Expert Systems With Applications 185 (2021) 115665

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Remora optimization algorithm

Heming Jia [a,1,*], Xiaoxu Peng [b,*,1], Chunbo Lang [c]

[a] Department of Information Engineering, Sanming University, Fujian 365004, China
[b] School of Astronautics, Harbin Institute of Technology, Harbin 150040, China
[c] School of Automation, Northwestern Polytechnical University, Xi 'an 710072, China

ARTICLE INFO

Keywords:
Remora optimization algorithm
Bionics-based
Natural-inspired
Meta-heuristic
Optimization

ABSTRACT

In this paper, Remora Optimization Algorithm (ROA) is proposed, which is a new bionics-based, natural-inspired, and *meta*-heuristic algorithm. The inspiration for ROA is mainly due to the parasitic behavior of remora. Different locations are updated in different hosts: In some large hosts, remora feeds on the host's ectoparasites or wreckage and evades natural enemies, for example in the case of giant whales. In some small hosts, remora follows the host to move to the bait-rich area to prey, taking the fast-moving swordfish as an example. In the case of these two update methods, remora also makes some judges based on experience. If it takes the initiative to prey, it updates the host, makes a global update. If it eat around the host, remora does not change the host, and continues to local update. This algorithm is more inclined to provide a new idea for memetic algorithm, because the host in ROA can be reasonably replaced, such as ships, turtles, etc. The above dynamic mode and behavior are simulated mathematically and the validity of the ROA is tested with 29 benchmark questions and 5 actual engineering questions. Parallel comparisons are made with 10 other natural heuristics. The statistical results and comparisons show that ROA provides a very promising prospect and a strong competitive ability compared to other state-of-the-art heuristic techniques.

1. Introduction

With the increase of data volume, the innovation of computer technology and the advancement of computing power, the emerging artificial intelligence has been reshaping our lives (Haeberle et al., 2019; Hossam et al., 2019). During the third industrial revolution–the computer technology revolution, the big data can be in analogy with steam and electricity, which are important energy sources to promote industrial progress. Now people have entered the era of Fourth Industrial Revolution. As a technology that can ignite these energy sources, intelligent optimization algorithm has become the "important foundation" of machine learning, the "important weapon" to solve the current complex optimization problems, or the "important engine" of the Fourth Industrial Revolution (Maximilian, Can, Tamara, & etc., 2019; Yuankang, Ziyang, Zhiqiu, & etc., 2010).

In essence, optimization belongs to "operations research". Operations research includes many familiar branches, such as robust optimization, multi-objective optimization, heuristic algorithm, etc. The heuristic algorithm depends on the technology of the problem. It may give its feasible solution in an acceptable time and space, but it does not guarantee to get the optimal solution. Because they are often too greedy, they usually fall into the local optimal state, they usually can not reach the global optimal. However, heuristic strategy provides an efficient way to obtain feasible solutions when the search for the optimal solution becomes impossible or difficult to complete. To adapt to a wide range of problems, the *meta*-heuristic algorithm is proposed as a black box. "Meta" is actually a philosophical concept, expressing the organizational unit of the world. Therefore, *meta*-heuristic algorithm can be treated as a basic method, independent of the existence of the problem. If necessary, some fine-tuning of its internal parameters will be able to adapt to the problems at hand. Nevertheless, occasionally, *meta*-heuristic algorithms show strong sensitivity to adjusting these user-defined parameters (Amir, Xin-She, & Siamak, 2013). Because of their simplicity and wide applicability, many algorithms, such as particle swarm optimization (PSO) (James & Russell, 1995), polar bear optimization algorithm (PBO) (Dawid & Marcin, 2017) and Red Fox Optimization Algorithm (RFO) (Dawid & Marcin, 2021) have been proposed successively. Correspondingly, there are many application scenarios in the field of engineering: Jaya algorithm has been used to optimize the quality of top beam structures with natural frequency constraints, to optimize the

* Corresponding authors.
E-mail addresses: jiaheming@fjsmu.edu.cn (H. Jia), pengxiaoxu@nefu.edu.cn (X. Peng), langchunbo@nefu.edu.cn (C. Lang).
[1] Heming Jia and Xiaoxu Peng contribuited equally.

https://doi.org/10.1016/j.eswa.2021.115665
Received 29 August 2020; Received in revised form 6 July 2021; Accepted 23 July 2021
Available online 31 July 2021

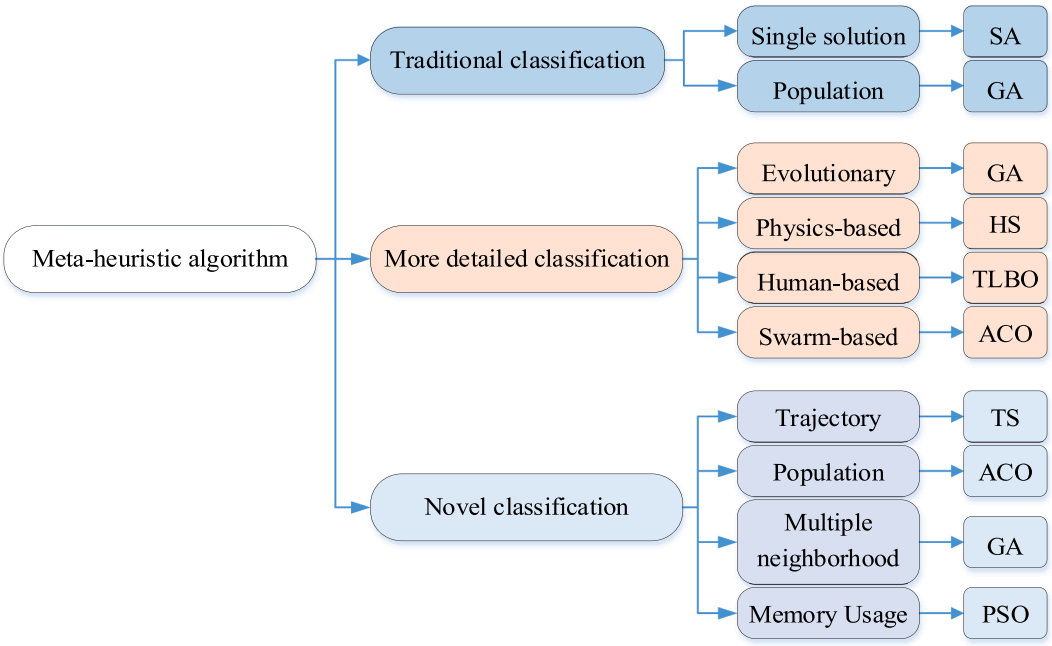


Fig. 1. The classification of *meta*-heuristic algorithms.

Fig. 2. The individual remora.

Fig. 3. Remora parasitizes on different hosts.

design of reinforced concrete retaining walls and to optimize the size of cable and prestress of cable-stayed Bridges to minimize the cost. (Grzywiński & Dede, 2019; Ztürk, Dede, & Türker, 2020, Atmaca, Dede, & Grzywiński, 2020). Designed a low-weight cantilever reinforced concrete retaining wall with shear key by using Grey wolf optimization algorithm (GWO) (Kalemci, İkizler, Dede, & Angın, 2020). Thus, it is proved that, it has been entered the blowout period of *meta*-heuristic algorithm. To date, there are many types of *meta*-heuristic algorithms. As shown in Fig. 1.

1. The traditional categorization is classified based on a single solution (such as simulated annealing (SA) (Scott, Gelatt, & Mario, 1983)) and population (such as genetic algorithm (GA) (John, 1992; El-Ghazal, 2009). Standing in the general direction of data processing, the main basis is the number of solutions processed in the iterative optimization phase. Among them, population-based *meta*-heuristics mostly imitate natural phenomena (Jia, Lang, Oliva, & etc., 2019; Mingjing, Huiling, Bo, & etc., 2017; Liming, Huiling, Zhe, & etc., 2016; Zhang, Huiling, Jie, & etc., 2018), which are easy to develop and have achieved remarkable results.
2. A more fine-grained classification method is to classify the *meta*-heuristic algorithm according to the source of inspiration, that is, to select the object of modeling: evolutionary algorithm (such as genetic algorithm (GA) (John, 1994)), based on physics (for instance, harmony search (HS) (Kang & Zong, 2004), based on human (such as teaching-based optimization (TLBO) (Rao, Savsani, & Vakharia, 2012)) and swarm intelligence algorithm (for example, ant colony optimization (ACO) (Marco, Vittorio, & Alberto, 1996)). This classification is intuitional. As the literal meaning indicates, they are either inspired by the laws of biological evolution, or imitate the laws

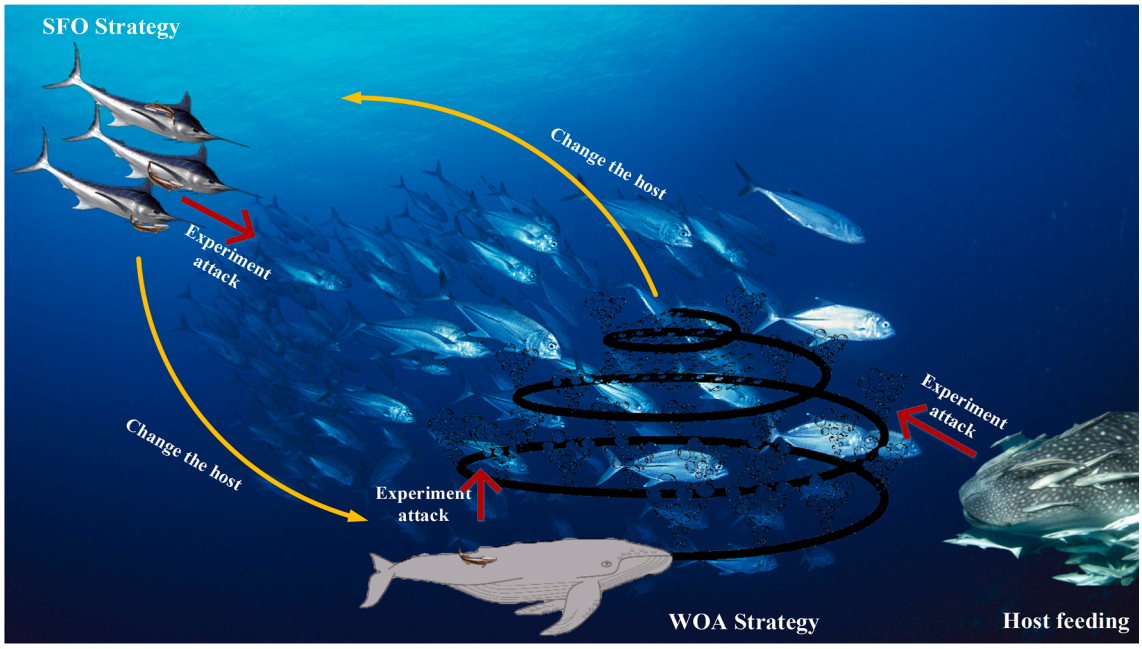


Fig. 4. The different states of remora.

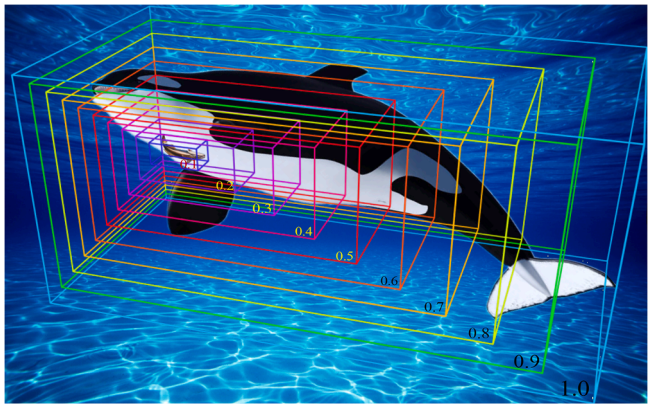


Fig. 5. The Host feeding model under on condition that $C = 0.1$.

Table 1
Mean ranks of convergence factor C.

C	0	0.1	0.2	0.3	0.4
Mean ranks	2.8000	1.7500	2.9000	3.4250	4.1250

of physics in the world, or learn the intelligent models among biological populations, or are based on human technology and the laws between them.

3. The third most proposed classification method is relatively novel. Trajectory methods: the main idea is to track a single search agent's trajectory, to avoid local optimization by moving a poor solution, taking the tabu search algorithm (TS) as an example (Fred, 1989); Population Based: using a group of search agents for optimization, these methods are suitable for the improvement of the exploration stage, the ant colony optimization algorithm (ACO) is taken as an example; Multiple neighborhood: using multiple neighborhood structures in the algorithm as the distinction, local and global optimum are realized by switching between neighborhoods, the appropriate method of segment is the mutation in the genetic algorithm (GA); Memory Usage: the relevant algorithm contains a valid memory function, which can preserve the local optimum and the global optimal in memory. A solution that provides a reference for future optimization, taking PSO as an example.

In practical terms, different classification methods can not only facilitate the understanding and learning of optimization algorithms. Furthermore, it also provides reference and theoretical support for improving the fusion of algorithms. Existing optimizers can be divided into two stages in search steps: exploration (diversification) and exploitation (enhancement) (Esmat, Hossein, & Saeid, 2009). In the exploration stage, the more random the algorithm provides, the better it is to explore the regions and boundaries of the feature space in depth. Without randomness, it will fall into local optimum. In order to obtain a higher probability to find the optimal solution, there must be enough randomness to be introduced by the algorithm. Specifically, it has better robustness and diversity in the search process. The randomness can be considered in every step of the algorithm, but the probability must be well controlled (Heidari et al., 2019). The exploitation stage usually takes place after the exploration stage. At this stage, the optimization program will pay more attention to finding a better solution. Usually, an effective exploration can avoid falling into local optimum, leading to an improved performance; Efficacious development can reduce the immature convergence defects and hopefully bring a qualitative improvement to the algorithm. The balance between the two will bring a qualitative leap. A well-organized optimizer can achieve a reasonable and fine balance between exploration and development trends. As increasing new algorithms and hybrid algorithms are proposed, the theorem of no-free lunch (NFL) (David & William, 1997) is more accepted by people. In theory, there is no algorithm that can be used as a general optimal optimizer. The average performance of all stochastic algorithms is the same, and there is likely no stochastic algorithm that can completely sovle all optimization problems. This determines the application limitation of the specific stochastic algorithm, that is, the algorithm with certain characteristics can only achieve better performance on a particular problem.

In order to design and develop a more comprehensive optimization algorithm, this paper proposes a new *meta*-heuristic optimization algorithm, the Remora Optimization Algorithms (ROA), which is inspired by the "intelligent traveler" in the ocean, the remora. Through the idea of parasitism and random host replacement of remora, a remora-based

Table 2
Two dimensional scatter plots of test functions under different convergence factor C.

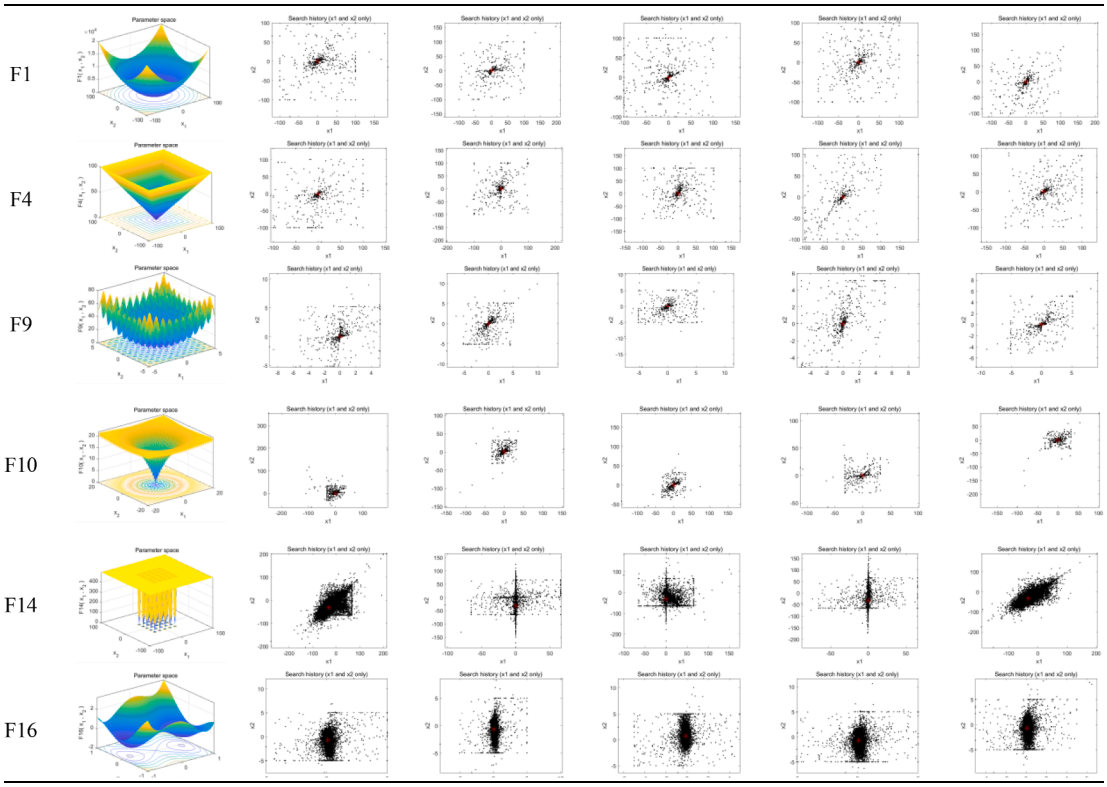


Table 3
Pseudocode for the ROA algorithm.

Initialize the population location $R_i(i = 1, 2, \cdots n)$ and memory location R_{pre};
Initialize the optimal solutionR_{Best} and corresponding optimal fitness $f(R_{Best})$;
While t < Max_iter **do**
Calculat the fitness function value of each remora;
Check if any search agent goes beyond the search space and amend it;
Update a,α and V;
For each remora indexed by i **do**
If H(i) = 0 **then**
Using Eq. (5) to update the position of attached whales;
Elseif H(i) = 1 **then**
Using Eq. (1) to update the position of attached Sailfishes;
Endif
Make a one-step prediction by Eq. (2);
Determine the value of H(i) by Eqs. (3) and (4) to juge whether host replacement is necessary;
If the host is not replaced, Eq. (9) is used as the host feeding mode for remora;
End for
End while

framework was created. It can learn the effective characteristics of the host. In this paper, whale optimization algorithm (WOA) (Seyedali & Andrew, 2016) and swordfish optimization algorithm (SFO) (Shadravan, Naji, & Bardsiri, 2019) are taken as examples. They are recognized as effective algorithms and relatively new algorithms, respectively. Based on their effective position update formula, the main research object of this paper is constructed. WOA is a typical whale optimization algorithm which was proposed by Mirjalili in 2016. The superiority of the algorithm has been validated by numerous application scenarios, and a variety of variants have been developed. SFO was proposed by S. Shadravan etc. in 2019 whose main inspiration of the SFO algorithm is based on the attack-alternation strategy of sailfish's group hunting. Similar to all the other group hunting algorithms, sailfish are also updated according to the target. Interestingly, swordfish and sardine populations are updated with elite strategies in each iteration. And it also takes advantage of the rotation attack characteristics of the swordfish population.

