Applied Mathematical Modelling 73 (2019) 109–123

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Multi-strategy boosted mutative whale-inspired optimization approaches

Jie Luo[a], Huiling Chen[a,*], Ali Asghar Heidari[b,c], Yueting Xu[a], Qian Zhang[a], Chengye Li[d]

[a] *Department of Computer Science, Wenzhou University, Wenzhou 325035, China*
[b] *School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, 1439957131, Iran*
[c] *Department of Computer Science, School of Computing, National University of Singapore, 117417, Singapore*
[d] *Department of Pulmonary and Critical Care Medicine, the First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, China*

A R T I C L E I N F O

Article history:
Received 26 July 2018
Revised 20 March 2019
Accepted 28 March 2019
Available online 5 April 2019

Keywords:
Whale optimization algorithm
Function optimization
Chaotic local search
Gaussian mutation
Chaotic initialization

A B S T R A C T

This paper presents an improved Whale Optimization Algorithm (WOA) for global optimization. WOA is a recently introduced meta-heuristic algorithm mimicking the hunting behavior of humpback whales. Owing to its simplicity in exploratory and exploitative operators and the satisfactory efficacy, this algorithm has found its place among the well-established population-based approach utilized in many engineering and science areas. However, this method is easy to fall into local optimum when dealing with some optimization cases. In order to further enhance its exploratory and exploitative performance, three strategies are incorporated into the original method to keep a better balance between exploitation and exploration tendencies. First, the chaotic initialization phase is introduced into the optimizer to initiate the swarm of chaos-triggered whales. Then, Gaussian mutation is employed to intensify the diversity level of the evolving population. At last, a chaotic local search with a 'shrinking' strategy is used to enhance the exploitative leanings of the basic optimizer. In order to verify the effectiveness of the improved WOA, it is compared to four meta-heuristic and state-of-the-art evolutionary algorithms on representative benchmark functions. Trial results and simulations reveal that not only the proposed improved WOA is significantly better than those basic algorithms including original WOA but also it is superior to compared state-of-the-art approaches. Moreover, the proposed algorithm is successfully applied to realize three constrained engineering test cases, which the results suggest that the improved WOA can effectively deal with the constrained functions as well.

1. Introduction

Global optimization problems are difficult to be solved efficiently because of their high nonlinearity and multiple local optima (LO). Traditional gradient-based methods usually face difficulties in dealing with some classes of problems [1]. In recent decades, many meta-heuristic optimization algorithms have been used as alternative solvers to deal with such

* Corresponding author.
E-mail addresses: jieluocn@foxmail.com (J. Luo), chenhuiling.jlu@gmail.com (H. Chen), as_heidari@ut.ac.ir, aliasgha@comp.nus.edu.sg, t0917038@u.nus.edu (A.A. Heidari), yuetingXu96@163.com (Y. Xu), qz_zhangqian@163.com (Q. Zhang), lichengye41@126.com (C. Li).

https://doi.org/10.1016/j.apm.2019.03.046
0307-904X/

problems, including classical meta-heuristic algorithms such as Genetic Algorithm (GA) [2], Particle Swarm Optimization (PSO) [3,4] and some algorithms proposed in recent years, like Grey Wolf optimizer (GWO) [5], Salp Swarm Algorithm (SSA) [6,7], Bacterial Foraging Optimization algorithm (BFO) [8], Fruit Fly Optimization algorithm (FOA) [9], Ant Colony Optimization algorithm (ACO) [10] and Grasshopper Optimization Algorithm (GOA) [11]. Mirjalili and Lewis recently proposed a novel meta-heuristic algorithm, WOA, designed based on the chasing deeds of a class of marine predators, called humpback whales [12].

Since its appearance, WOA has attracted attention and therefore, it has been utilized to realize and find feasible solutions for many contemporary problems. However, numerous experiments with complex, high dimensional and multimodal benchmark functions have proven that WOA has a mediocre convergence performance and still easily gets trapped in LO as the dimension of search space increases. Consequently, researchers have tried to modify the WOA in recent year. In [13], Trivedi et al. proposed a novel adaptive WOA, which integrates an adaptive technique with WOA. In [14], Emary et al. proposed an adaptive walk WOA (AWOA) based on time-varying exchanging of random walk per each whale. In [15], Hu et al. introduced inertia weight to WOA to develop the improved WOA (IWOA). In [16], Kaveh and Ghazaan suggested an enhanced WOA (EWOA) by enhancing the original formulation of WOA. In [17], Ling et al. proposed a Lévy flight trajectory-based WOA (LWOA), which employs the Lévy flight trajectory to improve the diversity of the population against premature convergence. In [18], Abdel-Basset et al. developed an integrated WOA-based technique with an exploitative engine for dealing with the scheduling of the permutation flow shop. In [19], Khalilpourazari and Khalilpourazary proposed a new hybrid optimizer, which is entitled Sine–Cosine WOA (SCWOA). In [20], Olivaa et al. developed Chaotic WOA (CWOA) for the parameters estimation of solar cells. In [21], Lévy flight and chaotic local search were embedded into WOA for practical engineering design problems. In [22], associative learning mechanism and beta hill climbing approach were introduced into WOA to improve the exploitation process of the original WOA, which was then employed to tackle a wide range of numerical optimization problems.

