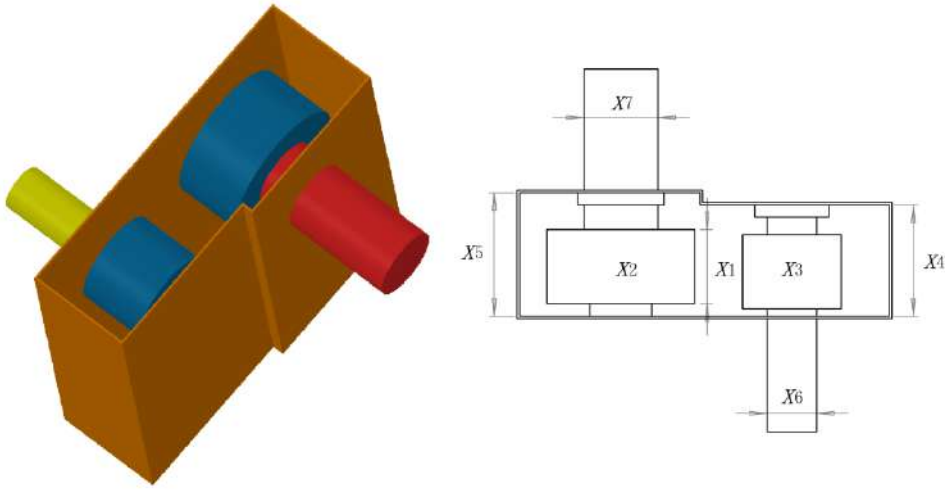

Figure 8. The speed reducer design problem is the 3D model diagram (**left**) and structural parameters (**right**).

Minimize

$$f(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3), \quad (22)$$

Subject to

$$\begin{aligned} g_1(\vec{x}) &= \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \\ g_2(\vec{x}) &= \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, \\ g_3(\vec{x}) &= \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0, \\ g_4(\vec{x}) &= \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0, \\ g_5(\vec{x}) &= \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6}}{110.0x_6^3} - 1 \leq 0, \\ g_6(\vec{x}) &= \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5\times10^6}}{85.0x_6^3} - 1 \leq 0, \\ g_7(\vec{x}) &= \frac{x_2x_3}{40} - 1 \leq 0, \\ g_8(\vec{x}) &= \frac{5x_2}{x_1} - 1 \leq 0, \\ g_9(\vec{x}) &= \frac{x_1}{12x_2} - 1 \leq 0, \\ g_{10}(\vec{x}) &= \frac{1.5x_6+1.9}{x_4} - 1 \leq 0, \\ g_{11}(\vec{x}) &= \frac{1.1x_7+1.9}{x_5} - 1 \leq 0, \end{aligned} \quad (23)$$

Variable Range

$$\begin{aligned} &2.6 \leq x_1 \leq 3.6, \\ &0.7 \leq x_2 \leq 0.8, \\ &17 \leq x_3 \leq 28, \\ &7.3 \leq x_4 \leq 8.3, \\ &7.8 \leq x_5 \leq 8.3, \\ &2.9 \leq x_6 \leq 3.9, \\ &5.0 \leq x_7 \leq 5.5, \end{aligned} \quad (24)$$

Table 10 shows the results of EROA compared with Aquila Optimizer (AO), Arithmetic Optimization Algorithm (AOA), Sine Cosine Algorithm (SCA), Particle Swarm Optimization (PSO), Moth-Flame Optimization (MFO), Harmony Search (HS) [64], and Decomposition Algorithm (MDA) [65]. It can be noticed that EROA ranked first with 2998.9886 where X = (3.49692, 0.7, 17, 7.66313, 7.8, 3.3505, 5.28582).

Table 10. Optimization results for the the speed reducer design problem.