A novel algorithmic fusion framework is established by switching between remora and two kinds of hosts. The framework contains two movement modes corresponding to the exploration and exploitation stages respectively. Mode switching is achieved through a self-built "one small step try". "Remora factor" is put forward in the exploration stage, which can improve the precision of optimization and result in effectively convergence. The establishment of these effective mechanisms makes ROA achieve promising results in the research outcomes. In order to explain the overall optimization capability of ROA in more detail, some significant points are as follows:

- The formulas in ROA involving SFO and WOA were improved on the original form, and a formula which was more suitable for the movement of remora was obtained.
- By simulating the whole process of remora's predation, the exploration and exploitation were divided into two processes: "Free travel" and "Eat thoughtfully", and the "experience" of remora was taken as the main basis for the judgment of phase switching.
- "Free travel" is further divided into "SFO Strategy" and "Experiment attack" global optimization methods, which correspond to "big global" and "small global" respectively in the algorithm. Besides, "Experiment attack" is the main way for remora to accumulate experience.
- "Eat thoughtfully" is further divided into two feeding modes which are "WOA Strategy" and "Host feeding". Different convergence methods enhance the stability of the algorithm.
- The "SFO Strategy" and "WOA Strategy" in the whole algorithm framework can be reasonably replaced by other algorithms. Therefore, a more important highlight of this paper is to provide a framework for memetic algorithm (Talbi, 2002).

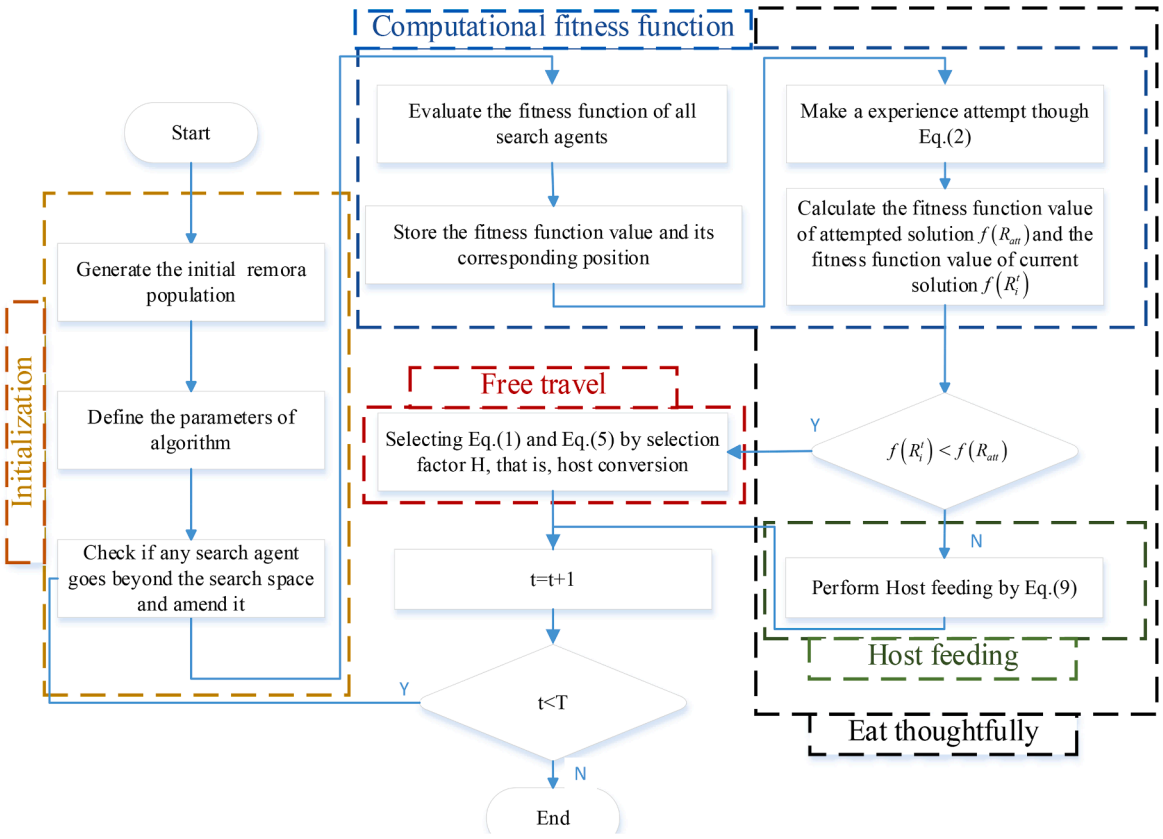


Fig. 6. The flow chart of ROA.

Table 4
Unimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	[−100, 100]	0
$F_4(x) = max\{ \lvert x_i \rvert, 1 < i < n \}$	30	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[−100, 100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[−1.28, 1.28]	0
Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	[−100, 100]	0
$F_4(x) = max\{ \lvert x_i \rvert, 1 < i < n \}$	30	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[−100, 100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[−1.28, 1.28]	0

- "Host feeding" creates a solution area that converges gradually around the host, which refines and enhances the ability of local optimization. Innovative remora factor is proposed, which perfectly expresses the position of remora on the host vividly. The purpose of distinguishing the locations of remora and its host was achieved.

The overall structure of this paper is as follows: Section 2 describes the remora optimization algorithm, including the source of inspiration and mathematical modeling of various strategies. In Section 3, the experimental results of the test function are introduced and discussed. In Section 4, the research aspects of solving practical engineering problems with different benchmarks are discussed and analyzed. Finally, this paper summarizes the work and describes the perspective of future development.

2. Remora Optimization Algorithm (ROA)

2.1. Inspiration

Remora, suckerfish, diskfish, and sucker are some of the names describing eight species of marine fishes in the Family Echeneidae (Fischer, Fao, & FIR, 1978; Nelson, Crossman, & Espinosapérez, 2006). Remora has a extremely prolonged body, the head is flat, and gradually becomes cylindrical backward. At the top, there are suckers deformed from the first dorsal fin. Its fins split from the center of the sucker to both sides, forming about 21–28 fin flaps (laminae) (Ralf & Johnson, 2012). Remora's more detailed body features can be found in reference (Fertl & Landry, 1999). As shown in Fig. 2.

Remora is famous for its ability to swim on whales or other marine animals or oceangoing hulls. This habit not only labor-saving but also free from the enemy's invasion. Usually it distributed in tropical waters, but it also follow the movement of the host to the cold waters. Remora mainly feeds on other fish or invertebrates. When it reaches the sea area rich in bait, it will leave the host, ingest food, and then adsorb to the new host and continue to transfer to another sea area. But in some cases, just like a cleaner, it is also possible to eat the wreckage food or ectoparasites outside the carrier (Cressey & Lachner, 1970; Williams, Mignucci-Giannoni, & Bunkley-Williams, 2003). The host varies greatly. Whales, sharks, sea turtles, sunfish, swordfish, and boats can all be boarding objects, even with divers (Bruce, 2002). Seen in Fig. 3.

How does remora attach itself to swordfish or whale? It turns out that whenever a remora sees a swordfish or a big shark passing by, it immediately swims forward, and hold its body tightly to them. The

Table 5
Multimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i sin\left(\sqrt{x_i}\right)$	30	[−5.12, 5.12]	0
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10cos(2\pi x_i) + 10\right]$	30	[−5.12, 5.12]	0
$F_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	30	[−32, 32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600, 600]	0
$F_{12}(x) = \frac{\pi}{n}\{10sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1}) + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < a \end{cases}$	30	[−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	[−50, 50]	0

Table 6
Fixed-dimenstion multimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[−65, 65]	1
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5, 5]	0.0003
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4p^2}x_1^2 + \frac{5}{p}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8p}\right)cosx_1 + 10$	2	[−5, 5]	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	[−2, 2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[1, 3]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	[0, 1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	–10.5363

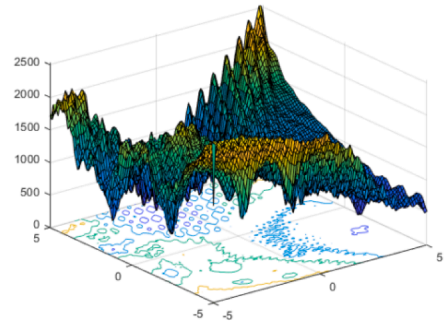
Table 7
Composition benchmark functions.

Function	Dim	Range	f_{min}
$F_{24}(C16)$ $f_1, f_2, f_3, \ldots, f_{10}$= Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	30	[−5, 5]	0
$F_{25}(C18)$ $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	30	[−5, 5]	0
$F_{26}(C19)$ $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1, 1, 1, \ldots, 1]$	30	[−5, 5]	0
$F_{27}(C20)$ f_1, f_2= Ackley's Function, f_3, f_4= Rastrigin's Function, $f_5.f_6$= Weierstrass Function, f_7, f_8= Griewank's Function, f_9, f_{10}= Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] =$ $[5/32, 5/32, 1, 1, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	30	[−5, 5]	0
$F_{28}(C21)$ f_1, f_2= Rastrigin's Function, f_3, f_4= Weierstrass Function, f_5, f_6= Griewank's Function, f_7, f_8= Ackley's Function, f_9, f_{10}= Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] =$ $[1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	30	[−5, 5]	0
$F_{29}(C25)$ f_1, f_2= Rastrigin's Function, f_3, f_4= Weierstrass Function, f_5, f_6= Griewank's Function, f_7, f_8= Ackley's Function, f_9, f_{10}= Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] =$ $[0.1,\ 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [0.1*1/5, 0.2*1/5, 0.3*5/0.5, 0.4*5/0.5, 0.5*5/100,$ $0.6*5/100, 0.7*5/32, 0.8*5/32, 0.9*5/100, 1*5/100]$	30	[−5, 5]	0

membrane and cartilage plates are then immediately raised so that the water is squeezed out of the sucker. At this point, the part of the sucker that becomes a vacuum. Enormous pressure of the sea water outside the suction cup and the many vertical plates on the suction cup increase the resistance of remora to slip, thus enhancing the friction force that remain attached to the moving host and remora is firmly fixed. It has been determined that a 60 cm remora can withstand the pull of 10 kg (Gudger, 1919).

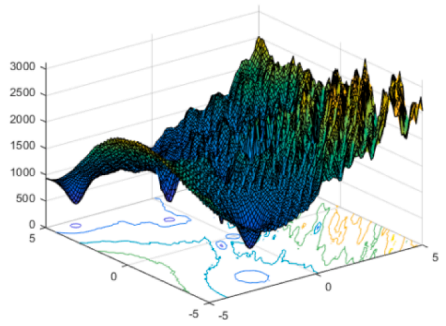
It is worth mentioning that the "intelligence" of remora has been applied to many fields, but its whole process from hunting for prey to eating has rarely utilized. In this algorithm, the two species of sea creatures allowing remora to be attached to. They are humpback whales and sailfishes. Normally, humpback whales hunts alone. The most interesting thing about the humpback whales is their special hunting

F24: Rotated Hybrid Composition Function



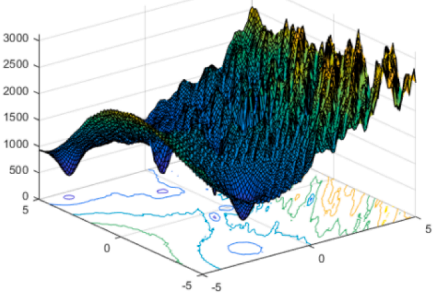
Properties: MM, R, NS, S

F25: Rotated Hybrid Composition Function



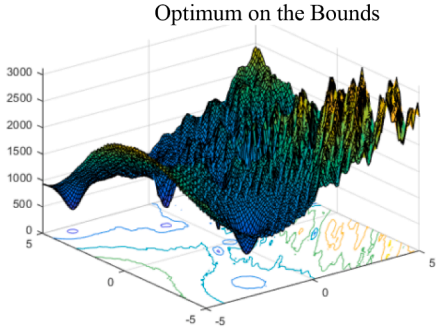
Properties: MM, R, NS, S

ɔ: Rotated Hybrid Composition Function with narrow basin global optimum



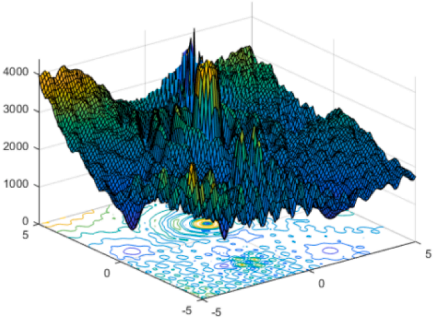
Properties: MM, NS, S

F27: Rotated Hybrid Composition Function with Global Optimum on the Bounds



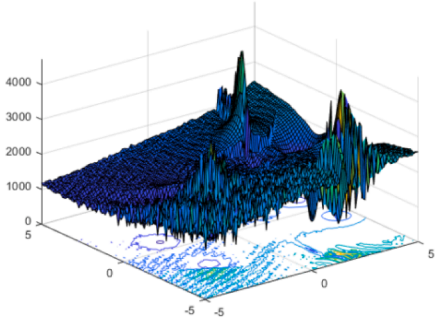
Properties: MM, NS, S

F28: Rotated Hybrid Composition Function



Properties: MM, R, NS, S

F29: Rotated Hybrid Composition Function without bounds



Properties: MM, NS, S

Fig. 7. Demonstration of composition test functions (MM: Multi-modal, R: Rotated, NS: Non-Separable, S: Scalable).

method which called bubble-net feeding method. Sailfish is the fastest fish in the ocean that can reach maximum speeds of around 100 km/h. Normally, they hunt in groups. Their special hunting method is take turns attacking elite tactics. This two special sea creatures can be a perfect representation of remora's host. One is a huge body and usually hunting alone, the other is fast and usually hunting in group. The well-recognized effect motion formulas of WOA algorithm and the latest swordfish algorithm are extracted as the motion formulas of remora. As a "smart" fish, remora's attack is based on a certain experience. Whatever whale or swordfish swims, remora always keeps close to them. As a rule of thumb, if its favorite food is near or it feels too hungry, it rushes like an arrow and grabs the food in its mouth first. When there is no need to take too much risk, the leftover food residues of whales and swordfishes or their parasites can satisfy remora. In this paper, the above interesting biological habits are mathematically modeled and the purpose of optimization is achieved.

In the ROA, remora initially did a global movement, the exploration stage, and named this stage "Free travel". This stage includes a "big global" movement, which corresponds to remora adsorbing on the sailfish, and update the position of the sailfish for long-distance position update. In addition, there is also a "small global" movement, which is a one-step attempt to update remora itself in an "Experiment attack". At

Table 8
Parameters of the compared algorithms.

Reference	Algorithm	Parameters	Value
(Ali et al., 2019)	HHO	Initial energy E_0	(−1, 1)
		Jump strength J	(0,2)
		Prey escape probability r	0.5
		Random parameters r_1, r_2, r_3, r_4	[0,1]
(Shadravan et al., 2019)	SFO	Population percentage PP	0.2
		Prey density PD	(0,1)
		Coefficientλ_i	(−1,1)
		Random parameter r	(0,1)
		Coefficients of power attack A,ε * *MERGEFORMAT	4, 0.001
		Number of update sardines	α
		Number of variables of update sardines	β
(Gaurav & Vijay, 2018)	EPO	Temperature Profile (T′)	[1, 1000]
		$\vec{A}$Constant	[−1.5, 1.5]
		Function S()	[0, 1.5]
		Parameter M	2
		Parameter f	[2,3]
		Parameter l	[1.5,2]
(Gaurav & Vijay, 2019)	SOA	Movement behavior A	[0,2]
		Control parameterf_c	2
		Range of parameter k	[0,2π]
		Helical parameter u	1
		Helical parameter v	1
		Random parameter rd	(0,1)
(Dhiman & Kumar, 2017)	SHO	Control Parameter $\vec{h}$	[0,5]
		Constant $\vec{M}$	[0.5,1]
		Co-efficient vector $\vec{B}$	[0,2]
		Co-efficient vector $\vec{E}$	[0,2]
(Seyedali et al., 2017)	SSA	Control Parameter c_1	$_{[2,e^{-16}]}$
		Random parameters c_2,c_3	(0,1)
(Seyedali & Andrew, 2016)	WOA	Coefficient vectors$\vec{A}$	[0,2]
		Coefficient vectors$\vec{C}$	[0,2]
		Helical parameter b	1
		Helical parameter l	[−1,1]
(Seyedali, 2015a)	MFO	Logarithmic spiral constant b	0.75
		Random parameter t	[−1,1]
		Convergence Constant r	[−1,−2]
(Seyedali et al., 2015)	MVO	Mining capability$p$	1/6
		Random parametersr_1,r_2,r_3r_4	[0,1]
		Contrast parameter H	0.5
		WEP	[0.2,1]
		TDR	[0,1]
(Seyedali, Seyed, & Andrew, 2014)	GWO	Convergence constant a	[0,2]
		Random parametersr_1,r_2	[0,1]

Table 9
Test function performance indicators.

Category	Name	Formulation	Reference
Performance evaluation	Average Operating Time (AT)	$Average\_Time = \frac{\sum_{i=1}^{N} time}{N}$	(Oliva et al., 2017)
	Average fitness function value (AF)	$Average\_Fitness = \frac{\sum_{i=1}^{N} f_i}{N}$	(Ibrahim et al., 2018)
	Standard deviation (Std)	$Std = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(f_i - Average)^2}$	(Jia et al., 2019)
Statistical evaluation	Wilcoxon's Rank-Sum test (WT)	$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$ $R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$	(Derrac et al., 2011)
	Friedman test (FT)	$F_f = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]$	(Friedman, 1937)

the same time, the result of the attempt is an important basis for whether to change the host. During the exploration phase, remora performs partial renewal. At this stage, remora gradually converges towards the target, that is, eat. This stage is named "Eat thoughtfully". There are two ways of feeding for remora, that is, to follow a large host to eat or to feed on parasites on the host. This corresponds to the special "bubble net" location update mode of WOA and the "Host feeding" in the algorithm, respectively. In the module of "Host feeding", a "remora factor" was created according to the different positions of the other parts of the whale to improve the accuracy of the optimization. To give an example, at the beginning of the "Free travel" stage, remora is attached to the sailfish to do a quick global search for optimization (SFO Strategy), looking for areas rich in bait (optimum solution). During this period, remora continues to make a small step to try to prey (Experiment attack). When the surrounding bait is low, remora randomly chooses whether to change the host. Then enter the "Eat thoughtfully" stage. If the host is changed to a whale, remora preys with the whale (WOA Strategy). During this period, remora continues to try (Experiment attack). If the host is not changed, remora feeds on the host (Host feeding).Continue this process until the optimal solution is found. To show the above process more vividly, Fig. 4 is drawn. The relevant formulas are introduced in the next section.

2.2. Mathematical model and Remora Optimization Algorithm (ROA)

In this subsection, the mathematical models of "Free travel", and "Eat thoughtfully" are provided. Then the advantages of ROA algorithm are summarized

2.2.1. Initialization

In the proposed ROA algorithm, it is assumed that the candidate solution is remora, and its position R in the search space is the variable of the problem. When remora swim in one-dimensional, two-dimensional, three-dimensional or super-dimensional space, their position vector changes accordingly. The current position is $R_i = (R_{i1}, R_{i2}, \cdots, R_{id})$, where i represents the number of remora and d represents the dimension in the search space of remora. In the similar way, $R_{Best} = (R_1^*, R_2^*, \cdots, R_d^*)$ represents food (target) in biological habits, which also represents the optimal solution in the algorithm. In an algorithm, each candidate solution should have a corresponding fitness function. Here it can be expressed as $f(R_i) = f(R_{i1}, R_{i2}, \cdots, R_{id})$. f is the corresponding rule for calculating the fitness function value. $f(R_{\text{Best}}) = f(R_1^*, R_2^*, \cdots, R_d^*)$ save the best fitness value corresponding to the best remora location. Notable, remora is the main factor to find a solution, and it is scattered in the search space. Other marine organisms or ships are just tools to assist remora in location updating, and it has its own way of updating locations. With these tools, remora can find the best location in the region. In fact, there are two modes for determining the best position, mainly based on whether or not to take the initiative to select different modes.