In this paper, we propose a variant of WOA that improves the optimization performance of original WOA by introducing three strategies including chaos-induced initialization step, chaotic exploitation with a ‘shrinking’ mode, and Gaussian mutation. The proposed improved WOA is called CCMWOA. In CCMWOA, the chaotic initialization strategy was first adopted to generate high-quality initial population, ensuring fast convergence and an excellent final solution for the proposed algorithm. Then, Gaussian mutation is involved to boost the diversity of swarm because of its narrow tail facilitates the generation of new offspring nearby the core parent. At the last, the chaotic local search (CLS) with a ‘shrinking’ mode is utilized to enhance the exploitative trends of WOA. In theory, the proposed algorithm can make a better trade-off between exploration and exploitation by properly combining these three strategies with different characteristics of WOA.

In order to verify the effectiveness of the method, the four most representative benchmark functions selected from CEC2005 and CEC2014 were used to test the performance of CCMWOA and other competitors. Those competitors include four well-established meta-heuristic algorithms: original WOA, BA, FA, MFO and four state-of-the-art Evolutionary Algorithms (EAs): particle swarm optimization (PSO) with an aging leader and challengers (ALCPSO) [23], biogeography-based learning PSO (BLPSO) [24], PSO with chaotic and Gaussian local search procedures (CGPSO) [25] and differential evolution based on CLS (DECLS) [26]. Simulation results expose that not only the proposed CCMWOA is significantly better than WOA, BA, FA, MFO algorithms but also superior to those state-of-the-art algorithms, which suggests that new operators can meaningfully enrich the efficacy of basic WOA. Moreover, CCMWOA was successfully applied to treat constrained cases including tension/compression spring design, welded beam design, and pressure vessel design. The solutions obtained by CCMWOA are superior to those obtained by other algorithms reported in recent literature, indicating that the proposed algorithm can also solve the constrained problems.

This paper is structured as follows. Section 2 briefly describes WOA, chaos-based initialization phase, Gaussian mutation, and CLS with a ‘shrinking’ mode. The proposed algorithm is explained in detail in Section 3. The experimental results of the methods on benchmark and engineering problems are presented and analyzed in Section 4. In Section 5, conclusions and some future directions are summarized.

2. Material and methods

2.1. A brief overview of WOA

This optimizer is a state-of-the-art swarm intelligence algorithm proposed by Mirjalili and Lewis [12]. This algorithm is mimicking the predation behavior of humpback whales. The mathematical model employed to simulate three stages in the predation of whales are including the search for prey, encircling prey, and bubble-net attacking. When searching for prey, the search agent randomly follows another search agent in the population. To model these facts, the position vector of search agent X can be updated with the following formula:

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X} \right| \tag{1}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \tag{2}$$

where C and A are some important coefficient factors, X_{rand} is the position vector of search agent randomly selected from the current population, t indicates the current iteration and $(\bullet)$ is an element-by-element multiplication. The A and C can

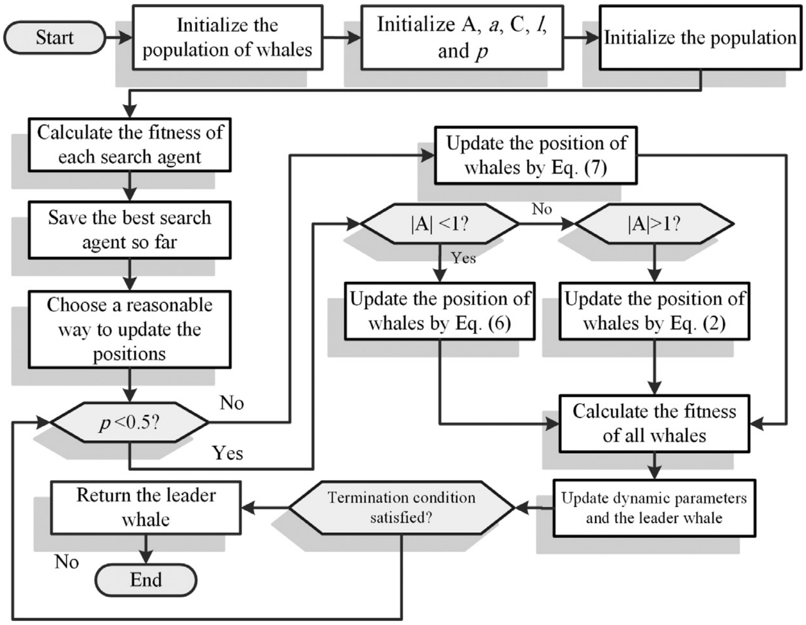

Fig. 1. Flowchart of the proposed WOA technique.