Algorithm	Optimum Variables							Best Weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
EROA	3.49692	0.7	17	7.66313	7.8	3.3505	5.28582	2998.9886
AO [34]	3.5021	0.7	17	7.3099	7.7476	3.3641	5.2994	3007.7328
AOA [44]	3.50384	0.7	17	7.3	7.72933	3.35649	5.2867	2997.9157
SCA [41]	3.50875	0.7	17	7.3	7.8	3.46102	5.28921	3030.563
PSO [17]	3.5001	0.7	17.0002	7.5177	7.7832	3.3508	5.2867	3145.922
MFO [27]	3.49745	0.7	17	7.82775	7.71245	3.35178	5.28635	2998.9408
GA [47]	3.51025	0.7	17	8.35	7.8	3.36220	5.28772	3067.561
HS [64]	3.52012	0.7	17	8.37	7.8	3.36697	5.28871	3029.002
MDA [65]	3.5	0.7	17	7.3	7.67039	3.54242	5.24581	3019.58336

5.3. Tension/Compression Spring Design Problem

Tension/compression spring design problem aims to determine the minimum cost of spring fabrication. It has three variables as shown in Figure 9. The mathematical model can be shown below.

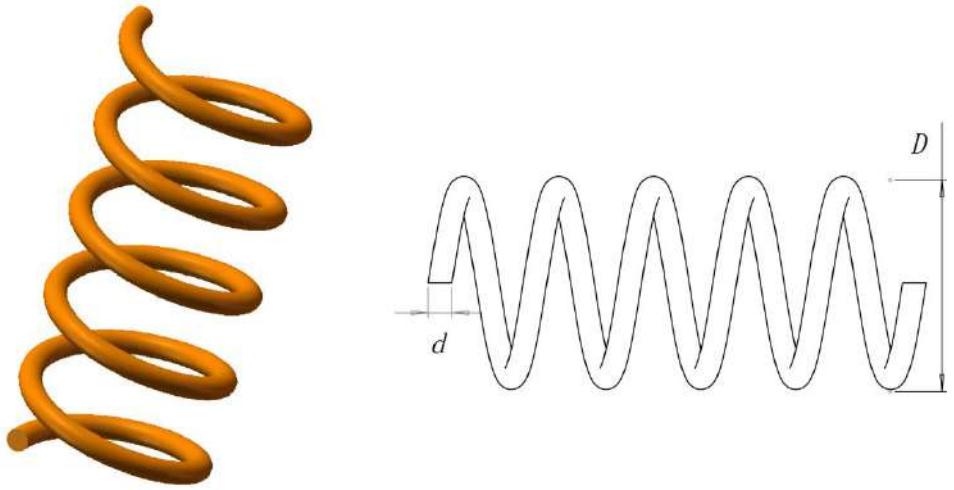

Figure 9. The tension/compression spring design problem: 3D model diagram (**left**) and structural parameters (**right**).

Consider

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [d\ D\ N], \tag{25}$$

Minimize

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2, \tag{26}$$

Subject to

$$\begin{aligned} g_1(\vec{x}) &= 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0, \\ g_2(\vec{x}) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0, \\ g_3(\vec{x}) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\ g_4(\vec{x}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0, \end{aligned} \tag{27}$$

Variable Range

$$\begin{aligned} 0.05 \leq x_1 \leq 2.00, \\ 0.25 \leq x_2 \leq 1.30, \\ 2.00 \leq x_3 \leq 15.00, \end{aligned} \tag{28}$$

Results of tension/compression spring design is given in Table 11, where the EROA is compared with Aquila Optimizer (AO), Harris Hawks Optimization (HHO), Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA), Grey Wolf Optimizer (GWO), Multi-Verse Optimization (MVO), Particle Swarm Optimization (PSO), Improved Teaching-Learning Based Optimization algorithm (RLTLBO) [66], Genetic Algorithm (GA), and Harmony Search (HS). From this table, we can seen that, EROA has achieved the best results.

Table 11. Optimization results for the tension/compression design problem.

Algorithm	Optimum Variables			Best Weight
	d	***D***	***N***	
EROA	0.053799	0.46951	5.811	0.010614
AO [34]	0.0502439	0.35262	10.5425	0.011165
HHO [25]	0.051796393	0.359305355	11.138859	0.012665443
WOA [21]	0.051207	0.345215	12.004032	0.0126763
SSA [23]	0.051207	0.345215	12.004032	0.0126763
GWO [19]	0.05169	0.356737	11.28885	0.012666
MVO [45]	0.05251	0.37602	10.33513	0.012790
PSO [17]	0.051728	0.357644	11.244543	0.0126747
RLTLBO [60]	0.055118	0.5059	5.1167	0.010938
GA [47]	0.051480	0.351661	11.632201	0.01270478
HS [64]	0.051154	0.349871	12.076432	0.0126706

5.4. Three-Bar Truss Design Problem

The three-bar design problem aims to find the minimum structure burden. It has two variables, as shown in Figure 10. The mathematical model is given below.