2.2.2. Free travel (Exploration)

- SFO Strategy

When remora attach themselves to the swordfish, its position can be considered as being updated at the same time. Based on the elite idea of this algorithm, the formula of its location update was improved and get the following formula:

$$R_i^{t+1} = R_{Best}^t - \left(rand(0,1) * \left(\frac{R_{Best}^t + R_{rand}^t}{2}\right) - R_{rand}^t\right) \quad (1)$$

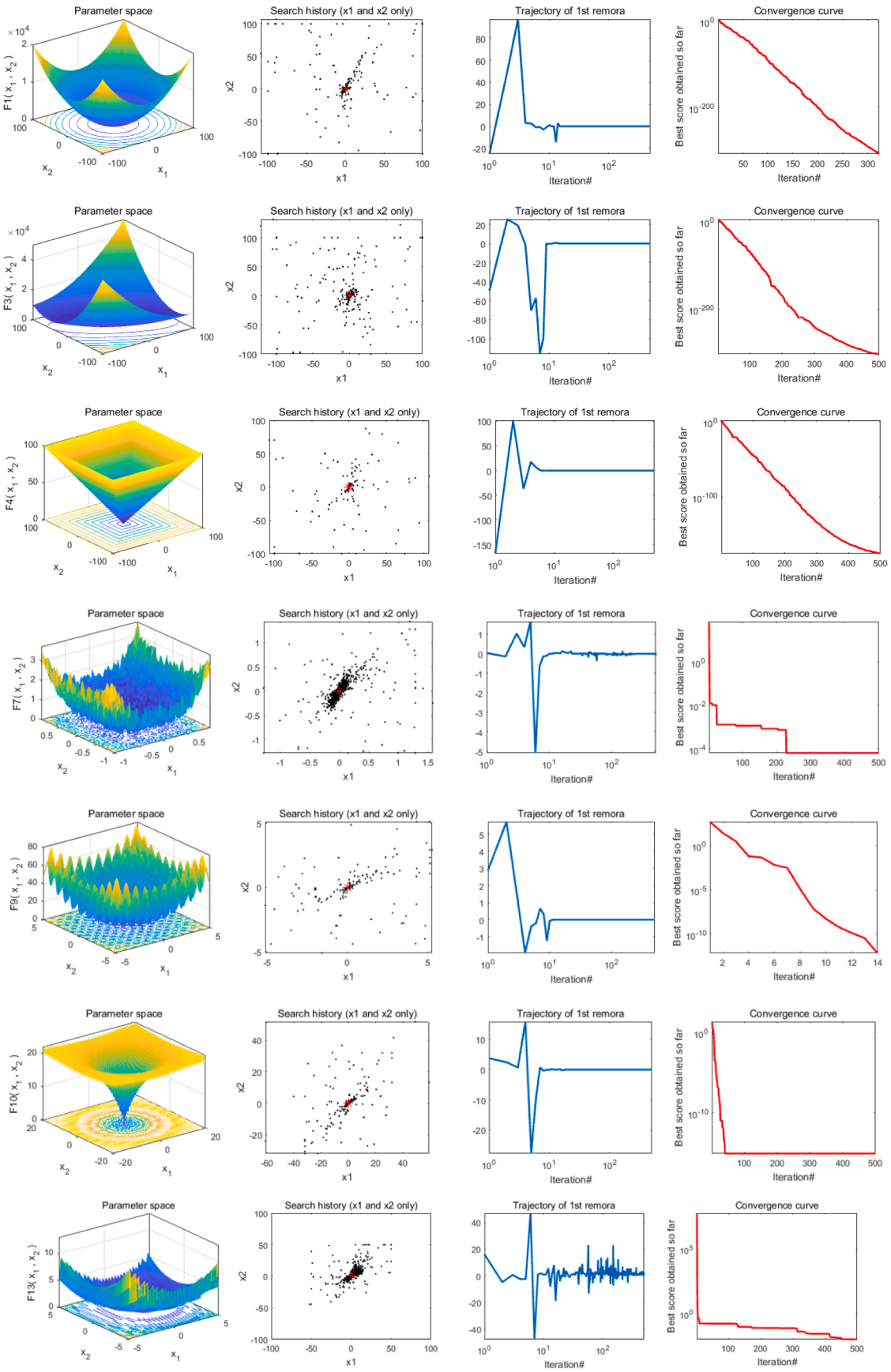


Fig. 8. Quantitative measurement effect charts of functions 1, 3, 4, 7, 9, 10, 13.

Table 10
Results of benchmark functions (F1 - F13) under different dimension.

Benchmark		30	100	500	1000
F1	AF	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	STD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
F2	AF	2.80E−191	2.72E−183	5.41E−168	4.38E−183
	STD	6.07E−161	0.00E + 00	8.15E−160	1.32E−157
F3	AF	1.26E−321	6.79E−305	1.45E−302	4.40E−285
	STD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
F4	AF	3.16E−169	5.12E−167	2.62E−167	3.39E−165
	STD	0.00E + 00	1.50E−157	2.86E−157	4.45E−162
F5	AF	2.70E + 01	9.81E + 01	4.95E + 02	9.90E + 02
	STD	4.61E−01	4.17E−01	2.22E−01	5.65E−01
F6	AF	3.90E−02	1.35E + 00	1.01E + 01	1.07E + 01
	STD	1.30E−01	6.96E−01	5.96E + 00	1.69E + 01
F7	AF	3.55E−04	4.81E−05	4.41E−04	4.63E−05
	STD	1.54E−04	2.79E−04	2.22E−04	1.76E−04
F8	AF	−1.26E + 04	−4.12E + 04	−2.09E + 05	−4.04E + 05
	STD	1.03E + 03	5.97E + 02	8.20E + 03	1.08E + 04
F9	AF	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	STD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
F10	AF	8.88E−16	8.88E−16	8.88E−16	8.88E−16
	STD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
F11	AF	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	STD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
F12	AF	6.93E−03	2.23E−02	8.92E−03	2.66E−02
	STD	6.60E−03	1.13E−02	2.23E−02	2.06E−02
F13	AF	3.16E−01	1.07E + 00	1.17E + 01	1.75E + 01
	STD	1.38E−01	9.86E−01	3.64E + 00	9.22E + 00

Table 11
Friedman test for AF of different dimensional benchmark functions (F1–F13).

Standard	30	100	500	1000
FT	1.8654	2.3269	2.7885	3.0192

Here t represents the number of current iterations and T is the maximum number of iterations. Where R_{rand} is a random location. Elite select remora's historical optimal position to lead the update. Meanwhile, random selection of remora is added to ensure the exploration of search space. The selection of remora for different hosts mainly depends on whether it has eaten prey or not, that is, whether the fitness function value obtained at present is better than that of the previous generation. In fact, the current fitness function value is obtained by "Experience attack".

- Experience attack

In order to determine whether it is necessary to change the host, the tuyu needs to continuously make a small step around the host, similar to the accumulation of experience. When the above ideas are modeled, the formula is as follows:

$$R_{att} = R_i^t + \left(R_i^t - R_{pre}\right)*randn \tag{2}$$

Where R_{pre} represents the position of the previous generation, which can be seen as a kind of experient. R_{att} indicates a tentative step. When remora makes such an active step, it can be regarded as a "small global" movement, therefore, the*randn*was choosed. This mechanism can taking into account the predictability while effectively jump out of the local optimum, and has been developed in a wider range of development. After this step, a step of judgment is required, after which, remora randomly chooses whether to replace the host or not. The judgment of this step in the algorithm can be expressed as the comparison between the fitness function values of the current solution $f(R_i^t)$ and the attempted solution $f(R_{att})$. Take solving the minimum problem as an example, if the fitness function value obtained by the attempted solution is smaller than that of the current solution,

$$f\left(R_i^t\right) > f(R_{att}) \tag{3}$$

Remora then selects a different feeding method for local optimization, which is illustrated in the next section. If the fitness function value of the attempted solution is larger than that of the current solution, then back to host selection.

$$f\left(R_i^t\right) < f(R_{att}) \tag{4}$$

2.2.3. Eat thoughtfully (Exploitation)

- WOA Strategy

On the basis of the original WOA algorithm, the position updating formula of remora attached to the whale was extracted. As shown below:

$$R_{i+1} = D*e^{\alpha}*cos(2\pi\alpha) + R_i \tag{5}$$

$$\alpha = rand(0,1)*(a-1)+1 \tag{6}$$

$$a = -\left(1 + \frac{t}{T}\right) \tag{7}$$

$$D = |R_{Best} - R_i| \tag{8}$$

In a larger solution space, when a remora is on a whale, their positions can be regarded as the same. Where D is the distance between hunter and prey (current optimal solution),$\alpha$is a random number in $[-1,1]$, and a goes down linearly between $[-2,-1]$.

- Host feeding

Host feeding is a further subdivision in the exploitation process. At this stage the solution space can be reduced to the location space of the host. Moving on or around the host can be thought of as small steps, which can be mathematical modeled as:

$$R_i^t = R_i^t + A \tag{9}$$

$$A = B*\left(R_i^t - C*R_{Best}\right) \tag{10}$$

$$B = 2*V*rand(0,1) - V \tag{11}$$

$$V = 2*\left(1 - \frac{t}{Max\_iter}\right) \tag{12}$$

Here A was used to denote a small step of movement which is related to the volume space of the host and remora. In order to distinguish the location of the host and remora in the solution space, a remora factor C was used to narrow the position of remora. Suppose that the volume of the host is 1, then the volume of remora is approximately a fraction of that of the host. The Fig. 5 show the "Host feeding" model under on condition that$C = 0.1$. In order to determine the value of C, the remora factor is taken as the following five values, 0, 0.1, 0.2, 0.3, 0.4. Under the conditions of different remora factors, two functions are selected from three test functions, each of which runs 30 times independently. Friedman test is used to rank the integrity of ROA optimization algorithms with different remora factors. The evaluation criteria include fitness function value, standard deviation of 30 optimal fitness function value, 30 average running time and 30 standard deviation of running time. As can be seen from Table 1, when C is taken as 0.1, the best effect can be obtained. Meanwhile, Table 2 also gives the two-dimensional scatter diagram of the test function under different remora factor C.

The experiment proves that the effect is better when the value range of C is$[0, 0.3]$. Above 0.3, the overall effect is not good enough. Further, it is biologically understood that remora has the greatest ability to eat and avoid predators when its exploratory motility is one-tenth the size of its host. Thereby achieving more refined optimization though the Host feeding. Since the volume of the host is random, use B to simulate a

Table 12
Results of benchmark functions (F1 - F13) under 30 dimension.

Benchmark		ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	AF	**0.00E + 00**	1.34E−10	1.52E−03	1.37E + 01	3.27E−03	3.17E−14	9.15E−10	1.78E−75	3.56E−14	2.11E−50	1.61E−61
	STD	0.00E + 00	1.38E−93	4.06E−02	3.13E + 00	5.04E−02	0.00E + 00	2.50E−10	2.27E−69	1.90E−13	0.00E + 00	9.25E−57
F2	AF	**0.00E + 00**	7.86E−51	5.89E−02	4.07E−24	7.66E−04	5.06E−24	7.02E−06	3.44E−51	9.23E−09	1.61E−61	6.30E−33
	STD	0.00E + 00	6.33E−51	5.40E−02	1.78E−07	1.65E−02	0.00E + 00	1.09E−05	1.33E−50	1.04E−08	0.00E + 00	1.01E−32
F3	AF	**0.00E + 00**	1.97E−11	4.11E−04	1.04E + 04	3.46E + 02	9.00E + 00	2.17E−07	2.21E + 02	1.01E−02	4.24E−50	1.24E−25
	STD	0.00E + 00	6.49E−85	7.34E−02	1.24E + 03	1.57E + 05	3.29E + 01	6.95E−08	2.32E + 02	2.11E + 03	0.00E + 00	1.55E−24
F4	AF	**0.00E + 00**	1.02E−49	4.77E−02	3.72E + 01	8.57E−01	3.00E + 00	1.81E−05	2.30E + 00	6.26E−02	6.66E−14	6.73E−19
	STD	0.00E + 00	3.38E−49	1.41E−01	1.55E + 01	1.20E + 00	1.51E + 00	2.27E−05	1.97E + 01	3.02E + 00	0.00E + 00	3.83E−18
F5	AF	6.56E + 00	**8.47E−05**	9.03E + 00	9.24E + 03	2.79E + 00	9.00E + 00	9.46E + 00	8.71E + 00	2.41E + 02	3.19E + 02	6.25E + 00
	STD	9.94E + 01	1.14E−02	3.28E + 00	4.93E + 04	7.64E + 08	2.85E + 00	1.50E + 02	6.27E−01	1.26E + 03	4.53E−01	6.44E−01
F6	AF	**1.42E−13**	8.80E−05	1.50E + 00	1.38E + 00	3.73E−02	2.50E + 00	4.89E−10	3.98E−01	3.82E−04	2.50E + 00	2.55E−06
	STD	0.00E + 00	5.03E−05	3.55E−01	6.49E + 00	5.02E−02	0.00E + 00	4.22E−10	2.05E−01	2.39E−14	1.59E−04	1.10E−06
F7	AF	1.58E−04	8.72E−05	1.13E−03	5.69E−01	9.71E−03	**3.00E−05**	5.99E−03	5.85E−03	9.50E−03	8.63E−05	2.51E−04
	STD	4.52E−05	1.00E−04	1.92E−03	2.19E−01	1.53E−02	4.32E−05	7.45E−03	2.16E−03	4.40E−03	1.21E−04	3.65E−04
F8	AF	4.19E + 03	**4.36E + 01**	1.42E + 03	2.66E + 03	4.19E + 03	2.59E + 03	1.51E + 04	1.11E + 02	3.59E + 03	2.02E + 03	4.40E + 01
	STD	3.57E + 02	9.90E + 44	2.03E + 02	5.81E + 02	4.34E−03	4.93E + 02	4.74E + 03	9.71E + 10	3.07E + 02	4.00E + 01	4.06E + 36
F9	AF	**0.00E + 00**	**0.00E + 00**	3.09E−05	7.28E + 01	5.96E−100	9.12E−146	1.59E + 01	6.91E−105	3.38E + 01	4.65. E−289	3.32E + 00
	STD	0.00E + 00	0.00E + 00	9.35E−02	3.91E + 01	3.59E + 02	0.00E + 00	8.02E + 00	0.00E + 00	1.20E + 01	0.00E + 00	1.50E + 00
F10	AF	**8.88E−16**	**8.88E−16**	2.47E−01	3.00E + 00	1.53E−05	**8.88E−16**	1.03E−05	**8.88E−16**	8.53E−08	**8.88E−16**	7.99E−15
	STD	0.00E + 00	0.00E + 00	1.75E−01	6.89E + 00	2.19E + 03	0.00E + 00	1.03E + 00	2.40E−15	7.93E−08	0.00E + 00	2.02E−15
F11	AF	**0.00E + 00**	**0.00E + 00**	5.70E−01	6.73E−01	2.10E−04	**0.00E + 00**	2.81E−01	**0.00E + 00**	1.48E−01	4.40E−01	**0.00E + 00**
	STD	0.00E + 00	0.00E + 00	2.45E−01	4.85E−01	3.91E + 02	5.86E−02	1.26E−01	2.98E−01	1.14E−01	0.00E + 00	2.60E−02
F12	AF	2.01E−03	**1.35E−07**	8.70E−01	2.35E + 03	7.52E−03	5.30E−01	1.27E + 00	4.70E−03	3.11E−01	1.37E−01	3.52E−06
	STD	3.54E−01	2.67E−05	4.67E−01	5.52E + 03	2.55E−02	1.56E−01	1.06E + 00	8.79E−03	4.92E−01	4.78E−03	8.47E−03
F13	AF	7.31E−04	1.99E−04	1.07E + 00	6.25E + 00	6.81E−02	1.00E−01	**3.65E−11**	1.45E−01	1.10E−02	2.00E−01	2.16E−06
	STD	7.38E−02	1.46E−04	1.52E−01	3.48E + 03	5.25E + 00	2.30E−01	5.31E−03	8.66E−02	5.67E−03	8.33E−03	2.82E−06

random host volume space. During the feeding of remora on the host, the search space can be seen as gradually decreasing. For ease of understanding, the above novel concepts and pseudo code of different location update modes are given in Table 3. The flow chart is shown in Fig. 6.

In the third chapter, various features of this algorithm are explored through experiments.

2.3. Computational complexity

In order to analyze the validity and practicability of the algorithm in more detail, the computational complexity of the algorithm is given according to the related factors such as the number, dimension and maximum iteration times of remora. It is worth noting that there is no sort algorithm. In the initialization process the computational complexity is $O(N)$. The computational complexity of the fitness function can be expressed as $O(F(R))$. Computational complexity expressed as $O(N \times T \times D)$ for Free travelling stage and can also be expressed as $O(N \times T \times D)$ for Eat thoughtfully stage. The overall computational complexity of the whole algorithm can be expressed as following formulas:

$$\begin{aligned} O(ROA) &= O(fitness\_function) \times \left(O\left(T \times \begin{pmatrix} O(initialization)+ \\ O(Free\ travelling)+ \\ O(Eat\ mindfully) \end{pmatrix} \right) \right) \\ &= O(F(R)) \times \left(O\left(\begin{pmatrix} O(N)+ \\ O(N \times T \times D)+ \\ O(N \times T \times D) \end{pmatrix} \right) \right) \\ &= O(F(R)) \times O(N \times (TD+1)) \end{aligned} \tag{13}$$

3. Results and discussion

3.1. Experimental Setup

All the experimental series were carried out on MATLAB R2017b (The Math Works Inc., Natick, MA, USA), and the computer was configured as Intel(R) Pentium (R) CPU G4560 @3.50 GHz (Intel, Santa Clara, CA, USA), using Microsoft Windows 7 system (Microsoft, Redmond, WA, USA).

In this section, in order to study the numerical efficiency performance of the proposed ROA algorithm, a set of 29 unconstrained

Table 13
Results of benchmark functions (F1 - F13) under 100 dimension.