be calculated by the following formula:

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

$$\vec{C} = 2\vec{r} \tag{4}$$

where r is a random vector in [0,1] and a is a variable that decreases from 2 to 0 as iterations go on. During encircling prey, the search agent targets the currently found optimal candidate solution and updates its status based on the target. This performance can be presented as follows:

$$\vec{D} = \left|\vec{C}\cdot\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{5}$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}\cdot\vec{D} \tag{6}$$

where X^* is the position vector of the target. In the bubble-net attacking stage, the search agent moves toward the target in a spiral motion. The model is presented as follows:

$$\vec{X}(t+1) = \overrightarrow{D'}\cdot e^{bl}\cdot\cos(2\pi l) + \overrightarrow{X^*}(t) \tag{7}$$

$$D' = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{8}$$

where D' is the distance of the search agent to target, b is a constant and l is a random number in [−1,1]. In WOA, the search agent has half the probability of choosing to update its position in shrinking inclosing mode or the spiral model. The mathematical model is as follows:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}\cdot\vec{D} & if\ \ p < 0.5 \\ \overrightarrow{D'}\cdot e^{bl}\cdot\cos(2\pi l) + \overrightarrow{X^*}(t) & if\ \ p \geq 0.5 \end{cases} \tag{9}$$

where p is a random number in [0,1]. The flowchart of the conventional WOA is shown in Fig. 1.

2.2. Chaotic initialization

In the last decade, chaotic sequences have been widely employed in various areas such as parameter optimization, feature selection, and chaos control. This is due to its three unique properties of chaos: the sensitive to initial conditions, the semi-stochastic property, and ergodicity. The signal of the logistic map can be generated by Eq. (10):

$$\beta_{i+1} = \mu\beta_i * (1-\beta_i), i = 1,\ \ 2,\ldots,S-1 \tag{10}$$

where μ denotes a control parameter, often $\mu = 4$. When $\mu = 4$, chaos appears. β_1 denotes a random number between (0, 1) and S is the number of search agents.

In fact, the majority of optimizers have a stochastic nature with a statement of uniform distribution. This statement may result in divergence from the best solutions in several iterations, which increases the uncertainty of the procedure. It is obvious that in such a case, the efficacy of the basic version might be biased as well. To a certain extent, the effectiveness is affected by the initial position of whales. A well-distributed initial population ensures a faster convergence trend and a satisfactory quality of the final outcome for population-based techniques [38, 39]. Chaos can be explained as a dynamic motion with ergodicity and stochasticity that is very sensitive to early conditions. To deepen the diversity of the immature population, chaos-triggered initialization was employed in the proposed technique. The disturbance process is explained as follows:

$$X_i^c = \beta_i X_i \tag{11}$$

where X_i^c is the position of ith whale with chaotic disturbance and β_i is ith chaotic value in the chaotic sequence.

2.3. Gaussian mutation

The Gaussian mutation scheme has its roots in Gaussian normal distribution, which has attracted many researchers in evolutionary computation community from a long time ago [27]. For instance, in previous works, the Gaussian mutation has been used to augment the efficacy of the ABC [28], DE [29], PSO [30], and SSA [7] techniques. Since the tail of the Gaussian distribution is narrower, it is more likely to generate fresh offspring nearby the main parents. The search agents (whales) update their location in small steps, which makes it possible to explore every side of the feature space. In theory, the Gaussian mutation mechanism can enable the algorithm to converge faster. The Gaussian density function is obtained by Eq. (12):

$$f_{gaussian(0,\sigma^2)}(\alpha) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\alpha^2}{2\sigma^2}} \tag{12}$$

where σ^2 denotes the variance value for every whale in the swarm. In addition, this rule is reduced to create a unique $n-$dimensional randomized value by fixing the value of mean measure to zero and the value of standard deviation metric to 1. The new random value is employed to make a fresh whale, which can be obtained using Eq. (13):

$$X'(t) = X(t) \oplus (1 + G(\alpha)) \tag{13}$$

where X'denotes the location of the new whale and $G(\alpha)$ denotes a d-dimensional Gaussian step vector attained based on Gaussian density rule. In this rule, α is a Gaussian random value inside [0, 1].

2.4. CLS with a 'shrinking' mode

An exploitation engine or local search (LS) can often search within a restricted area to focus on the possible candidate optimal solutions and their neighborhood [31]. LS can be utilized to generate and refine the quality of solutions within a limited radius. Most of the optimizers have some LS processes within their main frameworks. However, in situations that the exploration depth is not enough, LS may lead to unsatisfactorily quick convergence. This event may cause another main pitfall, which is entrapment to local optima for the whole population. In such cases, the optimizer generates low-quality solutions that are far from global optimums. In the case of any hasty convergence, the merit of the CLS is that it can assist the main operations of the algorithm in avoiding the premature convergence. The reason is that there is randomicity in chaos. In this chaos-based scheme, we can shrink the feature space with regard to more function evaluations to refine the agents in later stages and enhance the exploitative propensities of CLS. This process can be drawn as follows:

$$X_k^{\prime(c)} = (1 - \lambda)X^* + \lambda(LB + \beta_k(UB - LB)) \tag{14}$$

where $X_k^{\prime(c)}$ denotes the kth new location formed by CLS, X^* represents the best whale gotten so far, β_k is kth chaos-generated element in used signal, LB and UB show the limits of space. λ shows a shrinking scale obtained by Eq. (15):

$$\lambda = 1 - \left|\frac{FEs - 1}{FEs}\right|^m \tag{15}$$

where FEs show the used function evaluations and m is utilized to manage the shrinking rate.