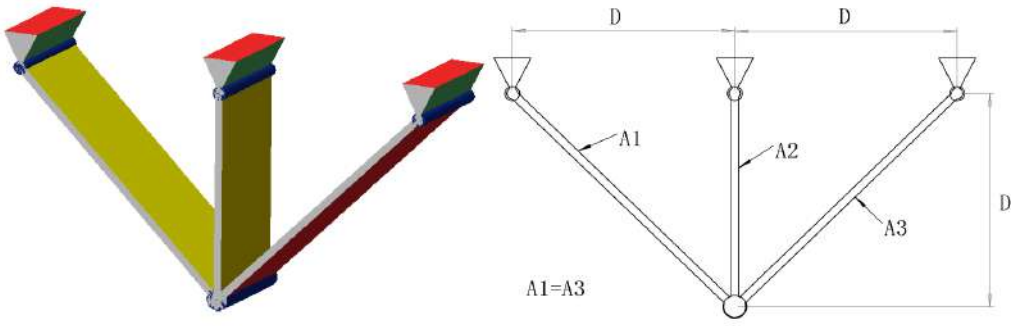

Figure 10. The three-bar truss design problem is the 3D model diagram (**left**) and structural parameters (**right**).

Consider

$$\vec{x} = [x_1\ x_2] = [A_1\ A_2], \tag{29}$$

Minimize

$$f(\vec{x}) = (2\sqrt{2}x_1 + x_2) * l, \tag{30}$$

Subject to

$$\begin{aligned} g_1(\vec{x}) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0, \\ g_2(\vec{x}) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0, \\ g_3(\vec{x}) &= \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0, \end{aligned} \tag{31}$$

Variable Range

$$0 \leq x_1, x_2 \leq 1, \tag{32}$$

where $l = 100$ cm, $P = 2$ KN/cm^2, $\sigma = 2$ KN/cm^2.

Results of the three-bar truss are given in Table 12, in which EROA has been compared with Sailfish Optimizer (SFO), Aquila Optimizer (AO), Arithmetic Optimization Algorithm (AOA), Harris Hawks Optimizer (HHO), Salp Swarm Algorithm (SSA), Ant Lion Optimizer (ALO), Multi-Verse Optimizer (MVO), Moth-Flame Optimization (MFO), Grasshopper Optimization Algorithm (GOA) [67], and Improved Hybrid Aquila Optimizer and Harris Hawks Optimization (IHAOHHO) [68]. It's obvious that EROA has ranked first with fitness value equal to 263.8552.

Table 12. Optimization results for the the three-bar truss design problem.

Algorithm	Optimum Variables		Best Weight
	x_1	x_2	
EROA	0.78645	0.41369	263.8552
SFO [57]	0.7884562	0.40886831	263.8959212
AO [34]	0.7926	0.3966	263.8684
AOA [44]	0.79369	0.39426	263.9154
HHO [25]	0.788662816	0.408283133832900	263.8958434
SSA [23]	0.78866541	0.408275784	263.89584
ALO [20]	0.7886618	0.4082831	263.8958434
MVO [45]	0.78860276	0.408453070000000	263.8958499
MFO [27]	0.788244771	0.409466905784741	263.8959797
GOA [67]	0.788897555578973	0.407619570115153	263.895881496069
IHAOHHO [68]	0.79002	0.40324	263.8622

5.5. Welded Beam Design Problem

The fifth engineering problem is welded beam design which tries to find the minimum price of welded beam manufacturing. It has four variables, as shown in Figure 11. The mathematical model of welded beam design is given below.