Benchmark		ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	AF	0.00E + 00	4.10E−106	2.31E−03	2.73E + 00	5.10E−02	9.88E−100	1.48E−07	1.98E−70	9.72E−01	8.80E−10	5.29E−28
	STD	0.00E + 00	1.81E−92	2.01E−01	1.84E + 01	8.59E−02	1.20E + 00	9.85E−07	4.59E−68	2.58E + 03	1.00E + 00	1.14E−27
F2	AF	**0.00E + 00**	5.67E−53	3.15E−01	1.42E−34	2.67E−03	1.82E−171	1.62E + 00	4.07E−51	3.01E + 01	7.44E−15	5.48E−17
	STD	0.00E + 00	1.31E−50	1.17E−01	2.24E−06	1.33E−02	2.36E−160	1.09E + 00	1.80E−47	2.16E + 01	2.13E−20	1.13E−16
F3	AF	**1.84E−309**	**1.44E−84**	5.01E−01	8.51E + 04	6.37E + 03	2.02E + 02	1.75E + 02	5.39E + 04	3.94E + 04	4.40E−20	4.09E−06
	STD	0.00E + 00	1.30E−79	3.29E + 00	1.69E + 04	1.26E + 04	3.62E + 02	8.51E + 02	1.45E + 04	9.55E + 03	0.00E + 00	3.94E−06
F4	AF	**1.21E−171**	1.10E−55	2.34E−01	3.61E + 01	6.89E−01	1.00E + 00	7.88E + 00	8.10E + 01	6.52E + 01	1.00E + 00	7.02E−07
	STD	0.00E + 00	1.09E−48	2.00E−01	1.63E + 01	7.66E−01	1.10E + 00	2.71E + 00	2.33E + 01	8.44E + 00	1.75E + 00	5.82E−07
F5	AF	2.70E + 01	**3.41E−03**	2.89E + 01	1.65E + 04	9.12E + 00	2.90E + 01	2.93E + 03	2.83E + 01	2.29E + 03	2.29E + 02	2.62E + 01
	STD	4.59E−01	4.35E−03	5.40E + 00	2.26E + 05	3.46E + 00	1.02E + 01	7.68E + 02	4.56E−01	2.06E + 07	1.03E + 02	8.32E−01
F6	AF	3.43E−02	1.81E−03	7.43E + 00	5.27E + 01	1.70E−01	7.50E + 00	**1.46E−07**	2.48E + 00	2.35E + 00	7.50E + 00	5.05E−01
	STD	8.11E−05	4.81E−04	8.30E−01	2.49E + 01	7.66E−02	2.98E + 00	3.08E−07	6.46E−01	4.86E + 03	7.46E + 00	4.70E−01
F7	AF	**1.13E−05**	2.64E−05	7.49E−04	6.90E−02	1.87E−02	2.48E−05	2.17E−01	6.93E−04	2.79E + 00	7.86E−05	5.11E−03
	STD	5.68E−05	6.48E−05	3.20E−03	3.35E−01	3.25E + 03	1.29E−04	6.51E−02	7.16E−03	5.93E + 00	6.90E−05	1.32E−03
F8	AF	**2.61E + 03**	6.16E + 34	1.26E + 04	1.12E + 04	1.21E + 04	1.25E + 04	4.72E + 04	1.50E + 104	8.30E + 03	6.17E + 03	2.52E + 34
	STD	4.74E + 01	1.27E + 45	3.61E + 02	8.83E + 02	1.36E + 02	1.13E + 03	1.01E + 04	4.80E + 103	9.55E + 02	6.16E + 02	9.07E + 34
F9	AF	**0.00E + 00**	**0.00E + 00**	9.96E−03	3.26E−02	5.60E + 00	1.00E + 00	4.97E + 01	2.30E + 00	1.63E + 02	4.53E + 01	2.12E + 00
	STD	0.00E + 00	0.00E + 00	9.86E−01	3.96E + 01	4.56E + 02	0.00E + 00	1.81E + 01	0.00E + 00	1.77E + 01	0.00E + 00	2.20E + 00
F10	AF	**8.88E−16**	**8.88E−16**	4.13E−02	6.21E−06	8.90E−03	8.99E−16	1.78E + 00	4.44E−15	1.99E + 01	9.88E−16	1.00E−13
	STD	0.00E + 00	0.00E + 00	2.09E−01	3.58E + 00	5.13E + 00	0.00E + 00	4.80E−01	2.40E−15	6.92E + 00	0.00E + 00	1.21E−14
F11	AF	**0.00E + 00**	**0.00E + 00**	1.10E−01	5.72E−02	6.10E + 00	2.00E−01	3.03E−03	3.50E−01	9.01E−01	1.64E−01	9.60E−03
	STD	0.00E + 00	0.00E + 00	3.08E−01	2.89E−01	9.12E + 00	0.00E + 00	1.15E−02	3.36E−02	5.34E + 01	1.54E−01	9.74E−03
F12	AF	1.16E−03	**8.26E−06**	1.04E + 00	4.05E + 01	6.07E−03	6.94E−01	4.61E + 00	5.39E−02	5.29E + 00	7.13E−01	3.99E−02
	STD	7.21E−03	7.52E−06	2.42E−01	2.40E + 04	7.90E−01	3.52E−01	5.35E + 00	9.05E−02	5.46E + 00	1.63E−01	1.80E−02
F13	AF	2.40E−02	**3.50E−08**	3.09E + 00	5.40E + 00	2.14E + 01	2.60E + 00	2.15E + 00	5.74E−01	2.09E + 01	1.60E + 00	5.43E−01
	STD	8.76E−02	1.27E−04	2.32E−01	1.47E + 05	1.48E + 04	7.16E−01	1.45E + 01	4.74E−01	1.06E + 08	1.87E−01	2.61E−01

functions are taken from the classic benchmark functions CEC2005 (Suganthan, Hansen, Liang, & etc., 2005). Which can be divided into three main types:

- Unimodal (UM) functions: These functions are used to assess the solution precision and the convergence rate of the proposed algorithm. Which can be used to validate the exploitation (hardening) capabilities of different optimizers. As shown in Table 4 is (F1–F7).
- Multi-modal (MM) functions: These functions can be used to verify the potential of the algorithm to exploration (diversify) and avoid falling into local optimum which are from F8 to F23 in Tables 5 and 6.
- Composition (CM) functions: Many papers have used these CM cases for evaluation, which can well test the propensity of balancing exploration and development, and can also be used as a challenging problem to evaluate the ability to avoid local optimum, thus showing the performance of the proposed optimizer. Reference (Marco et al., 1996) for details. Which can be found in Table 7 (F24 to F29).

The functional composition, dimension, range limitation and optimal position of functions has been given in each table. It is noteworthy that Dim represents the number of variables designed for mathematical functions. Some typical two-dimensional diagrams of test functions for test cases are shown in Tables 4–7, the details of hybrid composition function are also given in Fig. 7.

In order to verify the performance of the proposed algorithm, the proposed ROA is compared with 10 other state of the art optimization algorithms according to the test function results. The details of the comparison algorithm are as follows:

- Harris hawks optimization (HHO): In 2019, a new population-based and nature-based optimization model. The main inspiration comes from Harris hawks' cooperative behavior and pursuit style. According to the dynamic characteristics of the scene and the escape mode of prey, a variety of chase modes are proposed. The whole algorithm simulates the dynamic model and behavior in a comprehensive and vivid way.
- Sailfish optimizer (SFO): A new natural inspired heuristic optimization algorithm in 2019, which consists of two populations, sailfish is used to enhance the best search so far, sardines are used to scatter search space.
- Emperor penguin optimizer (EPO): Proposed in 2018 to simulate the crowding behavior of Emperor Penguin. The main steps are to generate the crowding boundary, calculate the temperature around the crowding pile, calculate the distance and find the effective mover.
- Seagull optimization algorithm (SOA): A bionic algorithm published in 2018, mathematically simulates the migration and attack of seagull nature, which strengthens the exploration and development in a given search space.
- Spotted hyena optimizer (SHO): A new *meta*-heuristic algorithm based on the behavioral characteristics of spotted hyenas was proposed in 2017. The main concept is the social relationship between spotted hyenas and their cooperative behavior. The three basic steps of SHO are to search for prey, surround and attack prey.
- Salp Swarm Algorithm (SSA): The main inspiration comes from the social behavior of salps when sailing and foraging in the ocean, which can effectively improve the initial stochastic solution and converge to the optimal solution quickly. (2017)
- Whale optimization algorithm (WOA): Simulates the social behavior of humpback whale, and the search strategy of foam net is the core of the algorithm. So far, it can still be used as a reference algorithm for comparison with many other algorithms, which enough to see its excellence. (2016)
- Moth-flame optimization algorithm (MFO): Inspired by the moth's lateral orientation navigation method. When flying at night, moths fly long distances in a straight line by keeping a fixed angle with the moon, and eventually run on a spiral path (2015).
- Multiverse optimizer (MVO): Through the concepts of white hole, black hole and wormhole in cosmology. Mathematical models are established for exploration, development and local search, respectively. Location updating is mainly through roulette mechanism (2015).

Table 14
Results of benchmark functions (F1 - F13) under 500 dimension.

Benchmark		ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	AF	**0.00E + 00**	4.43E−98	2.55E−01	4.92E−02	5.06E−01	5.05E−52	1.44E + 03	1.95E−60	4.78E + 04	1.51E−12	4.95E−13
	STD	0.00E + 00	1.15E−95	4.06E + 00	8.16E + 01	8.74E + 02	4.12E + 00	4.83E + 02	3.56E−61	1.66E + 04	1.21E−01	1.71E−12
F2	AF	**6.15E−172**	3.27E−53	6.12E−01	5.89E−07	9.01E−04	1.50E−72	5.67E + 01	1.42E−54	3.46E + 02	4.75E−17	3.44E−08
	STD	4.45E−162	4.20E−50	8.13E−01	3.22E−05	9.93E−03	2.50E−02	5.85E + 00	5.31E−44	3.91E + 01	1.80E−02	2.63E−08
F3	AF	**1.24E−299**	1.56E−70	3.57E + 01	1.04E + 06	3.51E + 05	1.62E + 03	9.29E + 04	7.56E + 05	3.23E + 05	1.00E + 02	1.99E + 03
	STD	0.00E + 00	1.26E−55	1.76E + 01	2.29E + 05	3.81E + 06	1.13E + 04	2.55E + 04	2.96E + 05	6.37E + 04	8.22E + 02	1.62E + 03
F4	AF	**0.00E + 00**	2.03E−54	2.19E−01	5.45E + 01	3.59E−01	2.00E−164	2.99E + 01	5.03E + 01	9.35E + 01	7.40E + 01	8.35E−01
	STD	4.80E−161	1.53E−47	2.36E−01	1.78E + 01	7.19E−01	2.53E + 00	3.59E + 00	2.44E + 01	1.97E + 00	9.35E + 00	7.96E−01
F5	AF	9.72E + 01	**8.14E−04**	1.17E + 02	3.58E + 06	2.64E + 01	9.90E + 01	3.50E + 05	9.82E + 01	1.66E + 08	1.87E + 02	9.58E + 01
	STD	4.38E−01	6.97E−02	3.18E + 01	3.21E + 06	3.02E + 10	1.81E + 01	7.11E + 04	1.36E−01	6.18E + 07	1.71E + 02	7.54E−01
F6	AF	1.58E + 00	**2.40E−04**	2.64E + 01	1.39E + 01	6.30E−01	2.50E + 01	1.52E + 03	1.20E + 01	7.55E + 04	2.50E + 01	1.11E + 01
	STD	6.72E−01	5.39E−04	2.22E + 00	1.26E + 02	1.30E + 00	0.00E + 00	3.74E + 02	2.72E + 00	1.42E + 04	0.00E + 00	9.43E−01
F7	AF	**3.64E−05**	5.26E−04	6.60E−04	3.29E−01	6.01E−04	6.33E−05	3.26E + 00	2.62E−03	4.30E + 02	1.86E−04	9.27E−03
	STD	7.72E−05	2.28E−04	4.76E−03	3.91E + 00	4.91E + 03	3.48E−04	6.86E−01	2.93E−03	1.01E + 02	1.17E−04	2.38E−03
F8	AF	**4.49E + 03**	2.85E + 31	4.19E + 04	4.13E + 04	4.19E + 04	4.13E + 04	1.78E + 05	1.52E + 101	2.18E + 04	1.87E + 04	4.88E + 30
	STD	1.72E + 02	8.25E + 43	6.17E + 02	3.71E + 03	6.38E + 02	4.36E + 03	2.81E + 04	2.03E + 105	2.25E + 03	1.12E + 03	1.29E + 35
F9	AF	**0.00E + 00**	**0.00E + 00**	3.38E−01	1.59E−07	1.58E + 294	1.00E + 00	2.36E + 02	**0.00E + 00**	9.23E + 02	1.00E + 00	1.04E + 00
	STD	0.00E + 00	0.00E + 00	2.31E + 00	4.79E + 00	1.01E + 00	1.02E + 01	4.09E + 01	2.08E−14	6.60E + 01	1.86E + 01	2.92E + 01
F10	AF	**8.88E−16**	**8.88E−16**	1.02E−01	6.39E−02	1.76E−04	8.90E−16	9.56E + 00	8.90E−14	1.97E + 01	8.90E−15	1.76E−07
	STD	0.00E + 00	1.00E−26	1.60E−01	3.09E + 00	1.40E + 00	2.56E + 00	1.20E + 00	3.58E−15	1.14E−01	9.80E−05	5.66E−08
F11	AF	**0.00E + 00**	**0.00E + 00**	7.04E−01	5.45E−08	4.79E−01	4.06E−100	1.53E + 01	5.20E−160	8.25E + 02	4.87E−01	2.93E−02
	STD	0.00E + 00	0.00E + 00	3.11E−01	3.66E−01	3.57E + 00	0.00E + 00	4.29E + 00	2.03E−17	1.25E + 02	8.20E−02	9.19E−03
F12	AF	7.51E−03	**2.40E−06**	1.26E + 00	5.55E + 05	1.78E−03	7.26E−01	3.85E + 01	1.25E−01	4.62E + 08	1.10E + 00	3.58E−01
	STD	1.06E−08	7.90E−06	6.30E−02	4.65E + 05	4.42E + 01	3.18E−01	1.14E + 01	1.03E−01	2.04E + 08	1.75E−01	8.02E−02
F13	AF	6.03E−01	**5.09E−07**	1.15E + 01	2.24E + 06	3.75E + 04	8.30E + 00	9.19E + 02	7.26E + 00	8.47E + 08	7.90E + 00	7.36E + 00
	STD	6.29E−05	3.33E−04	1.18E + 00	6.84E + 06	6.47E + 02	2.10E + 00	6.03E + 03	1.30E + 00	4.29E + 08	3.26E−01	5.10E−01

- Grey Wolf Optimizer (GWO): According to the leadership and hunting mechanism of gray wolves in nature. Four gray wolves, Alpha, Beta, Delta and Omega, were used to simulate leadership. The establishment of three main steps of hunting, sticking and attacking prey has been realized (2014).

Eleven comparison algorithms correspond to different algorithms from 2019 to 2014, and the selection of these algorithms is novel or well received. The parameters of these algorithms are selected from the references related to the original algorithms, as shown in Table 8. All algorithms use a population size and maximum iteration of 30 and 500, respectively.

3.2. Experimental results indicators

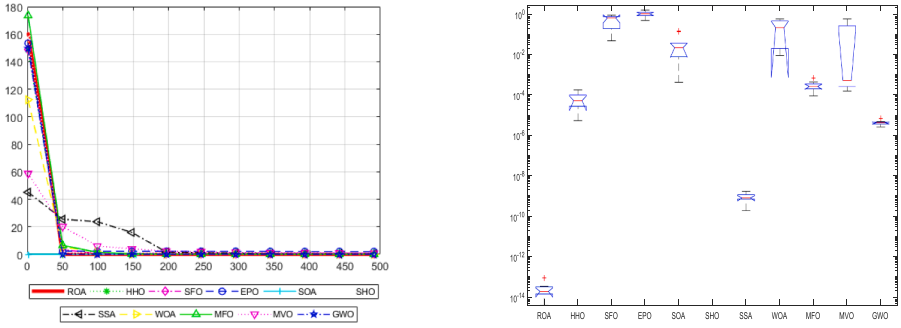
In order to clearly and vividly demonstrate the experimental results and performance of HHO, the following indicators in Table 9 are used as criteria to compare with other optimization techniques.

Generally speaking, these indicators can be divided into two categories. One is called performance evaluation, for instance, AT, AF and Std, which are mainly based on individual algorithms. AT represents the average execution time of each algorithm running independently for 30 times. The smaller the numerical value, the faster the execution speed and the lower the computational complexity of the algorithm. In order to make the results of the function more fair and universal, the AF and Std can also be used as reference standards for comparison.

The other type is called statistical test, such as, the Wilcoxon's Rank-Sum test and Friedman test, which are mainly based on the overall analysis of large data sets. WT is used to answer the question "Does two samples represent two different populations?" $p > 0.05$ (or $h = 1$), there is a significant difference, otherwise it is not. FT is a nonparametric simulation of nonparametric variance bidirectional analysis. Used to answer the question "Does at least two samples in a group of k samples represent populations with different median values?" Designed to detect significant differences between the behavior of two or more algorithms, the overall performance of the algorithm can be ranked. Which has also

Table 15
Results of benchmark functions (F1 - F13) under 1000 dimension.

Benchmark		ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	AF	**0.00E + 00**	2.90E−98	3.50E + 00	4.05E + 02	2.25E−02	1.59E−10	9.22E + 04	1.04E−68	1.12E + 06	6.56E−06	5.79E−03
	STD	0.00E + 00	1.85E−96	4.11E + 01	2.85E + 02	2.01E + 01	8.42E + 03	5.18E + 03	1.30E−61	3.75E + 04	4.70E−03	3.26E−03
F2	AF	**2.12E−186**	1.31E−53	1.17E + 00	3.52E−05	2.73E−01	2.54E−08	5.35E + 02	2.58E−47	8.22E + 99	2.24E−02	1.84E−02
	STD	0.00E + 00	3.74E−48	3.10E + 00	5.34E−04	1.33E−01	1.32E−01	1.89E + 01	9.77E−43	3.79E + 108	7.48E + 00	5.07E−03
F3	AF	**6.37E−284**	1.40E−65	2.45E + 02	1.86E + 07	5.50E + 06	5.33E + 05	1.49E + 06	2.98E + 07	4.83E + 06	1.19E + 06	3.26E + 06
	STD	0.00E + 00	5.34E−29	8.19E + 02	8.22E + 06	8.60E + 08	1.32E + 06	8.35E + 05	8.93E + 06	8.31E + 05	4.58E + 05	9.18E + 05
F4	AF	**1.14E−165**	1.20E−54	2.75E−01	5.13E + 01	7.29E + 00	3.00E + 00	4.07E + 01	9.63E + 01	9.84E + 01	9.40E + 01	6.85E + 01
	STD	5.44E−162	7.67E−46	2.37E−01	1.79E + 01	1.78E + 00	1.70E + 00	2.56E + 00	1.26E + 01	3.72E−01	1.97E + 00	5.79E + 00
F5	AF	4.94E + 02	**1.15E−01**	5.23E + 02	3.65E + 04	2.35E + 00	4.99E + 02	3.76E + 07	4.97E + 02	5.24E + 09	5.89E + 02	1.43E + 03
	STD	2.21E−01	3.08E−01	7.18E + 01	2.83E + 07	5.45E + 10	0.00E + 00	5.83E + 06	4.56E−01	2.69E + 08	1.03E + 02	3.70E + 02
F6	AF	2.04E + 01	**7.67E−04**	1.25E + 02	3.28E + 02	1.20E + 00	1.25E + 02	9.68E + 04	8.54E + 01	1.16E + 06	1.25E + 02	8.88E + 01
	STD	6.55E + 00	3.80E−03	7.83E + 00	8.37E + 01	1.65E + 01	0.00E + 00	5.62E + 03	1.04E + 01	3.38E + 04	0.00E + 00	2.49E + 00
F7	AF	**5.35E−05**	1.33E−04	2.49E−02	3.19E−02	3.15E−02	1.06E−04	2.45E + 02	5.37E−05	3.94E + 04	3.04E−04	1.95E−01
	STD	1.46E−04	1.74E−04	1.76E−02	1.22E + 02	1.45E + 02	4.99E−05	3.69E + 01	4.43E−03	1.81E + 03	7.00E−05	6.59E−02
F8	AF	**1.02E + 04**	5.49E + 33	2.09E + 05	1.49E + 05	2.09E + 05	2.09E + 05	3.99E + 05	1.57E + 104	6.17E + 04	5.38E + 04	4.93E + 34
	STD	5.27E + 03	4.14E + 40	1.76E + 03	2.69E + 04	1.85E + 01	2.85E + 04	6.00E + 04	7.82E + 105	5.34E + 03	2.43E + 03	2.40E + 34
F9	AF	**0.00E + 00**	**0.00E + 00**	2.43E−02	1.27E + 02	5.15E−01	5.13E + 00	3.15E + 03	4.58E + 02	6.87E + 03	8.00E + 03	1.25E + 02
	STD	0.00E + 00	0.00E + 00	5.44E + 00	2.11E + 02	7.84E + 01	4.56E + 02	1.18E + 02	6.87E + 03	1.60E + 02	2.38E + 03	3.65E + 01
F10	AF	**8.88E−16**	**8.88E−16**	2.59E−01	6.01E−03	7.14E + 00	9.88E−15	1.40E + 01	7.99E−15	2.03E + 01	8.98E−15	7.44E−03
	STD	0.00E + 00	0.00E + 00	2.02E−01	2.18E + 00	1.99E−03	1.66E−01	2.63E−01	2.63E−15	1.11E−01	6.55E−02	1.94E−03
F11	AF	**0.00E + 00**	**0.00E + 00**	5.89E−03	1.65E + 00	5.63E + 03	6.00E + 00	7.97E + 02	9.75E + 03	9.96E + 03	2.09E + 03	1.45E−03
	STD	0.00E + 00	0.00E + 00	4.31E−01	5.51E−01	5.56E−01	4.17E + 00	4.86E + 01	1.36E + 01	3.41E + 02	8.91E + 01	4.16E−02
F12	AF	4.81E−04	**1.84E−07**	1.20E + 00	1.58E + 03	2.84E−02	8.90E−01	1.61E + 06	1.83E−01	1.11E + 10	1.16E + 00	1.82E + 00
	STD	3.38E−02	4.07E−06	2.94E−02	6.76E + 06	5.52E + 02	2.96E−01	8.21E + 05	1.11E−01	8.05E + 08	1.34E−02	1.83E + 00
F13	AF	8.32E + 00	**6.91E−04**	7.21E + 01	2.34E + 05	7.90E + 04	2.39E + 01	3.82E + 07	2.50E + 01	2.13E + 10	4.85E + 01	1.02E + 02
	STD	3.72E + 00	6.35E−04	7.21E + 00	1.91E + 07	5.45E + 04	1.33E + 01	8.60E + 06	8.84E + 00	1.14E + 09	2.61E−01	3.94E + 01



F11 Box diagram

Fig. 9. The trend chart of fitness function value of function 1–13 and box diagram.

been used in experiments to determine the parameter C.