3. Proposed WOA-based method

3.1. Improved WOA (CCMWOA)

In this section, the proposed method will be described in detail. This algorithm introduces three strategies based on the original WOA to improve the global search ability of the original method. In the proposed algorithm, the chaotic initialization strategy was first adopted to generate high-quality initial population, ensuring fast convergence and an excellent final

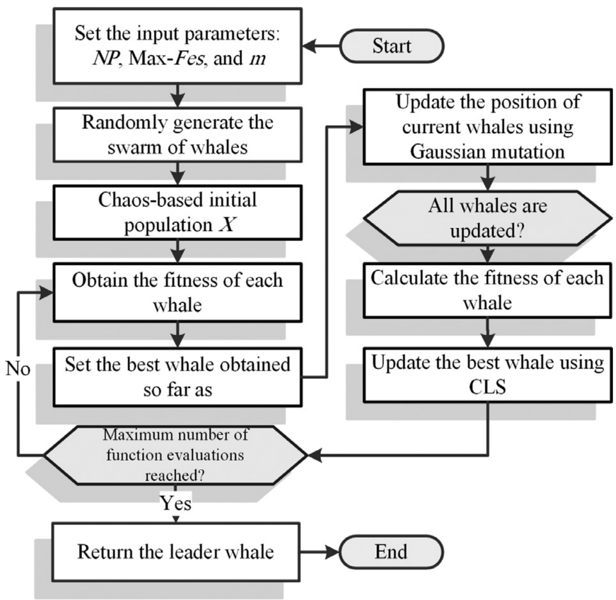

Fig. 2. Flowchart of the proposed CCMWOA technique.

solution for the proposed algorithm. Then, we borrow the merits of Gaussian mutation to deepen the diversity index of the swarm of whales because its narrow tail facilitates the appearance of new offspring nearby to the core parent. As a result, the search rule can take smaller steps, which assists the whole search to scan the target space, more deeply. The Gaussian mutation strategy can be described as follows:

$$\overrightarrow{X'}(t) = \overrightarrow{X^A}(t) \oplus (1 + G(\alpha)) \tag{16}$$

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^A}(t) & f\left(\overrightarrow{X^A}(t)\right) > f\left(\overrightarrow{X'}(t)\right) \\ \overrightarrow{X'}(t) & otherwise \end{cases} \tag{17}$$

where $X^A(t)$ is the new position vector calculated by Eq. (9) and $f(\bullet)$ is the fitness function. When the updating of locations is complete, a CLS with a 'shrinking' mode will be performed. This strategy can help to escape rash convergence [38]. Fig. 2 shows the flowchart of the CCMWOA. The Pseudo code of the enhanced optimizer is also reported in Algorithm 1 as follows:

3.2. Computational complexity of CCMWOA

The computational complexity of CCMWOA depends on population size (n), the dimensions of the problem (d), and the maximum function evaluations (m). The number of iterations (t) is determined by a maximum number of function evaluations and the size of population: $t = m/2n$. Therefore, we have O(CCMWOA)=O(Chaotic initialization step)+$t\times$(O(Estimate the fitness of whales)+O(Update the position of all whales with Gaussian mutation)+O(CLS with a 'shrinking' mode)). The complexity of chaos-based initialization phase is $O(2n\times d + 2n \times \log(2n))$. Assessing the fitness of all whales is $O(n \times d)$. Updating the position of all agents with the Gaussian mutation is $O(n \times d)$. CLS with a 'shrinking' mode is $O(n \times d)$. Therefore, we have: $O(\text{CCMWOA}) = O(2n\times d + 2n \times \log(2n) + (m/2n) \times (3 \times n \times d)) \approx O(n \times d + n \times \log n + m \times d)$.

4. Experimental studies

In this section, first, simulation experiments are performed on four standard cases to validate the effectiveness of the proposed CCMWOA and then, the algorithm is applied to three well-regarded test problems. All experiments are conducted using MATLAB R2014a software under a Windows Server 2012 R2 operating system and the hardware platform used is configured with Intel (R) Xeon(R) Sliver 4110 CPU (2.10 GHz) and 16 GB RAM.