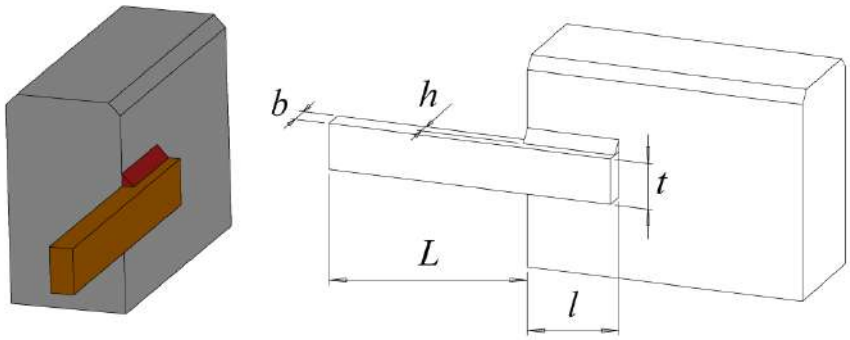

Figure 11. The welded beam design problem is the 3D model diagram (**left**) and structural parameters (**right**).

Consider

$$\vec{x} = [x_1,\ x_2,\ x_3,\ x_4] = [h,\ l,\ t,\ b] \tag{33}$$

Minimize

$$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{34}$$

Subject to

$$\begin{aligned}
&g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0,\\
&g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leq 0,\\
&g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0,\\
&g_4(\vec{x}) = x_1 - x_4 \leq 0,\\
&g_5(\vec{x}) = P - P_C(\vec{x}) \leq 0,\\
&g_6(\vec{x}) = 0.125 - x_1 \leq 0,\\
&g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \leq 0,
\end{aligned} \tag{35}$$

where

$$\begin{aligned}
&\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\\
&\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ M = P(L + \frac{x_2}{2}),\\
&R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2},\\
&J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2\right]\right\},\\
&\sigma(\vec{x}) = \frac{6PL}{x_3^2x_4},\ \delta(\vec{x}) = \frac{6PL^3}{Ex_3^3x_4},\\
&P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),
\end{aligned} \tag{36}$$

Variable Range

$$\begin{aligned}
&P = 6000 \text{ lb}, L = 14 \text{ in}, E = 30 \times 10^6 \text{ psi}, G = 12 \times 10^6 \text{ psi}\\
&\tau_{\max} = 13600 \text{ psi}, \sigma_{\max} = 30000 \text{ psi}, \delta_{\max} = 0.25 \text{ in},
\end{aligned} \tag{37}$$

Results of welded beam design are given in Table 13, where the EROA is compared with the original ROA, Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), Moth-Flame Optimization (MFO), Multi-Verse Optimization (MVO), Coevolutionary Particle Swarm Optimization (CPSO), Ray Optimization (RO), Improved Wild Horse Optimization (IWHO) [69], Improved HHO (IHHO), and Harmony Search (HS). From this table, we can conclude that EROA has ranked first in comparison with 11 algorithms.

Table 13. Optimization results for the welded beam design problem.

Algorithm	Optimum Variables				Best Weight
	h	*l*	*t*	*b*	
EROA	0.20352	3.3013	9.0091	0.20735	1.7059
ROA [24]	0.200077	3.365754	9.011182	0.206893	1.706447
WOA [21]	0.205396	3.484293	9.037426	0.206276	1.730499
GWO [19]	0.205676	3.478377	9.03681	0.205778	1.72624
MFO [27]	0.2057	3.4703	9.0364	0.2057	1.72452
MVO [45]	0.205463	3.473193	9.044502	0.205695	1.72645
CPSO [63]	0.202369	3.544214	9.048210	0.205723	1.73148
RO [42]	0.203687	3.528467	9.004233	0.207241	1.735344
GA [47]	0.1829	4.0483	9.3666	0.2059	1.82420
IWHO [69]	0.2057	3.2530	9.0366	0.2057	1.6952
IHHO [7]	0.20533	3.47226	9.0364	0.2010	1.7238
HS [64]	0.2442	6.2231	8.2915	0.2443	2.3807

5.6. Tubular Column Design Problem

The sixth engineering problem is called tubular column design [65], which aims to find the lowest cost to design tubular columns. It consists of two variables, whereas the mathematical description of this problem is shown below.