3.3. Independent evaluation of algorithm performance

3.3.1. Qualitative analysis

The basic performance of ROA was measured primarily. Several standard single-mode and multi-mode test problems are evaluated. The results of three basic data evaluation: Search history, Track of the first remora, and Convergence effects are shown in Fig. 8.

- Search history: The first two dimensions of the search agent are selected, and the corresponding two-dimensional scatter plot is made by iteration, which shows the changing trend of particles in the plane image. It can be seen that the initial particle first fill the solution space within a limited range, then converge slowly from all directions, and finally converge to the optimal solution. Through observation and analysis, it can be seen that the particles are dispersed in the whole solution space in the process of searching for simple functions, with uniform distribution of scatter points in the early stage and rapid concentration in the later stage. In the early and late stages of the optimization process of complex functions, the density of scattering points is obviously different. This shows that when dealing with different problems, the algorithm can change the balance between exploration and exploitation.
- Track of the first remora:Taking the trajectory of the first Remola as an example, the characteristics of the algorithm are further examined. It can be seen from the trajectory diagram in Fig. 8 that there is a large sudden change in the early optimization stage. The magnitude of the change basically covers the entire solution space, can make a big jump, does not fall into the early local optimum, reveals that ROA has better exploration ability. In the late optimization stage, the simple function converges directly, and the complex

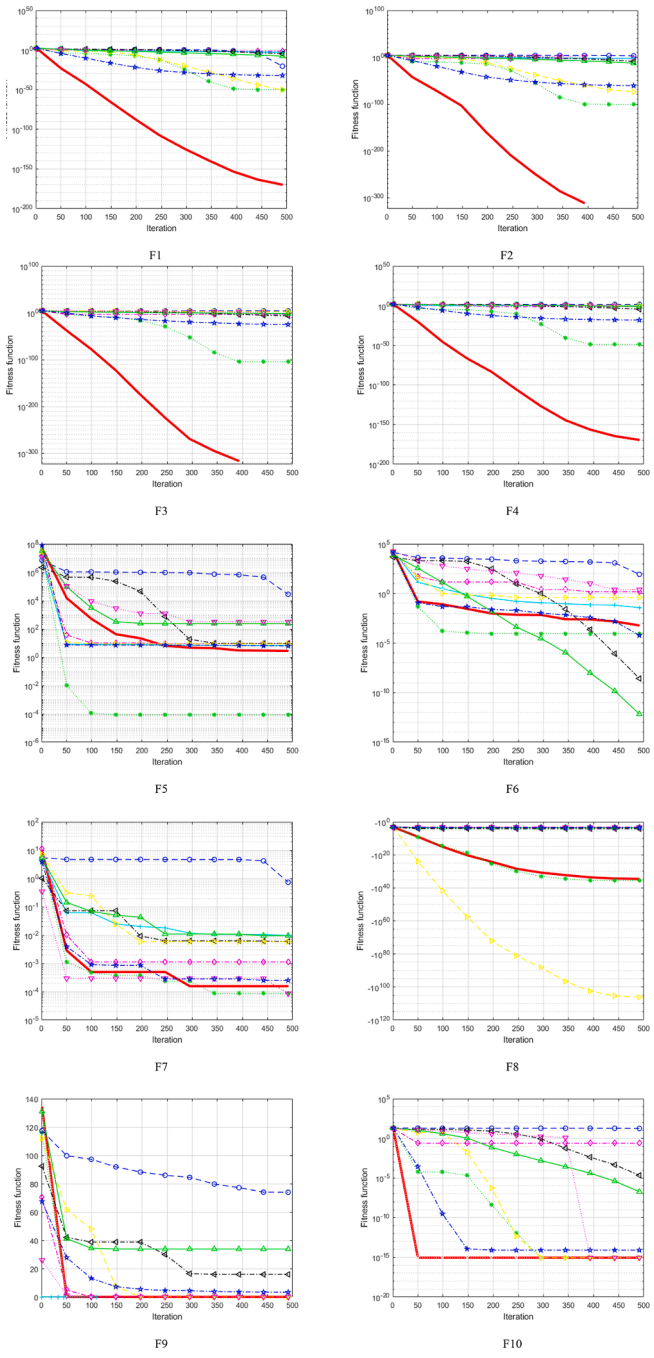


Fig. 9. (*continued*).

function gradually changes. This proves that the update of the population continues, and also corresponds to the process from exploration to development. As the number of iterations increases, the magnitude of these fluctuations is gradually reduced, as shown by the trajectories of F7 and F13 in Fig. 8. At the end of the search for excellence, Remora's movement tends to be stable and converges to the optimal solution.

- Convergence metric:The convergence curve shows the trend of the fitness function value. Simple functions such as F1, F3, and F4 have a smoother fitness function curve that lasts 500 iterations. Each generation can effectively find the best, and finally find the exact minimum value of 0. More complex functions such as F7, F10, and F13 are stepwise updated, indicating the existence of stagnation between populations. However, there are still fluctuations in the curve at the late stage of convergence, indicating that the diversity between populations still exists.

3.3.2. *Scalability analysis*

A new type of algorithm needs to be comprehensively analyzed. In the literature, scalability is an important indicator. It mainly compares the conclusions by changing the dimensions of the test function, and effectively judges the influence of the dimension expansion on the execution efficiency of the algorithm. According to experience, the F1 and F13 of UM and MM were tested in four dimensions of 30, 100, 500 and 1000. The AF and STD indicators were selected, and the results of independent operation 30 times and 500 iterations were made in Table 10. Use FT to further make a simple statistical analysis of the data in Table 10, as shown in Table 11. It can be seen from the data

Table 16
Results of benchmark functions (F14 - F29).

Benchmark		ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F14	AF	**9.98E−01**	**9.98E−01**	2.85E + 00	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	5.93E + 00	**9.98E−01**	**9.98E−01**	2.98E + 00
	STD	0.00E + 00	3.14E−01	3.19E + 00	3.67E + 00	1.04E + 04	3.22E + 00	1.96E−16	8.43E−01	2.33E + 00	5.42E + 00	4.35E + 00
F15	AF	**3.11E−04**	3.32E−04	8.88E−04	2.82E−02	2.25E−03	1.48E−01	1.08E−03	5.74E−04	7.83E−04	1.48E−01	1.64E−03
	STD	1.74E−02	3.46E−05	1.04E−03	1.32E−02	5.27E + 17	5.43E−02	5.85E−04	6.21E−04	4.61E−04	1.62E−04	3.61E−04
F16	AF	**−1.03E + 00**	**−1.03E + 00**	−1.02E + 00	**−1.03E + 00**	**−1.03E + 00**	0.00E + 00	**−1.03E + 00**	**−1.03E + 00**	**−1.03E + 00**	0.00E + 00	**−1.03E + 00**
	STD	0.00E + 00	5.91E−11	1.05E−02	4.54E−01	2.70E−07	2.93E−01	2.24E−14	2.02E−10	0.00E + 00	5.77E−08	7.67E−08
F17	AF	**3.98E−01**	**3.98E−01**	1.76E + 00	4.61E−01	**3.98E−01**	6.45E−01	**3.98E−01**	**3.98E−01**	**3.98E−01**	4.98E−01	**3.98E−01**
	STD	0.00E + 00	4.12E−05	1.40E−01	2.49E−01	9.38E + 11	7.08E−02	2.70E−14	2.27E−05	0.00E + 00	2.73E−05	3.02E−06
F18	AF	**3.00E + 00**	**3.00E + 00**	7.00E + 00	3.20E + 01	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**	2.78E + 02	**3.00E + 00**
	STD	0.00E + 00	1.64E−06	3.03E + 01	1.92E + 00	9.55E + 10	0.00E + 00	2.33E−13	1.40E−04	1.61E−15	2.86E−05	2.38E−05
F19	AF	**−3.86E + 00**	**−3.86E + 00**	−3.78E + 00	−3.79E + 00	−3.55E + 00	−3.66E + 00	**−3.86E + 00**	**−3.86E + 00**	**−3.86E + 00**	−6.80E−02	**−3.86E + 00**
	STD	1.46E−17	2.98E−03	1.81E−01	2.78E−01	2.35E−01	3.61E−01	1.79E−12	1.17E−02	9.36E−16	1.20E−04	5.11E−04
F20	AF	**−3.20E + 00**	−3.09E + 00	−1.94E + 00	−1.63E + 00	−2.75E + 00	−1.15E + 00	−3.18E + 00	−3.19E + 00	−3.19E + 00	−5.11E−E−03	**−3.20E + 00**
	STD	9.14E−19	7.26E−02	2.80E−01	5.41E−01	5.39E−01	3.57E−01	6.51E−02	5.14E−02	6.19E−02	6.01E−02	5.89E−02
F21	AF	**−1.01E + 01**	−1.02E + 01	−1.33E + 00	−5.63E + 00	**−1.01E + 01**	−1.02E + 01	−2.63E + 00	−5.06E + 00	−2.68E + 00	−1.02E + 01	−1.02E + 01
	STD	2.63E + 00	3.98E−02	9.76E−01	1.44E + 00	2.25E−01	2.93E + 00	3.29E + 00	2.74E + 00	3.76E + 00	1.01E−02	1.08E−03
F22	AF	**−1.04E + 01**	**−1.04E + 01**	−2.76E + 00	−8.83E−01	−9.30E + 00	**−1.04E + 01**	−2.75E + 00	−2.76E + 00	**−1.04E + 01**	−5.13E + 00	**−1.04E + 01**
	STD	2.56E + 00	3.25E−02	1.06E + 00	1.43E + 00	1.21E−01	3.00E + 00	2.55E + 00	3.91E + 00	3.54E + 00	1.32E−02	8.89E−04
F23	AF	**−1.05E + 01**	**−1.05E + 01**	−2.15E + 00	−4.26E + 00	−1.01E + 01	**−1.05E + 01**	−2.43E + 00	−2.42E + 00	**−1.05E + 01**	**−1.05E + 01**	**−1.05E + 01**
	STD	2.79E + 00	3.22E−02	8.14E−01	1.40E + 00	2.31E−01	3.03E + 00	3.53E + 00	3.56E + 00	3.46E + 00	3.83E−02	1.00E−03
F24	AF	**2.29E + 02**	4.36E + 02	1.06E + 03	1.20E + 03	2.86E + 02	1.17E + 03	4.01E + 02	5.86E + 02	4.87E + 02	1.63E + 03	6.51E + 02
	STD	1.45E + 01	1.27E + 02	1.79E + 02	1.87E + 02	1.74E + 01	1.40E + 02	1.48E + 02	7.14E + 01	1.07E + 02	1.90E + 02	1.59E + 02
F25	AF	**9.00E + 02**	9.10E + 02	9.10E + 02	1.44E + 03	2.95E + 03	9.10E + 02	9.67E + 02	1.12E + 03	9.73E + 02	9.10E + 02	9.03E + 02
	STD	0.00E + 00	0.00E + 00	1.72E−01	2.99E + 01	2.12E + 01	0.00E + 00	1.22E + 01	1.34E + 02	2.46E + 01	0.00E + 00	4.05E + 01
F26	AF	**9.00E + 02**	9.10E + 02	9.10E + 02	1.38E + 03	9.10E + 02	9.10E + 02	9.65E + 02	1.12E + 03	9.18E + 02	9.10E + 02	9.38E + 02
	STD	0.00E + 00	0.00E + 00	4.69E−01	5.26E + 01	5.95E + 01	0.00E + 00	1.32E + 01	9.55E + 01	3.08E + 01	0.00E + 00	4.25E + 01
F27	AF	**9.00E + 02**	9.10E + 02	9.10E + 02	1.45E + 03	2.89E + 03	9.10E + 02	9.43E + 02	1.30E + 03	9.66E + 02	9.10E + 02	9.92E + 02
	STD	0.00E + 00	0.00E + 00	2.46E−01	4.44E + 01	7.98E + 01	0.00E + 00	2.05E + 01	8.97E + 01	2.85E + 01	0.00E + 00	6.22E + 01
F28	AF	**9.00E + 02**	9.28E + 02	1.43E + 03	1.48E + 03	1.43E + 03	1.43E + 03	**9.00E + 02**	1.39E + 03	1.10E + 03	1.40E + 03	1.20E + 03
	STD	5.32E + 00	1.11E + 02	6.30E + 00	4.16E + 01	2.18E + 02	2.27E + 01	1.58E + 02	9.14E + 01	9.16E + 00	3.01E + 01	1.16E + 02
F29	AF	**9.59E + 02**	1.21E + 03	1.42E + 03	1.45E + 03	1.43E + 03	1.31E + 03	1.16E + 03	1.30E + 03	1.11E + 03	1.41E + 03	1.33E + 03
	STD	5.25E + 00	1.54E + 02	1.41E + 01	3.22E + 01	5.12E + 02	2.85E + 01	2.51E + 02	2.62E + 01	1.35E + 01	5.46E + 01	1.01E + 02

comparison that as the dimension increases, the performance of the algorithm optimization gradually decreases, which is consistent with the results obtained by the FT statistical test. However, as can be seen from the ranking ratio that although the dimension increase is large, the increase in the ranking is small. It shows that ROA can maintain good exploratoryness under the condition of exploiting certain optimality when dealing with high-dimensional problems.

3.4. Comparative evaluation of algorithm performance

After exploring the characteristics of the ROA itself, further comparison with the previously selected algorithm. Mainly for three indicators: fitness function value, algorithm running time and statistical analysis.

- Fitness function value

Tables 12–15 shows that the 11 algorithms independently run the

Table 17
The running time of the benchmark function (F1–F29).