4.1. Function optimization

4.1.1. Benchmark function and performance evaluation measures

Experiments were carried out over four classical benchmark cases with different features. Among these functions, F1 is a unimodal and F2 is a multimodal problem, which are all selected from the classic benchmark function set CEC2005 [32].

Algorithm 1 Pseudo code of CCMWOA.

```
Begin
  Set the number of whales NP, the maximum function evaluations Max_FEs and m;
  Randomly generated whales X_i(i = 1,2,...,n);
  Chaotic initial population X to generate X^c according to Eq. (11);
  Select the NP best individuals from the X and X^c populations as the initial population of WOA;
  while (FEs ≤Max_FEs)
    Calculate the fitness of each whale;
    X*= the best whale;
    for each whale
      Update a, A, C, l, and p;
      if (p < 0.5)
        if (|A|<1)
          Update the position of the current whale by Eq. (6)
        else if (|A|<1)
          Select a random search agent X_rand
          Update the position of the current whale by Eq. (2)
        end if
      else if (p ≥ 0.5)
        Update the position of the current whale by Eq. (7)
      end if
      Update the position of the current whale with Gaussian mutation strategy;
    end for
    Preform the CLS with a 'shrinking' mode
    Check if any whale goes outside the limits of search space and adjust it
    Update the number of FEs
  end while
  return X*;
End
```

F3 and F4 are both composition functions taken from CEC2014 [33]. The definitions and description of benchmark functions are as follows:

Schwefel's Function (F1):

$$F_1(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$$

where D is dimension, $x \in [-100, 100]^D$, global optimum $x^*=0$.

Ackley's Function (F2):

$$F_2(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$$

where D is dimension, $x \in [-32, 32]^D$, global optimum $x^*=0$.

Composition Function 5 (F3), composed of five different basic functions:

- Rotated variants of HGBat
- Rastrigin's, Schwefel's
- Weierstrass
- High Conditioned Elliptic cases

Its search range is $[-100,100]^D$ where D is dimension and the global optimum is 2700.

Composition Function 8 (F4), composed of three different hybrid cases:

- Hybrid functions 4
- Hybrid functions 5
- Hybrid functions 6

Its search range is $[-100,100]^D$ where D is dimension and the global optimum is 3000.

For more detailed information on the above functions, please see the original literature.

To attain unbiased results, all algorithms were directed under identical settings: population size and maximum function evaluations are set to 30 and 10,000, respectively. For each benchmark function, all algorithms were tested 30 times, independently. The best result, worst result, mean of results and standard deviation (std.) of the results of these 30 independent experiments are recorded in the table. The Friedman test [34] is used to evaluate the results of all algorithms on the benchmark functions and give their rankings. The Friedman test is a non-parametric statistical test comparison test processed to spot differences between multiple test results. The Friedman test allocates the first ranks to those approaches that show a better performance. The Pseudo code is shown at Algorithm 2.

Algorithm 2 Pseudo code for calculating the Friedman rank.

Begin
Input a matrix with n rows (the blocks), k columns (the treatments) $\{x_{ij}\}_{n\times k}$;
for $i = 1{:}n$
Calculate the rank r_{ij} of x_{ij} within block i;
end for
Calculate mean rank $\bar{r}_j = \frac{1}{n}\sum_i^n r_{ij}$;
return $\bar{r}_j$;
End

Table 1
Results of various WOAs on benchmark functions.

Function	Metric	CCMWOA	WOA	CWOA	GWOA	CLSWOA
F1	worst	0.00E+00	1.79E+02	2.47E−13	0.00E+00	2.18E−02
	best	0.00E+00	2.98E−04	5.60E−73	0.00E+00	5.39E−07
	mean	0.00E+00	2.33E+01	9.24E−15	0.00E+00	4.65E−03
	std.	0.00E+00	4.70E+01	4.53E−14	0.00E+00	5.66E−03
	rank	1.5	5	3	1.5	4
F2	worst	8.88E−16	7.99E−15	7.99E−15	8.88E−16	7.99E−15
	best	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
	mean	8.88E−16	3.38E−15	2.43E−15	8.88E−16	2.66E−15
	std.	0.00E+00	1.90E−15	2.41E−15	0.00E+00	2.03E−15
	rank	2.26	3.9	3.15	2.26	3.41
F3	worst	2.90E+03	4.11E+03	3.90E+03	2.90E+03	4.12E+03
	best	2.90E+03	3.13E+03	2.90E+03	2.90E+03	2.90E+03
	mean	2.90E+03	3.79E+03	2.94E+03	2.90E+03	3.05E+03
	std.	0.00E+00	3.35E+02	1.87E+02	0.00E+00	3.67E+02
	rank	2.08	4.9	2.63	2.08	3.3
F4	worst	3.20E+03	5.16E+05	3.04E+05	1.73E+05	4.49E+05
	best	3.20E+03	2.70E+04	3.20E+03	3.20E+03	3.20E+03
	mean	3.20E+03	1.12E+05	1.32E+04	8.87E+03	1.06E+05
	std.	0.00E+00	9.85E+04	5.49E+04	3.10E+04	1.09E+05
	rank	1.73	4.56	2.7	1.8	4.2
Average rank		1.90	4.59	2.87	1.91	3.73
Overall rank		1	5	3	2	4