Minimize

$$f(d,\ t) = 9.8dt + 2d, \tag{38}$$

Subject to

$$\begin{aligned} g_1 &= \frac{P}{\pi dt\sigma_y} - 1 \leq 0, \\ g_2 &= \frac{8PL^2}{\pi^3 Edt(d^2+t^2)} - 1 \leq 0, \\ g_3 &= \frac{2.0}{d} - 1 \leq 0, \\ g_4 &= \frac{d}{14} - 1 \leq 0, \\ g_5 &= \frac{0.2}{t} - 1 \leq 0, \\ g_6 &= \frac{t}{0.8} - 1 \leq 0, \end{aligned} \tag{39}$$

Variable Range

$$0.01 \leq d,\ t \leq 100, \tag{40}$$

Results of the tubular column are given in Table 14, where EROA is compared with the classical ROA, Aquila Optimizer (AO), Harris Hawks Optimization (HHO), Whale Optimization Algorithm (WOA), Marine Predators Algorithm (MPA), Cuckoo Search (CS) [70], Modified Ant Lion Optimizer (MALO) [71], and Random Opposition-based Learning Grey Wolf Optimizer (ROLGWO). EROA has find the nearest to optimal value which equals to 26.5316.

Table 14. Optimization results for the tubular column design problem.

Algorithm	Optimal Values for Variables		Best Cost
	d	***t***	
EROA	5.4511	0.29198	26.5316
ROA [24]	5.433671	0.294813	26.598146
AO [34]	5.46300	0.29656	26.83540
HHO [25]	5.44380	0.29313	26.55820
WOA [21]	5.437032	0.294228	26.583393
MPA [22]	5.451389	0.291951	26.531737
CS [70]	5.45139	0.29196	26.53217
MALO [71]	5.451140	0.291967	26.531342
ROLGWO [59]	5.452650	0.291894	26.534764

5.7. Gear Train Design Problem

The last engineering problem discussed here is Gear train design [14] which aims to minimize gear ratio. It consists of four variables whereas the mathematical formulation is founded below.

Consider

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [n_A\ n_B\ n_C\ n_D], \tag{41}$$

Minimize

$$f(\vec{x}) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4}\right)^2, \tag{42}$$

Variable Range

$$12 \leq x_1, x_2, x_3, x_4 \leq 60, \tag{43}$$

Results of gear train is compared with Multi-Verse Optimization (MVO), Moth-Flame Optimization (MFO), Ant Lion Optimizer (ALO), Genetic Algorithm (GA), Cuckoo Search (CS), Artificial Bee Colony (ABC) [72], and Mine Blast Algorithm (MBA) [73] in Table 15.

Table 15. Optimization results for the gear train design problem.

Algorithm	Optimum Variables				Best Gear Ratio
	n_A	n_B	n_C	n_D	
EROA	49	19	16	43	2.7009×10^{-12}
MVO [45]	43	19	16	49	2.7009×10^{-12}
MFO [27]	43	19	16	49	2.7009×10^{-12}
ALO [20]	49	19	16	43	2.7009×10^{-12}
GA [47]	49	19	16	43	2.7009×10^{-12}
CS [70]	43	19	16	49	2.7009×10^{-12}
ABC [72]	49	19	16	43	2.7009×10^{-12}
MBA [73]	43	19	16	49	2.7009×10^{-12}

6. Conclusions

ROA is a recent swarm-based algorithm that simulates the intelligent traveler Remora behavior. In this study, a novel version of ROA, called Enhanced ROA (EROA), is proposed, which is based on three strategies; namely, (1) adaptive dynamic probability, (2) SFO with Levy flight, and (3) restart strategy. Two different datasets have been used to evaluate the performance of the suggested algorithm (CEC2005 & CEC2017) with 52 functions and different seven constrained real-world engineering problems. Moreover, Wilcoxon rank sum and Friedman tests have been used to prove the powerful of EROA. The statistical analysis and experimental results show the efficiency of EROA. In future, we can propose this version as a binary version and apply it to feature selection, classification, or unit commitment problem. We also can propose a multi-objective version of it to solve the more complex multiobjective problems.