Benchmark		ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	AT	2.20E−01	1.49E−01	6.86E−02	1.83E−01	6.53E−01	7.60E−02	6.29E−02	**4.35E−02**	9.38E−02	5.56E−01	6.15E−02
	STD	6.48E−03	2.82E−02	1.98E−02	1.41E−02	7.56E−02	1.39E−02	1.07E−02	8.85E−03	1.34E−02	8.23E−02	1.24E−02
F2	AT	2.87E−01	1.75E−01	8.91E−02	2.00E−01	6.66E−01	9.63E−02	7.92E−02	**5.98E−02**	1.13E−01	5.53E−01	7.75E−02
	STD	1.84E−04	1.64E−03	1.40E−03	3.11E−04	3.22E−02	5.18E−04	1.11E−03	3.51E−04	9.80E−04	1.81E−03	4.58E−04
F3	AT	7.11E−01	4.94E−01	2.16E−01	3.29E−01	7.47E−01	2.29E−01	2.05E−01	**1.85E−01**	2.39E−01	7.01E−01	2.01E−01
	STD	3.18E−04	9.88E−03	4.64E−04	8.48E−04	1.34E−02	6.31E−04	1.92E−03	6.24E−04	7.44E−04	4.21E−03	3.88E−03
F4	AT	2.30E−01	1.90E−01	6.98E−02	1.80E−01	6.39E−01	7.71E−02	6.67E−02	**4.41E−02**	9.56E−02	5.62E−01	5.98E−02
	STD	1.98E−04	1.94E−03	1.24E−03	8.74E−04	5.78E−03	3.80E−03	9.19E−04	3.79E−04	1.67E−03	2.19E−03	1.06E−03
F5	AT	3.22E−01	2.86E−01	9.58E−02	2.12E−01	6.68E−01	1.05E−01	8.92E−02	**6.89E−02**	1.23E−01	5.77E−01	8.55E−02
	STD	2.14E−04	2.84E−03	4.94E−04	3.50E−04	2.76E−02	1.06E−03	6.13E−04	1.06E−03	1.51E−03	1.68E−03	6.77E−04
F6	AT	2.29E−01	2.16E−01	6.92E−02	1.81E−01	6.59E−01	7.72E−02	6.35E−02	**4.32E−02**	9.51E−02	5.56E−01	6.03E−02
	STD	1.17E−04	1.64E−03	8.93E−04	8.27E−04	3.61E−02	1.01E−03	1.08E−03	9.36E−04	1.30E−03	2.16E−03	7.36E−04
F7	AT	3.70E−01	2.60E−01	1.07E−01	2.19E−01	6.95E−01	9.21E−02	9.85E−02	**7.78E−02**	1.33E−01	5.43E−01	9.61E−02
	STD	1.77E−04	2.33E−03	9.21E−04	4.26E−04	8.84E−03	9.18E−04	1.19E−03	7.61E−04	1.58E−03	2.63E−03	1.87E−03
F8	AT	3.28E−01	3.27E−01	1.04E−01	2.08E−01	6.77E−01	1.02E−01	**8.83E−02**	9.90E−02	1.22E−01	5.81E−01	9.03E−02
	STD	1.75E−04	4.60E−03	1.18E−03	1.25E−03	3.07E−02	8.93E−04	1.07E−03	1.03E−03	1.65E−03	1.52E−03	7.81E−04
F9	AT	2.15E−01	2.36E−01	7.76E−02	1.86E−01	6.14E−01	8.03E−02	7.12E−02	**5.31E−02**	1.01E−01	5.37E−01	6.67E−02
	STD	6.52E−04	2.03E−03	5.27E−04	1.42E−03	4.37E−02	1.34E−03	9.10E−04	1.16E−03	1.01E−03	2.23E−03	3.45E−03
F10	AT	2.79E−01	2.79E−01	9.87E−02	2.12E−01	6.28E−01	1.00E−01	9.04E−02	**6.75E−02**	1.20E−01	5.97E−01	8.33E−02
	STD	4.49E−04	2.40E−03	1.03E−03	1.24E−03	3.73E−02	7.45E−04	1.05E−03	9.91E−04	1.68E−03	3.04E−03	8.43E−04
F11	AT	3.31E−01	3.14E−01	1.21E−01	2.31E−01	6.31E−01	1.24E−01	1.05E−01	**8.17E−02**	1.41E−01	6.09E−01	9.90E−02
	STD	7.74E−04	2.74E−03	1.14E−03	1.20E−03	1.08E−02	9.75E−04	1.29E−03	1.80E−03	2.33E−03	2.95E−03	1.40E−03
F12	AT	8.83E−01	6.72E−01	2.62E−01	3.93E−01	8.60E−01	2.68E−01	2.53E−01	**2.27E−01**	2.86E−01	7.58E−01	2.48E−01
	STD	5.78E−04	4.88E−03	1.75E−03	1.00E−03	2.23E−02	8.52E−04	1.15E−03	4.97E−04	3.03E−03	4.27E−03	3.54E−03
F13	AT	8.74E−01	6.73E−01	2.65E−01	3.93E−01	8.25E−01	2.66E−01	2.53E−01	**2.28E−01**	2.87E−01	7.49E−01	2.50E−01
	STD	2.64E−04	6.07E−03	2.14E−03	1.70E−03	1.89E−02	2.89E−03	1.89E−03	1.37E−03	1.66E−03	2.28E−03	1.66E−03
F14	AT	2.85E + 00	2.20E + 00	8.71E−01	9.99E−01	1.47E + 00	8.72E−01	8.55E−01	**8.17E−01**	8.81E−01	1.30E + 00	8.32E−01
	STD	1.47E−03	1.89E−02	2.93E−03	2.57E−03	7.00E−02	4.81E−03	3.79E−03	3.46E−03	3.85E−03	3.42E−03	8.26E−03
F15	AT	2.68E−01	2.35E−01	8.21E−02	1.92E−01	6.68E−01	8.55E−02	6.72E−02	**5.27E−02**	9.74E−02	5.16E−01	5.67E−02
	STD	2.75E−04	1.65E−03	1.48E−03	5.25E−04	6.15E−02	7.73E−04	9.72E−04	1.20E−03	4.03E−04	3.57E−03	8.36E−04
F16	AT	2.05E−01	1.96E−01	6.12E−02	1.70E−01	6.04E−01	6.32E−02	4.72E−02	**3.32E−02**	7.58E−02	4.61E−01	3.36E−02
	STD	1.13E−04	2.39E−03	8.79E−04	2.57E−04	2.84E−02	2.47E−03	6.00E−04	9.54E−04	9.84E−04	2.82E−03	3.15E−04
F17	AT	1.67E−01	1.72E−01	5.36E−02	1.61E−01	8.92E−01	5.99E−02	3.88E−02	2.43E−02	6.81E−02	4.79E−01	**2.38E−02**
	STD	5.49E−04	3.14E−03	6.89E−03	3.11E−03	6.11E−02	3.45E−03	2.51E−02	1.19E−03	3.64E−03	5.26E−03	9.01E−04
F18	AT	1.67E−01	1.67E−01	5.05E−02	1.59E−01	6.04E−01	5.67E−02	3.72E−02	**2.28E−02**	6.39E−02	4.75E−01	2.34E−02
	STD	1.77E−04	1.35E−03	1.93E−03	2.34E−03	2.15E−02	7.42E−04	9.14E−04	1.09E−03	1.04E−03	3.65E−03	1.13E−03
F19	AT	3.81E−01	3.25E−01	1.14E−01	2.23E−01	6.93E−01	1.20E−01	9.68E−02	**8.16E−02**	1.28E−01	5.28E−01	8.69E−02
	STD	1.92E−04	1.41E−03	5.84E−04	6.65E−03	6.92E−03	8.89E−04	1.23E−03	6.01E−04	1.61E−03	5.38E−03	7.87E−04
F20	AT	3.77E−01	3.17E−01	1.15E−01	2.24E−01	6.93E−01	1.21E−01	1.03E−01	**8.33E−02**	1.34E−E−01	5.50E−01	9.34E−02
	STD	1.74E−04	2.09E−03	7.78E−04	4.83E−04	4.20E−02	9.59E−04	1.14E−03	8.72E−04	1.42E−03	2.91E−03	2.66E−03
F21	AT	7.12E−01	5.74E−01	2.15E−01	3.23E−01	7.83E−01	2.17E−01	2.00E−01	**1.79E−01**	2.31E−01	6.30E−01	1.85E−01
	STD	4.31E−03	4.06E−03	5.23E−04	9.53E−04	2.62E−02	9.67E−04	1.73E−03	1.35E−03	1.10E−03	3.18E−03	1.19E−03
F22	AT	8.85E−01	7.03E−01	2.72E−01	3.90E−01	8.49E−01	2.77E−01	2.55E−01	**2.33E−01**	2.88E−01	6.98E−01	2.36E−01
	STD	3.32E−04	4.12E−03	1.73E−03	7.84E−04	4.21E−02	1.27E−03	3.02E−03	1.57E−03	1.79E−03	2.72E−03	3.70E−03
F23	AT	1.18E + 00	9.18E−01	3.55E−01	4.73E−01	9.36E−01	3.64E−01	3.45E−01	**3.24E−01**	3.69E−01	7.80E−01	3.27E−01
	STD	2.83E−04	5.66E−03	1.13E−03	1.57E−03	4.68E−02	1.61E−03	1.39E−03	1.49E−03	1.33E−03	5.48E−03	2.29E−03
F24	AT	1.71E + 02	1.26E + 02	5.45E + 01	5.46E + 01	5.96E + 01	5.45E + 01	5.46E + 01	5.32E + 01	5.31E + 01	5.38E + 01	**5.31E + 01**
	STD	1.24E−01	1.01E + 00	4.38E−02	4.17E−02	6.81E−01	1.61E−01	1.42E−01	1.60E−01	8.47E−02	7.16E−02	1.30E−01
F25	AT	1.61E + 02	1.28E + 02	5.58E + 01	5.58E + 01	5.93E + 01	5.50E + 01	5.53E + 01	**5.38E + 01**	5.39E + 01	5.45E + 01	5.38E + 01
	STD	2.34E−01	6.60E−01	8.80E−02	7.25E−02	1.31E + 00	1.47E−01	1.80E−01	1.38E−01	1.23E−01	5.89E−02	1.06E−01
F26	AT	1.60E + 02	1.29E + 02	5.54E + 01	5.58E + 01	5.60E + 01	5.49E + 01	5.40E + 01	5.39E + 01	**5.37E + 01**	5.43E + 01	5.38E + 01
	STD	7.05E−01	6.00E−01	5.18E−02	9.63E−02	1.11E + 00	1.53E−01	1.27E−01	1.36E−01	1.71E−01	4.81E−02	1.50E−01
F27	AT	1.66E + 02	1.32E + 02	5.57E + 01	5.56E + 01	5.85E + 01	5.51E + 01	5.40E + 01	5.40E + 01	**5.37E + 01**	5.43E + 01	5.38E + 01
	STD	2.44E−01	8.29E−01	5.03E−02	4.01E−02	1.09E + 00	1.92E−01	7.09E−02	1.79E−01	9.11E−02	1.50E−01	1.17E−01
F28	AT	1.96E + 02	1.49E + 02	6.29E + 01	6.32E + 01	6.75E + 01	6.27E + 01	6.09E + 01	6.09E + 01	6.07E + 01	6.16E + 01	**6.06E + 01**
	STD	1.47E−01	1.07E + 00	1.59E−01	2.32E + 00	1.65E + 00	1.04E−01	2.34E−01	1.64E−01	2.01E−01	6.24E−01	1.99E−01
F29	AT	1.96E + 02	1.48E + 02	6.27E + 01	6.28E + 01	6.71E + 01	6.27E + 01	6.10E + 01	6.09E + 01	6.08E + 01	6.15E + 01	**6.07E + 01**
	STD	1.20E−01	8.60E−01	9.47E−02	3.69E−01	5.01E−01	1.36E−01	7.68E−02	1.35E−01	1.21E−01	1.46E−01	2.05E−01

Table 18
FT for AT of benchmark functions (F1–F29).

Standard	ROA	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
FT	6.1034	8.8276	4.4741	5.8103	10.6207	5.2155	4.3879	3.0000	5.5172	8.2931	3.7500
Rank	8	10	4	7	11	5	3	1	6	9	2

Table 19
WT for F1–F13 with 30 dimensions (p⩾0.05 is displayed in bold).

ROA vs. /	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	1.25E−103	2.34E−162	9.13E−164	9.85E−161	2.00E−93	1.24E−161	1.02E−124	9.87E−163	1.55E−129	2.10E−140
F2	4.95E−98	1.33E−163	9.24E−162	2.05E−159	1.38E−186	1.04E−163	3.41E−111	1.13E−163	4.91E−02	1.46E−136
F3	3.41E−106	7.03E−164	1.25E−164	4.60E−165	3.38E−183	5.57E−164	4.29E−164	1.09E−163	1.88E−126	2.00E−156
F4	3.49E−98	1.43E−172	9.22E−164	7.28E−164	6.97E−173	3.47E−164	2.38E−164	1.05E−163	8.45E−164	2.21E−151
F5	1.74E−141	1.74E−177	1.15E−163	1.24E−84	1.36E−178	3.80E−163	8.73E−15	1.13E−163	8.04E−163	7.97E−39
F6	8.11E−150	1.24E−173	1.29E−163	1.71E−10	9.70E−180	9.64E−28	5.84E−159	5.56E−163	5.13E−163	3.65E−81
F7	8.34E−51	2.01E−166	6.37E−168	8.94E−171	6.37E−191	6.96E−170	6.75E−97	3.18E−168	1.20E−12	8.79E−134
F8	4.21E−162	2.24E−187	1.10E−158	7.34E−132	7.37E−167	3.97E−153	6.76E−162	2.40E−159	3.98E−164	9.32E−152
F9	1.38E−52	1.59E−193	3.49E−184	4.15E−137	2.52E−04	9.29E−186	7.37E−71	2.52E−184	1.35E−19	6.62E−173
F10	2.45E−57	1.99E−188	1.63E−182	1.40E−188	1.14E−05	1.15E−181	1.38E−162	1.53E−182	5.31E−127	5.64E−161
F11	3.96E−45	1.01E−184	2.50E−183	1.51E−105	1.91E−12	5.79E−182	3.09E−55	1.88E−182	6.71E−183	1.16E−41
F12	2.11E−141	4.20E−181	1.13E−163	1.48E−149	1.45E−170	1.26E−162	2.72E−161	1.15E−162	3.69E−162	1.30E−160
F13	9.76E−154	2.52E−165	3.49E−164	2.91E−15	3.12E−167	9.03E−164	1.55E−14	1.46E−163	3.22E−162	1.15E−105

Table 20
WT for F1–F13 with 100 dimensions (p⩾0.05 is displayed in bold).

ROA vs. /	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	5.70E−91	1.39E−166	1.46E−163	2.06E−144	2.28E−70	7.04E−161	5.34E−118	1.39E−162	1.00E−127	1.37E−132
F2	3.47E−E−98	1.08E−164	3.55E−162	1.17E−162	1.37E−186	1.17E−163	1.65E−113	1.16E−163	**1.08E−01**	2.35E−130
F3	2.02E−102	1.66E−164	1.16E−163	2.18E−156	1.82E−181	2.58E−163	1.16E−163	1.16E−163	3.24E−126	1.67E−156
F4	9.31E−96	9.43E−169	1.07E−163	1.62E−161	4.22E−13	1.28E−163	2.80E−164	1.04E−163	4.03E−163	6.62E−153
F5	3.02E−146	3.04E−173	1.08E−163	7.21E−24	1.23E−176	1.91E−163	1.33E−105	3.81E−163	4.71E−163	8.40E−08
F6	2.67E−145	1.98E−163	1.37E−163	7.91E−39	7.41E−177	3.07E−23	7.93E−155	6.30E−163	4.34E−163	3.73E−47
F7	5.60E−82	6.22E−136	2.02E−173	5.30E−180	5.58E−65	5.22E−174	4.80E−139	1.77E−173	5.40E−61	4.30E−132
F8	4.54E−161	2.02E−180	4.53E−143	1.75E−157	2.49E−143	5.75E−148	4.87E−160	2.85E−153	1.24E−156	2.02E−146
F9	3.88E−32	8.17E−183	5.43E−183	1.94E−177	4.33E−04	1.35E−181	3.86E−48	5.02E−182	1.13E−19	4.74E−173
F10	1.55E−54	6.42E−192	5.80E−181	6.78E−136	**2.83E−01**	9.93E−182	6.05E−159	5.59E−181	2.19E−125	3.02E−154
F11	8.51E−49	3.57E−185	7.23E−184	4.06E−48	6.45E−78	1.41E−181	8.03E−58	1.66E−183	1.06E−183	1.66E−35
F12	2.01E−153	1.00E−167	1.18E−163	**9.06E−02**	1.59E−160	1.58E−163	2.94E−158	3.49E−163	3.36E−162	2.17E−140
F13	1.25E−155	4.26E−165	1.10E−163	3.40E−138	1.37E−168	3.60E−159	2.09E−160	1.03E−163	1.72E−162	4.32E−159

Table 21
WT for F1–F13 with 500 dimensions (p⩾0.05 is displayed in bold).

ROA vs. /	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	8.94E−101	7.04E−173	5.55E−164	6.78E−159	5.01E−27	1.64E−161	4.79E−129	6.20E−164	7.04E−130	4.10E−138
F2	1.43E−97	7.30E−163	9.55E−154	2.93E−159	1.37E−186	1.25E−161	2.39E−113	1.03E−162	**1.71E−01**	1.18E−128
F3	2.74E−111	2.15E−166	1.16E−163	1.43E−147	6.78E−184	8.16E−164	1.16E−163	1.15E−163	4.24E−125	2.92E−155
F4	1.33E−93	1.07E−163	8.33E−165	1.16E−160	2.03E−58	2.62E−163	1.01E−162	9.80E−164	1.08E−162	1.19E−150
F5	1.59E−142	4.13E−168	1.13E−163	9.25E−77	2.46E−179	1.79E−149	4.59E−116	4.90E−163	1.65E−163	1.56E−46
F6	8.44E−148	1.48E−162	1.21E−163	1.11E−13	3.26E−175	8.35E−24	1.17E−159	5.10E−163	2.76E−163	1.06E−98
F7	2.72E−05	9.93E−166	6.10E−171	6.04E−177	1.28E−111	1.03E−170	1.41E−164	5.91E−171	2.21E−57	2.25E−76
F8	2.24E−157	6.34E−175	4.14E−129	3.64E−162	2.96E−116	1.44E−163	2.31E−163	4.85E−126	1.48E−159	2.27E−153
F9	1.50E−32	5.41E−187	3.52E−184	2.50E−93	1.47E−04	1.91E−184	1.77E−61	3.50E−184	4.82E−18	5.36E−183
F10	4.00E−49	2.58E−195	5.89E−181	5.68E−152	7.41E−05	1.90E−181	9.14E−150	5.83E−181	1.11E−125	4.69E−148
F11	5.15E−46	5.80E−182	7.20E−182	3.64E−06	1.56E−55	2.53E−181	4.00E−52	1.06E−182	1.20E−182	6.95E−34
F12	2.27E−154	5.90E−169	1.12E−163	2.19E−105	3.36E−179	3.68E−164	6.00E−162	1.13E−163	1.55E−163	2.37E−147
F13	6.77E−151	1.98E−166	1.14E−163	5.18E−24	5.37E−181	8.34E−162	6.22E−161	1.13E−163	2.13E−162	4.50E−158

Table 22
WT for F1–F13 with 1000 dimensions (p⩾0.05 is displayed in bold).

ROA vs. /	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F1	5.96E−103	3.81E−163	8.30E−164	1.09E−160	5.71E−104	3.67E−162	6.00E−123	6.50E−163	2.07E−130	7.33E−136
F2	1.77E−103	7.39E−163	2.38E−153	8.07E−161	1.38E−186	1.95E−162	3.18E−120	1.09E−163	**4.56E−01**	3.92E−137
F3	8.99E−109	7.99E−171	2.29E−162	1.25E−153	3.78E−177	8.82E−162	2.30E−162	2.23E−162	5.02E−122	2.43E−153
F4	1.39E−96	4.80E−166	1.70E−164	4.25E−162	2.37E−106	4.96E−164	7.20E−164	1.06E−163	3.33E−163	3.29E−151
F5	5.61E−150	9.08E−162	1.17E−163	5.51E−165	1.00E−180	2.31E−163	4.72E−158	3.12E−163	2.34E−163	3.06E−94
F6	1.87E−136	1.42E−172	1.29E−163	2.43E−161	8.97E−177	6.57E−29	1.81E−159	4.57E−163	2.75E−163	1.96E−112
F7	2.12E−03	2.30E−142	1.17E−164	1.93E−150	1.39E−60	1.37E−164	3.49E−115	1.13E−164	9.63E−117	3.59E−54
F8	9.31E−158	1.75E−183	2.98E−108	7.97E−11	2.65E−140	4.05E−145	3.29E−161	3.13E−129	3.33E−160	2.99E−147
F9	1.05E−42	5.37E−186	8.02E−142	7.44E−37	1.27E−03	2.44E−185	7.26E−47	8.71E−185	3.41E−21	8.85E−184
F10	2.05E−51	1.26E−181	1.05E−181	1.50E−175	3.15E−03	1.85E−181	1.02E−64	9.70E−182	1.20E−122	2.53E−159
F11	6.29E−40	3.26E−183	3.64E−182	1.77E−75	7.25E−89	3.21E−182	3.13E−50	1.13E−182	1.00E−182	5.73E−43
F12	7.79E−153	1.39E−168	1.13E−163	4.95E−180	5.02E−174	4.12E−164	9.63E−151	9.85E−164	2.70E−163	3.07E−145
F13	1.19E−148	2.33E−177	1.05E−163	7.42E−110	1.08E−164	3.61E−163	1.39E−160	3.51E−163	2.01E−162	4.12E−155

Table 23
WT for F14–F29 (p≥0.05 is displayed in bold).

ROA vs. /	HHO	SFO	EPO	SOA	SHO	SSA	WOA	MFO	MVO	GWO
F14	6.13E−159	7.51E−180	3.77E−168	1.23E−167	1.04E−175	2.79E−10	4.36E−159	2.71E−174	8.91E−143	1.42E−170
F15	1.24E−04	1.54E−163	8.75E−164	1.09E−17	6.77E−188	1.30E−136	3.35E−99	4.43E−149	2.27E−188	7.82E−164
F16	2.94E−68	2.73E−162	8.31E−164	1.56E−22	2.91E−188	1.49E−05	5.55E−85	1.77E−117	2.02E−188	2.21E−14
F17	4.92E−10	4.83E−180	2.45E−08	2.58E−34	1.40E−182	**6.83E−02**	3.80E−75	1.03E−128	2.03E−182	7.60E−70
F18	5.99E−73	1.49E−163	1.10E−44	1.51E−92	5.12E−187	1.04E−08	**9.27E−01**	3.74E−130	5.12E−187	4.66E−128
F19	2.03E−150	1.01E−161	1.16E−164	1.07E−157	2.85E−188	2.30E−05	1.47E−147	7.25E−146	2.75E−191	**8.00E−01**
F20	4.67E−158	1.02E−169	5.47E−165	8.07E−160	8.83E−187	5.49E−06	5.09E−147	2.58E−138	2.88E−188	7.29E−17
F21	4.23E−95	6.57E−185	1.38E−156	1.26E−13	6.68E−165	4.02E−159	3.43E−156	2.99E−127	3.67E−167	9.83E−54
F22	8.59E−133	8.23E−183	8.24E−161	2.46E−83	2.05E−83	**9.81E−01**	2.29E−83	7.52E−105	1.93E−188	3.68E−68
F23	9.42E−124	1.53E−171	2.01E−162	3.84E−126	1.23E−176	1.99E−03	4.42E−147	4.07E−162	3.57E−78	4.66E−118
F24	6.26E−26	1.52E−170	2.06E−165	3.79E−182	1.82E−156	4.68E−25	2.39E−15	4.16E−135	5.70E−175	2.23E−159
F25	4.04E−49	7.19E−185	2.48E−186	2.20E−124	1.27E−03	6.05E−14	5.91E−185	1.76E−183	1.34E−21	6.82E−184
F26	4.04E−E−09	1.76E−182	1.02E−187	6.47E−109	4.33E−04	1.56E−180	2.65E−183	9.63E−181	1.85E−22	1.27E−181
F27	4.66E−36	2.51E−185	3.44E−189	1.03E−48	7.45E−04	7.79E−06	6.27E−185	1.81E−183	1.23E−20	1.14E−184
F28	3.00E−98	2.45E−181	9.81E−170	4.04E−101	8.10E−178	3.13E−20	5.09E−163	8.42E−134	3.81E−175	2.72E−160
F29	1.08E−16	1.42E−175	1.19E−164	1.66E−129	9.65E−173	5.71E−43	6.70E−142	3.00E−126	1.54E−169	1.43E−155

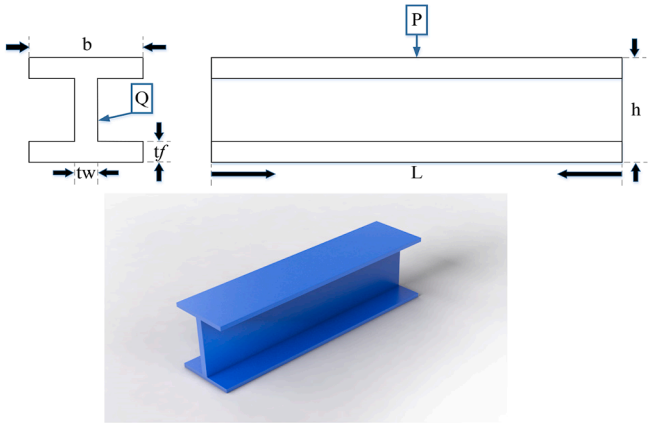


Fig. 10. Schematic of the I-beam (Above: Engineering drawing, Below: 3D).

first 13 test functions in four different dimensions for 30 times and iterate over 500 generations of AF and STD. The result closest to the AF standard value is displayed in bold. According to the AF and STD, the optimal solution can always be found by ROA in F1-4, F9-11 in any dimension. HHO can be used as the second best algorithm. Since most of the selected algorithms are up-to-date or well-known, some algorithms can sometimes find the optimal solution. HHO can generally find the optimal solution of F5, F12 and F13, which shows the effectiveness of HHO on complex functions. However, although ROA is not the most effective, it must be the second best.