4.1.2. The impact of chaotic initialization, Gaussian mutation and chaotic local search

According to Section 3, three strategies, namely, chaotic initialization, Gaussian mutation, and chaotic local search are combined with the original WOA. In order to investigate the impact of each strategy, four different improved WOAs and the original WOA were benchmarked on benchmark functions. The four improved WOAs are denoted as CWOA, GWOA, CLSWOA, and CCMWOA respectively. For CWOA, it means that WOA is only combined with chaotic initialization strategy. For GWOA, it is combined with WOA and Gaussian mutation strategy. For CLSWOA, it means that WOA is only combined with chaotic local search strategy. For CCMWOA, it is the combination of WOA and all the three strategies.

The results of various WOAs on benchmark functions are presented in Table 1. It can be seen from the table that CCMWOA achieved the best solution for the four problems. This indicates that the combination of the three strategies can take full advantage of the exploratory and exploitative leanings of the proposed mechanisms and core operators of WOA. For the original WOA, its overall Friedman test rank is behind CWOA, GWOA, and CLSWOA, which means all three strategies play an indispensable role in the improvement of WOA's global searching capability. The GWOA ranked second, indicating that the Gaussian mutation strategy is crucial to the improvement of WOA. For CWOA and CLSWOA, the results of CWOA on four functions are better than CLSWOA, which shows chaotic initialization strategy improves the performance of WOA more effectively than the chaotic local search strategy. The reason for the same best, mean, worst, and standard deviation of some methods including CCMWOA in Table 1 is that those variants have obtained the same quality results in all runs, which indicates that they have shown a more stable performance.

For statistical tests, the paired Wilcoxon signed-rank test was also employed in this study [35]. The performance of CCMWOA is considered to be significantly superior to other algorithms only when the p-value gotten by the test is less than 0.05.

From Table 2, it can be found that CCMWOA has significant advantages compared with original WOA, GWOA, and CLSWOA. For GWOA, the advantages of CCMWOA are not significant, but with reference to Table 1, CCMWOA is still advantageous.

Accordingly, it can be concluded that the efficacy of the original WOA can be considerably enriched by accurately assimilating these mechanisms to effectively scale the exploratory and exploitative propensities of WOA.

Table 2
The p-values of CCMWOA compared with four WOAs.

Function	WOA	CWOA	GWOA	CLSWOA
F1	1.73E−06	1.73E−06	1.00E+00	1.73E−06
F2	1.19E−05	1.95E−03	1.00E+00	1.22E−04
F3	1.73E−06	5.00E−01	1.00E+00	6.25E−02
F4	1.73E−06	2.44E−04	1.00E+00	3.79E−06

Table 3
The results obtained by CCMWOA and basic algorithms.

Function	Metric	CCMWOA	WOA	BA	FA	MFO
F1	worst	0.00E+00	1.21E+03	3.49E+01	2.50E+04	7.00E+04
	best	0.00E+00	2.23E−03	1.62E+01	1.45E+04	5.00E+03
	mean	0.00E+00	5.24E+01	2.45E+01	2.01E+04	2.04E+04
	std.	0.00E+00	2.20E+02	4.74E+00	2.79E+03	1.41E+04
	rank	1.00	2.20	2.80	4.50	4.50
F2	worst	8.88E−16	7.99E−15	2.00E+01	1.64E+01	2.00E+01
	best	8.88E−16	8.88E−16	3.61E+00	1.49E+01	4.44E−15
	mean	8.88E−16	3.61E−15	4.43E+00	1.59E+01	1.66E+01
	std.	0.00E+00	2.02E−15	2.94E+00	3.54E−01	6.71E+00
	rank	1.15	1.87	3.20	4.13	4.65
F3	worst	2.90E+03	4.14E+03	4.31E+03	3.92E+03	3.88E+03
	best	2.90E+03	3.12E+03	3.78E+03	3.31E+03	3.12E+03
	mean	2.90E+03	3.89E+03	4.05E+03	3.79E+03	3.67E+03
	std.	0.00E+00	2.69E+02	1.31E+02	9.77E+01	1.43E+02
	rank	1.00	4.17	4.60	3.00	2.23
F04	worst	3.20E+03	2.77E+05	2.51E+04	2.52E+05	1.79E+05
	best	3.20E+03	2.52E+04	5.64E+03	1.16E+05	9.38E+03
	mean	3.20E+03	8.28E+04	1.20E+04	1.79E+05	3.93E+04
	std.	0.00E+00	6.01E+04	4.49E+03	3.89E+04	3.49E+04
	rank	1.00	3.90	2.03	4.87	3.20
Mean rank		1.04	3.03	3.16	4.13	3.65
Overall rank		1	2	3	5	4

Table 4
The p-values from the signed-rank test for the CCMWOA as against other techniques.