Author Contributions: Conceptualization, S.W.; methodology, S.W. and H.J.; software, S.W.; validation, S.W., A.G.H. and R.Z.; formal analysis, L.A.; investigation, A.G.H.; resources, S.W.; data curation, S.W.; writing—original draft preparation, S.W. and A.G.H.; writing—review and editing, L.A. and A.G.H.; visualization, S.W.; supervision, H.J.; project administration, S.W.; funding acquisition, S.W. and H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Sanming University Introduces High-level Talents to Start Scientific Research Funding Support Project, grant number 20YG01, 20YG14; the Guiding Science and Technology Projects in Sanming City, grant number 2020-S-39; the Educational Research Projects of Young and Middle-aged Teachers in Fujian Province, grant number JAT200638; the Scientific Research and Development Fund of Sanming University, grant number B202029; School level education and teaching reform project of Sanming University, grant number J2010306; Higher education research project of Sanming University, grant number SHE2102; 2021 project of the 14th Five-year Plan of Education science in Fujian Province, grant number FJJKBK21-138; Fujian Natural Science Foundation Project, grant number 2021J011128; and Sanming University National Natural Science Foundation Breeding Project, grant number PYT2105.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the support of Fujian Key Lab of Agriculture IOT Application and IOT Application Engineering Research Center of Fujian Province Colleges and Universities, as well as the anonymous reviewers to help us improve the quality of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, X.S. *Engineering Optimization: An Introduction with Metaheuristic Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2010. [CrossRef]
2. Abualigah, L.; Gandomi, A.H.; Elaziz, M.A.; Hussien, A.G.; Khasawneh, A.M.; Alshinwan, M.; Houssein, E.H. Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis. *Algorithms* **2020**, *13*, 345. [CrossRef]
3. Hassanien, A.E.; Emary, E. *Swarm Intelligence: Principles, Advances, and Applications*; CRC Press: Boca Raton, FL, USA, 2018.
4. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Bhattacharyya, S.; Amin, M. S-shaped binary whale optimization algorithm for feature selection. *Recent Trends Signal Image Process.* **2019**, *727*, 79–87.
5. Fathi, H.; AlSalman, H.; Gumaei, A.; Manhrawy, I.I.; Hussien, A.G.; El-Kafrawy, P. An efficient cancer classification model using microarray and high-dimensional data. *Comput. Intell. Neurosci.* **2021**, *2021*, 7231126. [CrossRef] [PubMed]
6. Hussien, A.G.; Houssein, E.H.; Hassanien, A.E. A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection. In Proceedings of the 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; pp. 166–172. [CrossRef]
7. Hussien, A.G.; Amin, M. A self-adaptive harris hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 309–336. [CrossRef]
8. Abdullahi, M.; Ngadi, M.A.; Dishing, S.I.; Ahmad, B.I. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment. *J. Netw. Comput. Appl.* **2019**, *133*, 60–74. [CrossRef]
9. Besnassi, M.; Neggaz, N.; Benyettou, A. Face detection based on evolutionary Haar filter. *Pattern Anal. Appl.* **2020**, *23*, 309–330. [CrossRef]
10. Neshat, M.; Mirjalili, S.; Sergiienko, N.Y.; Esmaeilzadeh, S.; Amini, E.; Heydari, A.; Garcia, D.A. Layout optimisation of offshore wave energy converters using a novel multi-swarm cooperative algorithm with backtracking strategy: A case study from coasts of Australia. *Energy* **2022**, *239*, 122463. [CrossRef]
11. Eslami, M.; Neshat, M.; Khalid, S.A. A Novel Hybrid Sine Cosine Algorithm and Pattern Search for Optimal Coordination of Power System Damping Controllers. *Sustainability* **2022**, *14*, 541. [CrossRef]
12. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Zamani, H.; Bahreininejad, A. GGWO: Gaze cues learning-based grey wolf optimizer and its applications for solving engineering problems. *J. Comput. Sci.* **2022**, *61*, 101636. [CrossRef]
13. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
14. Hussien, A.G.; Oliva, D.; Houssein, E.H.; Juan, A.A.; Yu, X. Binary whale optimization algorithm for dimensionality reduction. *Mathematics* **2020**, *8*, 1821. [CrossRef]
15. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H. Swarming behaviour of salps algorithm for predicting chemical compound activities. In Proceedings of the 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; pp. 315–320.
16. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Amin, M.; Azar, A.T. New binary whale optimization algorithm for discrete optimization problems. *Eng. Optim.* **2020**, *52*, 945–959. [CrossRef]
17. Fearn, T. Particle swarm optimisation. *NIR News* **2014**, *25*, 27. [CrossRef]
18. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1996**, *26*, 29–41. [CrossRef]
19. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
20. Assiri, A.S.; Hussien, A.G.; Amin, M. Ant lion optimization: Variants, hybrids, and applications. *IEEE Access* **2020**, *8*, 77746–77764. [CrossRef]
21. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
22. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine predators algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [CrossRef]
23. Hussien, A.G. An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 129–150. [CrossRef]
24. Jia, H.; Peng, X.; Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [CrossRef]
25. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
26. Mostafa, R.R.; Hussien, A.G.; Khan, M.A.; Kadry, S.; Hashim, F. Enhanced coot optimization algorithm for dimensionality reduction. In Proceedings of the 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU), Riyadh, Saudi Arabia, 28–29 March 2022.
27. Hussien, A.G.; Amin, M.; Abd El Aziz, M. A comprehensive review of moth-flame optimisation: Variants, hybrids, and applications. *J. Exp. Theor. Artif. Intell.* **2020**, *32*, 705–725. [CrossRef]
28. Cuevas, E.; Cienfuegos, M.; Zaldívar, D.; Perez-Cisneros, M. A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst. Appl.* **2013**, *40*, 6374–6384. [CrossRef]
29. Hashim, F.A.; Hussien, A.G. Snake optimizer: A novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [CrossRef]