In order to visualize the data, Fig. 9 shows the trend of the corresponding fitness function values and box diagram. Since there are many comparison functions, different markers are selected for differentiation. As can be seen from the figure, compared with other functions ROA has a significant advantage in the optimization process of functions F1-F4, ranking in the first good position. Can rank second in function F5. In the function F6, the fourth place. In the functions F7, F8, the convergence trend of HHO is similar. There are certain advantages in the functions F9-F11 and F13. It is second only to HHO in function F12. Comparing each convergence curve, the probability of falling into local optimum is small in the iterative process of ROA. Effective updates are guaranteed both in the early and late iterations, and later fluctuations prove that the diversity between populations is still very rich. A lower position represents excellent property for the box diagram in this test. As it can be seen from the box diagram, the proposed algorithm's optimize performance is most stable and its STD is the smallest.

More comprehensive, the complex function F14-F29 was also tested. The experimental results are shown in Table 16. The closest result to the standard value is shown in bold. The results show that ROA achieves the best results in all test functions, and HHO ranks second. In the MM function (F14-F23), most of the algorithms selected in this paper can also obtain the optimal solution. The ROA in the CM function can achieve the minimum value and has obvious advantages.

- Algorithm running time

The running time of the algorithm is directly related to the complexity of the algorithm and can be used as an intuitive representation of the complexity. The experimental results are shown in Table 17, and the minimum value has been indicated by bolding. Among them the dimension of function F1-F13 is 30. According to the trend of the mark distribution, it can be determined that the WOA has the shortest running time, which also corresponds to its effective and convenient bubble network mode. It can be determined that the STD of each algorithm runs at the same time, indicating that the algorithm is stable enough. In order to draw conclusions more intuitively, the data was further subjected to FT, and the ranking of the running time of the

Table 24
Comparison results of I-beam design problem.

References	Algorithm	Optimum variables				Optimum weight
		b	h	t_w	t_f	
–	ROA	50	80	1.7600	5.0000	**0.0059**
(Shadravan et al., 2019)	SFO	50	80	1.7637	5.0000	0.00662584
(Seyedali, 2015a)	MFO	50	80	1.7647	5.0000	0.0066259
(Seyedali et al., 2017)	SSA	50	80	1.76470587	5.0000	0.006625981
(Wang, 2003)	ARSM	37.05	80	1.71	2.31	0.0157
(Wang, 2003)	IARSM	48.42	79.99	0.90	2.40	0.1310
(Gandomi, Yang, & Alavi, 2013)	CS	50	80	0.9	2.321675	0.0130747
(Cheng & Prayogo, 2014)	SOS	50	80	0.9	2.32179	0.0130741

*The best value is arranged from top to bottom, from small to large.

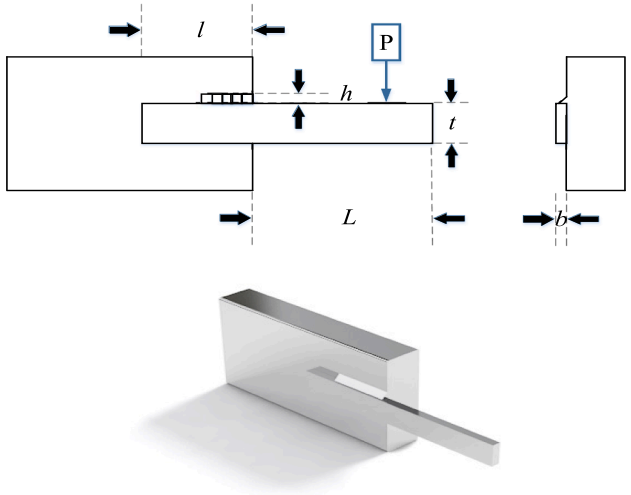


Fig. 11. Schematic of the welded beam (Above: Engineering drawing, Below: 3D).

algorithm was obtained, as shown in Table 18. From the results, WOA ranked first, ROA ranked eighth, and HHO ranked ninth. Within the allowable range, it can be argued that although ROA is not the fastest algorithm, ROA can get the most comprehensive excellence at the expense of a certain running time.

3.5. Statistical analysis

Further WT is performed for the above fitness function values to prove that the ROA results are significantly different from other algorithms. The WT results of F1 - F13 are recorded by Tables 19–22, and significant difference rates for the different dimensions are 100%, 97.67%, 99.23%, and 99.23%. The WT results of F14 - F29 are recorded by Table 23, and significant difference rate for the different dimensions is 96.92%. It can be considered that ROA has significant gaps compared with other algorithms in any dimension and different modes.

In summary, ROA can be considered to have strong competitiveness in many novel and excellent algorithms.

4. ROA for classical engineering problems

The effectiveness of a good algorithm lies in its application. ROA has shown certain advantages in different types of functions in the previous section. Next, five engineering problems were selected for further testing. They are: I-beam design problem, Welded beam design problem, Pressure vessel design problem, Three-bar truss design problem, Rolling element bearing problem. In this section, In this section, a detailed introduction to all engineering issues, including text descriptions and formula modeling are provided. For the sake of simplicity, all problems are implemented in MATLAB through the barrier penalty function. The results obtained by ROA and the corresponding results of different algorithms in the relevant literature are listed in the table at the end of each engineering problem, which is convenient for comparison observation. The ROA runs independently for each project 30 times, with a selected remora population of 30 and an iteration of 500. Finally, make a corresponding evaluation for different issues.

4.1. I-beam design problem

This problem is one of the well-known structural optimization problems. I-beam is usually made of structural steel and is a section steel with a cross-sectional shape of I-shaped, which is commonly used in construction and civil engineering. Vertical deflection value is a very important parameter in this problem. The smaller the value, the more precise the structure of the I-beam. This problem has several structural parameters such as section width (b), section height (h), web thickness (t_w) and flange thickness (t_f). The composition of different structures and the 3D renderings are shown in Fig. 10. Mathematically model the structure of the I-beam, establishing the objective function, and finding the minimum value of the objective function. The conditions of the corresponding penalty function and the range of related parameters are as follows:

Consider	$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [b \quad h \quad t_w \quad t_f]$
Minimize	$f(\vec{x}) = \dfrac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2}$
Subject to	$g(\vec{x}) = 2bt_w + t_w(h-2t_f) \leqslant 0$
Variable range	$10 \leqslant x_1 \leqslant 50$
	$10 \leqslant x_2 \leqslant 80$

(continued on next page)

Table 25
Comparison results of the welded beam design problem.

References	Algorithm	Optimum variables				Optimum weight
		h	l	t	b	
–	ROA	0.200077	3.365754	9.011182	0.206893	**1.706447**
(Seyedali, 2015a)	MFO	0.2057	3.4703	9.0364	0.2057	1.72452
(Kaveh & Mahdavi, 2014)	CBO	0.205722	3.47041	9.037276	0.205735	1.724663
(Mahdavi, Fesanghary, & Damangir, 2007)	IHS	0.20573	3.47049	9.03662	0.2057	1.7248
(Seyedali et al., 2014)	GWO	0.205676	3.478377	9.03681	0.205778	1.72624
(Seyedali et al., 2015)	MVO	0.205463	3.473193	9.044502	0.205695	1.72645
(Coello Coello and Mezura, 2002)	Coello and Montes	0.205986	3.471328	9.020224	0.206480	1.72822
(Seyedali & Andrew, 2016)	WOA	0.205396	3.484293	9.037426	0.206276	1.730499
(He & Wang, 2007)	CPSO	0.202369	3.544214	9.048210	0.205723	1.73148.
(Kaveh & Khayatazad, 2012)	RO	0.203687	3.528467	9.004233	0.207241	1.735344
(Coello Coello, 2002)	Coello	0.208800	3.420500	8.997500	0.2100	1.74831
(Deb, 2000)	GA	0.1829	4.0483	9.3666	0.2059	1.82420
(Esmat et al., 2009).	GSA	0.182129	3.856979	10.00000	0.202376	1.879952
(Lee & Geem, 2005)	HS	0.2442	6.2231	8.2915	0.2443	2.3807
(Ragsdell & Phillips, 1976)	Approx	0.2444	6.2189	8.2915	0.2444	2.3815
(Siddall, 1972)	Siddall	0.2444	6.2189	8.2915	0.2444	2.38154
(Ragsdell & Phillips, 1976)	David	0.2434	6.2552	8.2915	0.2444	2.3841
(Ragsdell & Phillips, 1976)	Ragsdell	0.2455	6.1960	8.2730	0.2455	2.38594
(Carlos & Coello, 2000)	GA	0.2489	6.1730	8.1789	0.2533	2.43312
(Ragsdell & Phillips, 1976)	Simplex	0.2792	5.6256	7.7512	0.2796	2.53073
(Ragsdell & Phillips, 1976)	Random	0.4575	4.7313	5.0853	0.6600	4.11856

*The best value is arranged from top to bottom, from small to large.

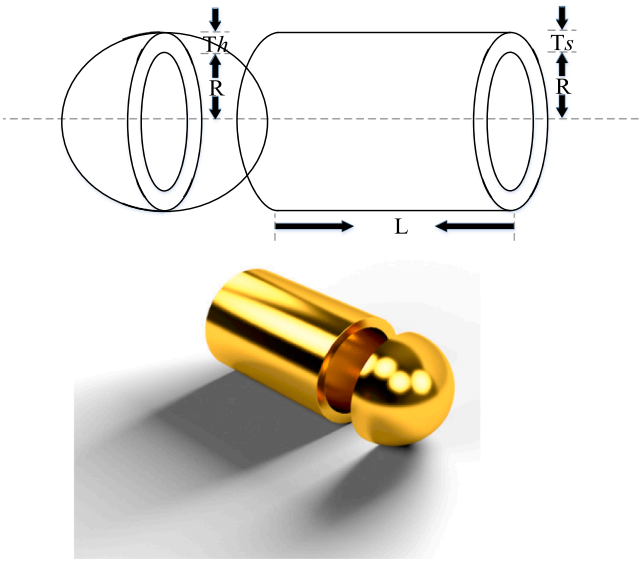


Fig. 12. Schematic of the pressure vessel (Above: Engineering drawing, Below: 3D).

(*continued*)

Consider	$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [b \quad h \quad t_w \quad t_f]$

(*continued on next column*)

(*continued*)

Consider	$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [b \quad h \quad t_w \quad t_f]$
	$0.9 \leqslant x_3 \leqslant 5$
	$0.9 \leqslant x_4 \leqslant 5$

The experimental results of the I-beam design problem are recorded in Table 24. At the same time, the data of different optimization algorithms for the design problem of I-beams are also entered in the table, and relevant literatures are also given.

The results show that ROA has great advantages compared to other algorithms and can achieve the ability to minimize vertical deflection.

4.2. Welded beam design problem

As it named, this problem deals with designing a welded beam to minimize the fabrication cost. The minimization process is subject to some constraints such as shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam, and side constraints. This optimum design has four parameters: thickness of weld (h), length of the clamped bar (l), height of the bar (t), and thickness of the bar (b) as shown in Fig. 11. The mathematical formulation is also illustrated as follows:

Consider	$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [h \quad l \quad t \quad b]$
Minimize	$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$
Subject to	$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leqslant 0$

(*continued on next page*)

Table 26
Comparison results of the pressure vessel design.

References	Algorithm	Optimum variables				Optimum weight
		T_s	T_h	R	L	
	ROA	0.729588	0.222651	40.432340	198.553762	**5311.917579**
(Dhiman and Kumar, 2017)	SHO	0.778210	0.384889	40.315040	200.0000	5885.5773
(Seyedali et al., 2014)	GWO	0.812500	0.434500	42.089181	176.758731	6051.5639
(Kaveh & Talatahari, 2010)	ACO	0.812500	0.437500	42.103624	176.572656	6059.0888
(Seyedali, 2015a)	MFO	0.812500	0.437500	42.098445	176.636596	6059.7143
(Li et al., 2007)	DE	0.812500	0.437500	42.098411	176.637690	6059.7340
(Seyedali & Andrew, 2016)	WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410
(Efrén & Carlos, 2008)	ES (Coello and Montes)	0.812500	0.437500	42.098087	176.640518	6059.7456
(Carlos & Efrén, 2002)	GA (Coello and Montes)	0.812500	0.437500	42.098087	176.640518	6059.7456
(Seyedali et al., 2015)	MVO	0.812500	0.437500	42.0907382	176.738690	6060.8066
(He and Wang, 2007)	PSO	0.812500	0.437500	42.091266	176.746500	6061.0777
(Carlos & Coello, 2000)	GA (Coello)	0.812500	0.434500	40.323900	200.0000	6288.7445
(Kalyanmoy, 1997)	GA (Deb and Gene)	0.937500	0.500000	48.329000	112.679000	6410.3811
(Mahdavi et al., 2007)	IHS	1.125000	0.625000	58.29015	43.69268	7197.730
(Kannan & Kramer, 1994)	Lagrangian Multiplier (Kannan)	1.125000	0.625000	58.291000	43.69000	7198.0428
(Sandgren, 1990)	Branch-bound (Sandgren)	1.125000	0.625000	47.700000	117.7010	8129.1036
(Esmat et al., 2009).	GSA	1.125000	0.625000	55.9886598	84.4542025	8538.8359

*The best value is arranged from top to bottom, from small to large.

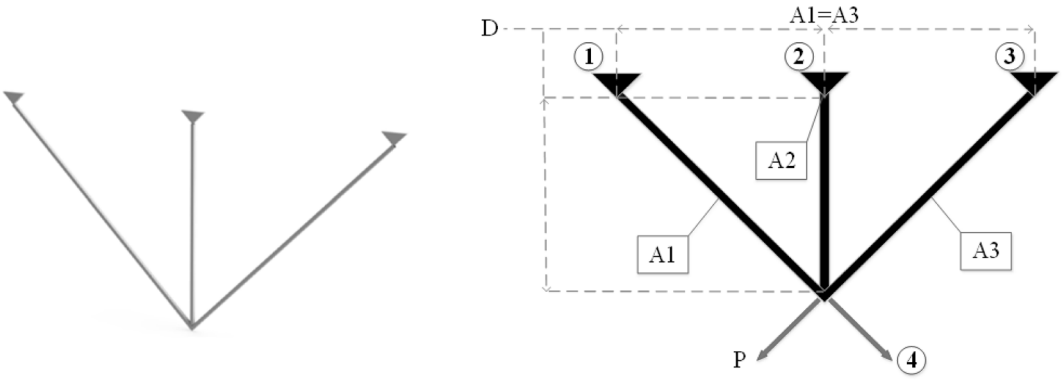


Fig. 13. Schematic of the Three-bar truss design (Left: 3D, Right: Engineering drawing).

Table 27
Comparison results of the Three-bar truss design problem.

References	Algorithm	Optimum variables		Optimum weight
		A_1	A_2	
	ROA	0.80047516	0.265380986	**186.6524003**
(Jung-Fa, 2005)	Tsa	0.788	0.408	263.68
(Hui, Zixing, & Yong, 2010)	PSO-DE	0.7886751	0.4082482	263.8958433
(Ali et al., 2019)	HHO	0.788662816	0.40828313	263.8958434
(Seyedali, 2015b)	ALO	0.7886618	0.4082831	263.8958434
(Seyedali et al., 2017)	SSA	0.78866541	0.40827578	263.8958434
(Min, Wenjian, & Xufa, 2008)	DEDS	0.78867513	0.40824828	263.8958434
(Seyedali et al., 2015)	MVO	0.78860276	0.40845307	263.8958499
(Ali et al., 2013)	MBA	0.7885650	0.4085597	263.8958522
(Shahrzad, Seyedali, & Andrew, 2017)	GOA	0.78889755	0.40761957	263.8958814
(Shadravan et al., 2019)	SFO	0.7884562	0.40886831	263.8959212
(Seyedali, 2015a)	MFO	0.7882447	0.4094669	263.8959796
	CS	0.78867	0.40902	263.9716
(Tapabrata and Pankaj, 2001)	Ray and Sain	0.795	0.395	264.3

*The best value is arranged from top to bottom, from small to large.

(*continued*)

Consider	$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [h \quad l \quad t \quad b]$
	$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leqslant 0$
	$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leqslant 0$
	$g_4(\vec{x}) = x_1 - x_4 \leqslant 0$
	$g_5(\vec{x}) = P - P_c(\vec{x}) \leqslant 0$
	$g_6(\vec{x}) = 0.125 - x_1 \leqslant 0$
	$g_7(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leqslant 0$
Variable range	$0.1 \leqslant x_1 \leqslant 2$
	$0.1 \leqslant x_2 \leqslant 10$
	$0.1 \leqslant x_3 \leqslant 10$
	$0.1 \leqslant x_4 \leqslant 2$
Where	$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$
	$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$
	$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$
	$J = 2\{\sqrt{2}x_1x_2[\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2]\}$
	$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4} \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4}$
	$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}})$
	$P = 6000lb, L = 14in, E = 30 \times 1^6 psi,$
	$\tau_{max} = 13600psi, \sigma_{max} = 30000psi$

This problem has also tested and the results are shown in the Table 25. It is shown that the proposed algorithm can find the lowest cost design. Thus, it is reasonable to think that the proposed algorithm is feasible in

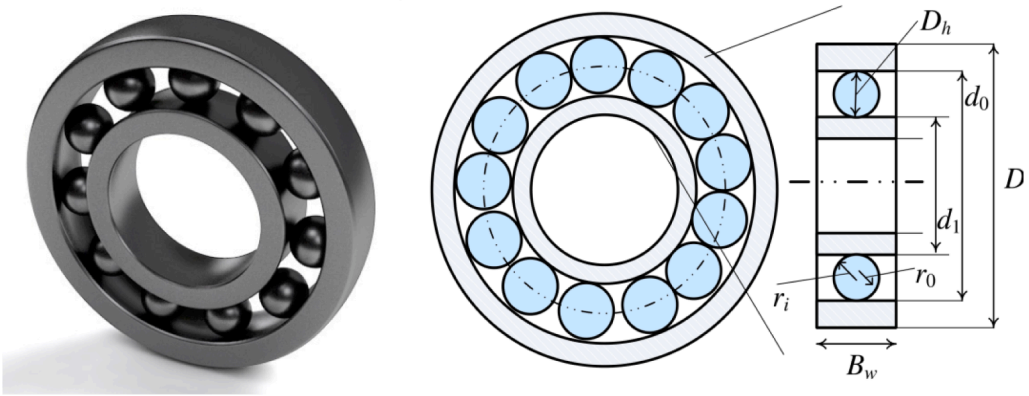


Fig. 14. Schematic of the rolling element bearing (Left: 3D, Right: Engineering drawing).