Function	WOA	BA	FA	MFO
F1	1.73E−06	1.73E−06	1.73E−06	1.73E−06
F2	1.02E−05	1.73E−06	1.73E−06	1.73E−06
F3	1.73E−06	1.73E−06	1.73E−06	1.73E−06
F4	1.73E−06	1.73E−06	1.73E−06	1.73E−06

4.1.3. Comparative study

To verify the efficacy of CCMWOA, it was compared to several recent successful algorithms: BA, FA, and MFO algorithms. The values of parameters for these algorithms used for comparison were fixed with regard to their original settings. For CCMWOA, its parameter $a\in[0\ 2]$ and $m = 1500$; For WOA, its parameter $a\in[0\ 2]$; For BA, its parameter $Q\in[0\ 2]$, $A = 0.5$ and $r = 0.5$; For FA, its parameter $\beta_0=1$, $\alpha\in[0\ 1]$ and $\gamma=1$; For MFO, its parameter $a\in[-1\ -2]$ and $b=1$.

The results obtained by CCMWOA and basic algorithms on functions are shown in Table 3. It can be seen from the table that CCMWOA outperformed all other techniques in dealing with four problems. The CCMWOA not only achieved the best solutions but also has the minimum std., which reveals that CCMWOA has superior optimization capabilities and high LO escaping capacities on different type problems. The reason for the same best, mean, worst, and zero standard deviation of CCMWOA in Table 3 is that it shows a more stable performance in different runs.

As can be seen in Table 4, all p-values are less than 0.05, which indicate that the performance of CCMWOA is statistically better than other peers on benchmark functions. These values show the improvements are significant at the same level.

Convergence rate is a vital criterion for investigating the efficacy and exploratory/exploitative powers of evolutionary algorithms. In order to more comprehensively verify the efficacy and searching trends of CCMWOA, the convergence curves of all algorithms are recorded and presented in Fig. 3. It can be perceived that the convergence rates of CCMWOA are faster than other competitors in all cases. In dealing with F1, CCMWOA always maintains very fast convergence speed. For F2, F3, and F4, CCMWOA converges very quickly to the best solution.