30. Hussien, A.G.; Amin, M.; Wang, M.; Liang, G.; Alsanad, A.; Gumaei, A.; Chen, H. Crow search algorithm: Theory, recent advances, and applications. *IEEE Access* **2020**, *8*, 173548–173565. [CrossRef]
31. Dhiman, G.; Kumar, V. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowl.-Based Syst.* **2018**, *159*, 20–50. [CrossRef]
32. Hussien, A.G.; Heidari, A.A.; Ye, X.; Liang, G.; Chen, H.; Pan, Z. Boosting whale optimization with evolution strategy and gaussian random walks: An image segmentation method. *Eng. Comput.* **2022**, 1–45. [CrossRef]
33. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]
34. Abualigah, L.; Yousri, D.; Elaziz, M.A.; Ewees, A.A.; Al-qaness, M.A.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
35. Kaur, S.; Awasthi, L.K.; Sangal, A.; Dhiman, G. Tunicate swarm algorithm: A new bioinspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541. [CrossRef]
36. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
37. Genc¸, H.M.; Eksin, I.; Erol, O.K. Big bang-big crunch optimization algorithm hybridized with local directional moves and application to target motion analysis problem. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10–13 October 2010; pp. 881–887.
38. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. Gsa: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
39. Abualigah, L.; Elaziz, M.A.; Hussien, A.G.; Alsalibi, B.; Jalali, S.M.J.; Gandomi, A.H. Lightning search algorithm: A comprehensive survey. *Appl. Intell.* **2021**, *51*, 2353–2376. [CrossRef] [PubMed]
40. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [CrossRef]
41. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
42. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray optimization. *Comput. Struct.* **2012**, *112*, 283–294. [CrossRef]
43. Anita; Yadav, A. Aefa: Artificial electric field algorithm for global optimization. *Swarm Evol. Comput.* **2019**, *48*, 93–108.
44. Abualigah, L.; Diabat, A.; Mirjalili, S.; Elaziz, M.A.; Gandomi, A.H. The Arithmetic Optimization Algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
45. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2015**, *27*, 495–513. [CrossRef]
46. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [CrossRef]
47. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
48. Sinha, N.; Chakrabarti, R.; Chattopadhyay, P. Evolutionary programming techniques for economic load dispatch. *IEEE Trans. Evol. Comput.* **2003**, *7*, 83–94. [CrossRef]
49. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
50. Moscato, P.; Cotta, C.; Mendes, A. Memetic algorithms. In *New Optimization Techniques in Engineering*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 53–85.
51. Passino, K.M. Bacterial foraging optimization. *Int. J. Swarm Intell. Res. (IJSIR)* **2010**, *1*, 1–16. [CrossRef]
52. Uymaz, S.A.; Tezel, G.; Yel, E. Artificial algae algorithm (aaa) for nonlinear global optimization. *Appl. Soft Comput.* **2015**, *31*, 153–171. [CrossRef]
53. Meng, Z.; Pan, J.S. Monkey king evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. *Knowl.-Based Syst.* **2016**, *97*, 144–157. [CrossRef]
54. Zheng, R.; Jia, H.; Abualigah, L.; Wang, S.; Wu, D. An improved remora optimization algorithm with autonomous foraging mechanism for global optimization problems. *Math. Biosci. Eng.* **2022**, *19*, 3994–4037. [CrossRef]
55. Liu, Q.; Li, N.; Jia, H.; Qi, Q.; Abualigah, L. Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation. *Mathematics* **2022**, *10*, 1014. [CrossRef]
56. Vinayaki, V.D.; Kalaiselvi, R. Multithreshold image segmentation technique using remora optimization algorithm for diabetic retinopathy detection from fundus images. *Neural Process. Lett.* **2022**, 1–22. [CrossRef]
57. Shadravan, S.; Naji, H.R.; Bardsiri, V.K. The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Eng. Appl. Artif. Intell.* **2019**, *80*, 20–34. [CrossRef]
58. Zhang, H.; Wang, Z.; Chen, W.; Heidari, A.A.; Wang, M.; Zhao, X.; Liang, G.; Chen, H.; Zhang, X. Ensemble mutation-driven salp swarm algorithm with restart mechanism: Framework and fundamental analysis. *Expert Syst. Appl.* **2021**, *165*, 113897. [CrossRef]
59. Long, W.; Jiao, J.; Liang, X.; Cai, S.; Xu, M. A Random Opposition-Based Learning Grey Wolf Optimizer. *IEEE Access* **2019**, *7*, 113810–113825. [CrossRef]
60. Dhiman, G.; Kaur, A. STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Eng. Appl. Artif. Intell.* **2019**, *82*, 148–174. [CrossRef]
61. Li, S.M.; Chen, H.L.; Wang, M.J.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
62. Rechenberg, I. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie*; Springer: Berlin/Heidelberg, Germany, 1978; Volume 8, pp. 83–114.

63. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **2007**, *20*, 89–99. [CrossRef]
64. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
65. Lu, S.; Kim, H.M. A regularized inexact penalty decomposition algorithm for multidisciplinary design optimization problems with complementarity constraints. *J. Mech. Des.* **2010**, *132*, 041005. [CrossRef]
66. Wu, D.; Wang, S.; Liu, Q.; Abualigah, L.; Jia, H. An Improved Teaching-Learning-Based Optimization Algorithm with Reinforcement Learning Strategy for Solving Optimization Problems. *Comput. Intell. Neurosci.* **2022**, *2022*, 1535957. [CrossRef]
67. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
68. Wang, S.; Jia, H.; Abualigah, L.; Liu, Q.; Zheng, R. An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems. *Processes* **2021**, *9*, 1551. [CrossRef]
69. Zheng, R.; Hussien, A.G.; Jia, H.; Abualigah, L.; Wang, S.; Wu, D. An Improved Wild Horse Optimizer for Solving Optimization Problems. *Mathematics* **2022**, *10*, 1311. [CrossRef]
70. Gandomi, A.H.; Yang, X.S.; Alavi, A.H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **2013**, *29*, 17–35. [CrossRef]
71. Wang, S.; Sun, K.; Zhang, W.; Jia, H. Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation. *Math. Biosci. Eng.* **2021**, *18*, 3092–3143. [CrossRef] [PubMed]
72. Sharma, T.K.; Pant, M.; Singh, V. Improved local search in artificial bee colony using golden section search. *arXiv* **2012**, arXiv:1210.6128.
73. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [CrossRef]