Table 28
Comparison results of the rolling element bearing problem.

References		(Shantanu, Rajiv, & Shivashankar, 2007)	(Ravipudi et al., 2011)	(Poonam & Vimal, 2016)	(Ali et al., 2019)	(Dhiman & Kumar, 2017)	
Algorithm		GA4	TLBO	PVS	HHO	SHO	ROA
Optimum variables	D_m	125.71910	125.71910	125.71906	125.00000	125.00000	127.41090
	D_b	21.423000	21.425590	21.425590	21.000000	21.407320	22.065953
	Z	11.000000	11.000000	11.000000	11.092073	10.932680	11.902282
	f_i	0.5150000	0.5150000	0.5150000	0.5150000	0.5150000	0.5249904
	f_0	0.5150000	0.5150000	0.5150000	0.5150000	0.5150000	0.5252150
	K_{dmin}	0.4159000	0.4242660	0.4004300	0.4000000	0.4000000	0.4077870
	K_{dmax}	0.6510000	0.6339480	0.6801600	0.6000000	0.7000000	0.6116139
	ε	0.3000043	0.3000000	0.3000000	0.3000000	0.3000000	0.3058120
	e	0.0223000	0.0688580	0.0799900	0.0504740	0.0200000	0.0259788
	ξ	0.7510000	0.7994980	0.7000000	0.6000000	0.6000000	0.6116873
Optimum weight		81843.30	81859.74	81859.741	83011.883	85054.532	**87971.853**

* The best values are arranged from left to right, from small to large.

solving such problems.

4.3. Pressure vessel design

The objective of this problem is to minimize the total cost of the vessel with four variables: Thickness of the shell (Ts). Thickness of the head (Th). Inner radius (R). Length of the cylindrical section without considering the head(L). And the optimum design is subjected to constraints on material, forming, and molding of a cylindrical vessel. The approximate shape and parameter distribution are shown in Fig. 12. The mathematical formulation of this optimization is as follows:

Table 26 Illustrates the results of ROA and the comparison algorithm. As it represents, the proposed algorithm can reach the lowest cost design. Therefore, it is hopefully that the proposed algorithm is suitable for such problems.

Consider	$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [T_s \quad T_h \quad R \quad L]$
Minimize	$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$
Subject to	$g_1(x) = -x_1 + 0.0193x_3 \leqslant 0$
	$g_2(\vec{x}) = -x_3 + 0.00954x_3 \leqslant 0$
	$g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leqslant 0$
	$g_4(\vec{x}) = x_4 - 240 \leqslant 0$
Variable range	$0 \leqslant x_1 \leqslant 99$
	$0 \leqslant x_2 \leqslant 99$
	$10 \leqslant x_3 \leqslant 200$
	$10 \leqslant x_4 \leqslant 200$

4.4. Three-bar truss design problem

As one of the most researched work cases, it is often used by various algorithms for testing. This task is also a minimization problem, mainly to find the minimum of the total weight of the structure. Fig. 13 illustrates the shape of the truss and the associated forces on the structure. The figure contains two parameters: Area of strip 1 = Area of strip 3, and Area of strip 2. This problem can be mathematically described as follows:

Table 27 shows the detailed results of other algorithms and proposed ROA. Based on the results in Table 27, ROA is significantly better than other optimizers, and ROA is observed to be exceptionally competitive.

4.5. Rolling element bearing problem

Different from the previous problems, this engineering problem is the maximum value of the dynamic bearing capacity of the solution target. A total of 10 variables are included, a schematic of which is shown in Fig. 14. The test case is expressed as follows:

Table 28 lists the results of the ROA and other optimizers. As can be seen from Table 28, the proposed ROA has detected the best solution with the greatest cost and made substantial progress compared to other algorithms.

5. Conclusion

A new type of bionic optimization algorithm ROA was proposed. The main source of inspiration comes from remora. Simulations were carried out for their living habits of parasitic feeding on different hosts. Two types of hosts–whales and swordfish, which exhibit the best effect and the most characteristic movement mode are selected to update the switch of different modes. Further, the remora factor is proposed to facilitate the adjustment of local renewal. 29 unconstrained benchmark problems such as unimodal functions and multimodal function combination functions were used to evaluate the performance of the ROA. To highlight the effectiveness of the proposed algorithm through comparison, one or two of the highly respected optimizers proposed in the last five years are selected annually. The exploitative, exploratory and local optimal avoidance capabilities of ROA are studied in depth. In addition, four restricted engineering tasks for minimization and one engineering task for maximum value were selected for the comprehensive development of ROA. The overall experimental results show that ROA has significant competitiveness in many state-of-the-art algorithms.

In the future work. the ROA will be refined to simulate a valid ROA framework and provide opportunities for more hosts. Furthermore, ROA will also be applied for the optimization of various engineering problems such as image segmentation, feature selection, and multi-objective problems.

CRediT authorship contribution statement

Heming Jia: Conceptualization, Methodology, Supervision. **Xiaoxu Peng:** Data curation, Writing - original draft, Software. **Chunbo Lang:** Visualization, Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Sanming University introduces high-level talents to start scientific research funding support project (20YG14), Guiding science and technology projects in Sanming City (2021-S-8), Educational research projects of young and middle-aged teachers in Fujian Province (JAT200618), Scientific research and development fund of Sanming University (B202009), and Funded By Open Research Fund Program of Fujian Provincial Key Laboratory of Agriculture Internet of Things Application (ZD2101).

References

Amir, H. G., Xin-She, Y., & Siamak, T. (2013). Metaheuristic algorithms in modeling and optimization. *Metaheuristic Applications in Structures and Infrastructures, 1–24*. https://doi.org/10.1016/B978-0-12-398364-0.00001-2

Ali, A. H., Seyedali, M., & Hossam, F. (2019). Harris hawks optimization: Algorithm and applications, Future Gener. *The Journal of Computer and System Sciences, 97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

Ali, S., Ardeshir, B., & Hadi, E. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing, 13*, 2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026

Atmaca, B., Dede, T., & M. Grzywiński. (2020). Optimization of cables size and prestressing force for a single pylon cable-stayed bridge with Jaya algorithm. Steel Compos. Struct., 34(6), 853-862. 10.12989/scs.2020.34.6.853.

Bruce, O'. T. (2002). Phylogeny of the species of the superfamily Echeneoidea (Perciformes: Carangoidei: Echeneidae, Rachycentridae, and Coryphaenidae), with an interpretation of echeneid hitchhiking behaviour. *Canadian Journal of Zoology, 80*, 596–623. https://doi.org/10.1139/z02-031

Cressey, R. F., & Lachner, E. A. (1970). The parasitic copepod diet and life history of diskfishes (Echeneidae). *Copeia, 1970*(2), 310. https://doi.org/10.2307/1441652

Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures, 139*, 98–112. https://doi.org/10.1016/j.compstruc.2014.03.007

Coello Coello, C. A., & Mezura, M. E. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*, 193–203. https://doi.org/10.1016/S1474-0346(02)00011-3

Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Comput. Method Appl. Mech. Eng., 191*(11-12), 1245–1287. https://doi.org/10.1016/S0045-7825(01)00323-1

Carlos, A., & Coello, C. (2000). Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering System, 17*, 319–346. https://doi.org/10.1080/02630250008970288

Carlos, A. C. C., & Efrén, M. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*, 193–203. https://doi.org/10.1016/S1474-0346(02)00011-3

David, H. W., & William, G. M. (1997). No free lunch theorems for optimization, IEEE Trans. *Evolutionary Computation, 1*, 67–82. https://doi.org/10.1109/4235.585893

Dawid, P., & Marcin, W. (2017). Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement and Dynamic Birth and Death Mechanism. *Symmetry, 9*(10), 203. https://doi.org/10.3390/sym9100203

Dawid, P., & Marcin, W. (2021). Red fox optimization algorithm. *Expert Systems with Applications, 166*(10), Article 114107. https://doi.org/10.1016/j.eswa.2020.114107

Derrac, J., García, S., Molina, D., & etc.. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*, 3–18. https://doi.org/10.1016/j.swevo.2011.02.002

Deb, K. (2000). An effiffiifficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*, 311–338. https://doi.org/10.1016/S0045-7825(99)00389-8

Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software, 114*, 48–70. https://doi.org/10.1016/j.advengsoft.2017.05.014

Efrén, M., & Carlos, C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problem. *International Journal of General Systems, 37*, 443–473. https://doi.org/10.1080/03081070701303470

El-Ghazal, T. (2009). *Metaheuristics: From Design to Implementation*. France: John Wiley & Sons (Chapter 3).

Esmat, R., Hossein, N., & Saeid, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*, 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

Fischer, W., Fao, R. E. D., & FIR (1978). FAO species identification sheets for fishery purposes, Western Central Atlantic (fishing area 31) / edited by W. Fischer. western central atlantic.

Fred, G. (1989). Tabu search I, ORSA. *Journal of Computation, 1*, 2093–2229. https://doi.org/10.1287/ijoc.1.3.190

Fertl, D., & Landry, A. M. (1999). Sharksucker (Echeneis naucrates) on a bottlenose dolphin (Tursiops truncatus) and a review of other cetacean-remora associations. *Marine Mammal Science, 15*(3), 859–863. https://doi.org/10.1111/mms.1999.15.issue-310.1111/j.1748-7692.1999.tb00849.x

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of American Statistical Association, 32*(200), 675–701. https://doi.org/10.1080/01621459.1937.10503522

Gudger, E. W. (1919). On the use of the sucking-fish for catching fish and turtles: Studies in Echeneis or Remora. *II. Am. Nat., 53*(628), 446–467. https://doi.org/10.1086/279724

Gaurav, D., & Vijay, K. (2018). Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowledge Based Syst., 159*, 20–50. https://doi.org/10.1016/j.knosys.2018.06.001

Gaurav, D., & Vijay, K. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge Based Syst., 165*, 169–196. https://doi.org/10.1016/j.knosys.2018.11.024

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*(1), 17–35. https://doi.org/10.1007/s00366-011-0241-y

Haeberle, H. S., Helm, J. M., Navarro, S. M., Karnuta, J. M., Schaffer, J. L., Callaghan, J. J., et al. (2019). Artificial Intelligence and Machine Learning in Lower Extremity Arthroplasty: A Review. *Journal of Arthroplasty, 34*(10), 2201–2203. https://doi.org/10.1016/j.arth.2019.05.055

He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*, 89–99. https://doi.org/10.1016/j.engappai.2006.03.003

Heidari, A. A., Aljarah, I., Faris, H., Chen, H., Luo, J., & Mirjalili, S. (2019). An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Computing & Applications, 32*(9), 5185–5211. https://doi.org/10.1007/s00521-019-04015-0

Hossam, F., Al-Zoubi, A. M., Heidari, A. A., Aljarah, I., Mafarja, M., Hassonah, M. A., et al. (2019). An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Information Fusion, 48*, 67–83. https://doi.org/10.1016/j.inffus.2018.08.002

Ztürk, H. T., Dede, T., & Türker, E. (2020). Optimum design of reinforced concrete counterfort retaining walls using tlbo. *Jaya algorithm, Structures, 25*, 285–296. https://doi.org/10.1016/j.istruc.2020.03.020

Hui, L., Zixing, C., & Yong, W. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*, 629–640. https://doi.org/10.1016/j.asoc.2009.08.031

Ibrahim, R. A., Elaziz, M. A., Lu, S., & etc.. (2018). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert System Application, 108*, 1–27. https://doi.org/10.1016/j.eswa.2018.04.028

James, K., & Russell, E. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, 4*, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

John, H. H. (1992). Genetic algorithms. *Scientific American, 267*, 66–73. https://doi.org/10.1038/scientificamerican0792-66

John, R. K. (1994). Genetic Programming II: Automatic Discovery of Reusable Programs. *Artificial Life, 1*, 439–441. https://doi.org/10.1162/artl.1994.1.4.439

Jia, H., Lang, C., Oliva, C., & etc.. (2019). Hybrid Grasshopper Optimization Algorithm and Differential Evolution for Multilevel Satellite Image Segmentation. *Remote Sens., 11*, 11–34. https://doi.org/10.3390/rs11091134

Jung-Fa, T. (2005). Global optimization of nonlinear fractional programming problems in engineering design. *Engineering Optimization, 37*, 399–409. https://doi.org/10.1080/03052150500066737

Kalemci, E. N., İkizler, S. B., Dede, T., & Angın, Z. (2020). Design of reinforced concrete cantilever retaining wall using Grey wolf optimization algorithm. *Structures, 23*, 245–253. https://doi.org/10.1016/j.istruc.2019.09.013

Kaveh, A., & Mahdavi, V. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & Structures, 139*, 18–27. https://doi.org/10.1016/j.compstruc.2014.04.005

Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: Ray optimization. *Computers & Structures, 112*, 283–294. https://doi.org/10.1016/j.compstruc.2012.09.003

Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computer: International Journal of Computer Aided Engineering, 27*, 155–182. https://doi.org/10.1108/02644401011008577

Kalyanmoy D., (1997). GeneAS: A Robust Optimal Design Technique for Mechanical Component Design, Presented at the D. Dasgupta, Z. Michalewicz (eds) Evolutionary algorithms in engineering applications, Berlin, 10.1007/978-3-662-03423-1_27.

Kang, S. L., & Zong, W. G. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & Structures, 82*(9–10), 781–798. https://doi.org/10.1016/j.compstruc.2004.01.002

Kannan, B. K., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116*(2), 405–411. https://doi.org/10.1115/1.2919393

Liming, S., Huiling, C., Zhe, Y., & etc.. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems, 96*, 61–75. https://doi.org/10.1016/j.knosys.2016.01.002

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Method Appl. Mech. Eng., 194*(36-38), 3902–3933. https://doi.org/10.1016/j.cma.2004.09.007

Li, L., Huang, Z., Liu, F., & etc.. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures, 85*, 340–349. https://doi.org/10.1016/j.compstruc.2006.11.020

M. Grzywiński, Dede, T. & Y. I. zdemír. (2019). Optimization of the braced dome structures by using jaya algorithm with frequency constraints. Steel Compos. Struct., 30(1), 47-55. 10.12989/scs.2019.30.1.047.

Maximilian, T., Can, B., Tamara, N., & etc.. (2019). Intelligent optimization and machine learning algorithms for structural anomaly detection using seismic signals. *Mechanical Systems and Signal Processing, 133*. https://doi.org/10.1016/j.ymssp.2019.106250

Marco, D., Vittorio, M., & Alberto, C. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern., B,26*, 29–41. https://doi.org/10.1109/3477.484436

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*, 1567–1579. https://doi.org/10.1016/j.amc.2006.11.033

Mingjing, W., Huiling, C., Bo, Y., & etc.. (2017). Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing, 267*, 69–84. https://doi.org/10.1016/j.neucom.2017.04.060

Min, Z., Wenjian, L., & Xufa, W. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences, 178*, 3043–3074. https://doi.org/10.1016/j.ins.2008.02.014

Nelson, J. S., Crossman, E. J., Espinosapérez, H., et al. (2006). Corrections to "Common and Scientific Names of Fishes from the United States, Canada, and Mexico". *Copeia*, 559–562.

Oliva, D., Hinojosa, S., Cuevas, E., & etc.. (2017). Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm, Expert Syst. *Appl., 79*, 164–180. https://doi.org/10.1016/j.eswa.2017.02.042

Poonam, S., & Vimal, S. (2016). Passing vehicle search (pvs): A novel metaheuristic algorithm. *Applied Mathematical Modelling, 40*, 3951–3978. https://doi.org/10.1016/j.apm.2015.10.040

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems. *Information Science, 183*, 1–15. https://doi.org/10.1016/j.ins.2011.08.006

Ralf, B., & Johnson, G. D. (2012). Ontogeny and homology of the skeletal elements that form the sucking disc of remoras (Teleostei, Echeneoidei, Echeneidae). *Journal of Morph, 273*(12), 1353–1366. https://doi.org/10.1002/jmor.v273.1210.1002/jmor.20063

Ragsdell, K., & Phillips, D. (1976). Optimal design of a class of welded structures using geometric programming, ASME. *Journal of Engineering for Industry, 98*, 1021–1025. https://doi.org/10.1115/1.3438995

Ravipudi, V. R., & Vimal, S. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*, 303–315. https://doi.org/10.1016/j.cad.2010.12.015

Suganthan, P. N., Hansen, N., Liang, J. J., & etc.. (2005). Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *Natural Computing*, 341–357. https://www.researchgate.net/publication/235710019_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2005_Special_Session_on_Real-Parameter_Optimization.

Seyedali, M., & Andrew, L. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Scott, K., Gelatt, C. D., Jr, & Mario, P. V. (1983). Optimization by simulated annealing. *Science, 220*, 671–680. https://doi.org/10.1126/science.220.4598.671

Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence, 80*, 20–34. https://doi.org/10.1016/j.engappai.2019.01.001

Seyedali, M., Amir, H. G., Seyedeh, Z. M., & etc.. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

Seyedali, M. (2015a). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge Based Syst., 89*, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006

Seyedali, M., Seyed, M. M., Abdolreza, H., & etc.. (2015). Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Computing & Applications, 27*, 495–513. https://doi.org/10.1007/s00521-015-1870-7

Seyedali, M., Seyed, M. M., & Andrew, L. (2014). Grey Wolf Optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Siddall, J. N. (1972). *Analytical decision-making in engineering design*. Englewood Cliffs NJ: Prentice-Hall. https://www.researchgate.net/publication/280940057_Analytical_decision-making_in_engineering_design.

Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design., 112*, 223–229. https://doi.org/10.1115/1.2912596

Seyedali, M. (2015b). The Ant Lion Optimizer. *Advances in Engineering Software, 83*, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

Shahrzad, S., Seyedali, M., & Andrew, L. (2017). Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Shantanu, G., Rajiv, T., & Shivashankar, B. N. (2007). Multi-objective design optimisation of rolling bearings using genetic algorithms. *Mechanism and Machine Theory, 42*, 1418–1443. https://doi.org/10.1016/j.mechmachtheory.2006.10.002

Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. *Journal of Heuristics, 8*(5), 541–564. https://doi.org/10.1023/A:1016540724870

Tapabrata, R., & Pankaj, S. (2001). Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization, 33*, 735–748. https://doi.org/10.1080/03052150108940941

Williams E. H., Mignucci-Giannoni A. A., & Bunkley-Williams L., etc. (2003). Echeneid-sirenian associations, with information on sharksucker diet, J. Fish Biol., 63, 1176-1183. 10.1046/j.1095-8649.2003.00236.x.

Wang, G. G. (2003). Adaptive response surface method using inherited latin hypercube design points. *Journal of Mechanical Design, 125*, 210–220. https://doi.org/10.1115/1.1561044

Yuankang, F., Ziyang, Z., Zhiqiu, H., & etc.. (2010). Multi-objective fuzzy clustering method for image segmentation based on variable-length intelligent optimization algorithm. *Advanced Computational Intelligence, 6382*, 329–337. https://doi.org/10.1007/978-3-642-16493-4_34

Zhang, Q., Huiling, C., Jie, L., & etc.. (2018). Chaos enhanced bacterial foraging optimization for global optimization. *IEEE Access*. https://doi.org/10.1109/ACCESS.2018.2876996