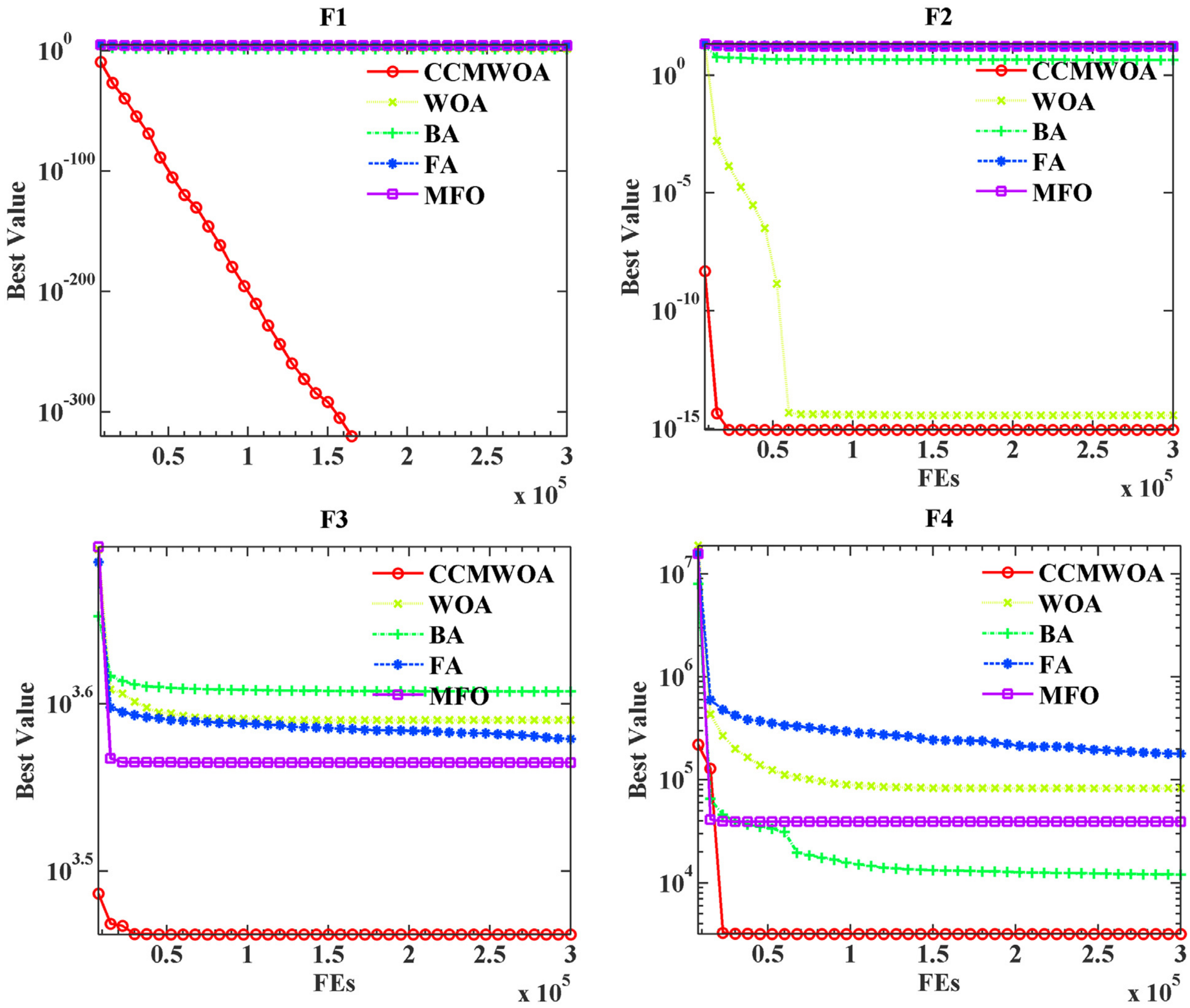

Fig. 3. Convergence trends for F1–F4 problems.

4.1.4. Comparison with state-of-the-art EAs

In this section, CCMWOA was further compared with some state-of-the-art EAs including ALCPSO [23], BLPSO [24], CGPSO [25], and DECLS [26]. The values of parameters for the above algorithms used for comparison were selected with regard to the original articles.

Table 5 presents the results found by CCMWOA and advanced EAs. It can be realized from the table that CCMWOA is superior to its state-of-the-art competitors on these benchmarks. For F1, CCMWOA achieved to the global optimal solution with zero std. values, while other counterparts failed to obtain a satisfactory solution. According to the overall rank, the BLPSO has a strong competitive edge but its performance on F1 is not good. CCMWOA won first place on all benchmark functions, which reveals its very competitive performance. The reason for the same best, mean, worst, and zero standard deviation of CCMWOA in Table 5 is that it shows a more stable performance in various runs.

The p-values are shown in Table 6. It can be understood that all p-values are less than 0.05, which means the performance of CCMWOA is significantly better than these state-of-the-art EAs. The same values show the improvements are significant at the same level.

Regarding the related convergence trends in Fig. 4, we detect that the CCMWOA has the fastest convergence speed in dealing with all benchmark problems. For F1, CCMWOA converges to the global optimal solution with a very fast convergence speed. For F2, F3, and F4, CCMWOA converges to the best solution after a few function evaluations. The first reason is that the chaotic initialization provides a high diversity initial population, which enables the CCMWOA to start the process with a boosted exploration trend. Then, it can perform a smooth transition from exploration to exploitation due to the role of CLS around the explored agents. Based on CLS, it has a higher chance to evade the immature convergence, which leads

Table 5
The results obtained by CCMWOA and state-of-the-art EAs.

Function	Metric	CCMWOA	DECLS	CGPSO	ALCPSO	BLPSO
F1	**worst**	0.00E+00	1.10E−01	2.89E−01	6.20E−10	1.69E−01
	best	0.00E+00	1.36E−09	1.53E−07	6.44E−14	4.09E−03
	mean	0.00E+00	6.67E−03	1.11E−01	5.54E−11	4.80E−02
	std.	0.00E+00	2.55E−02	8.14E−02	1.55E−10	4.02E−02
	rank	1.00	3.20	4.57	2.00	4.23
F2	**worst**	8.88E−16	7.99E−15	6.36E−05	2.27E+00	7.99E−15
	best	8.88E−16	4.44E−15	6.88E−07	7.99E−15	4.44E−15
	mean	8.88E−16	7.88E−15	1.76E−05	1.21E+00	4.80E−15
	std.	0.00E+00	6.49E−16	1.61E−05	7.76E−01	1.08E−15
	rank	1.00	2.98	4.23	4.70	2.08
F3	**worst**	2.90E+03	3.35E+03	3.83E+03	3.66E+03	3.10E+03
	best	2.90E+03	2.90E+03	2.90E+03	3.10E+03	3.00E+03
	mean	2.90E+03	3.02E+03	3.04E+03	3.46E+03	3.01E+03
	std.	0.00E+00	1.76E+02	2.76E+02	1.75E+02	2.58E+01
	rank	1.00	2.93	2.93	4.77	3.37
F4	**worst**	3.20E+03	9.93E+03	3.45E+04	1.19E+05	4.90E+03
	best	3.20E+03	3.45E+03	3.21E+03	5.31E+03	3.76E+03
	mean	3.20E+03	7.13E+03	1.00E+04	1.95E+04	4.27E+03
	std.	0.00E+00	1.26E+03	9.08E+03	2.36E+04	2.92E+02
	rank	1.00	3.83	3.17	4.43	2.57
Mean rank		1.00	3.24	3.73	3.98	3.06
Overall rank		1	3	4	5	2

Table 6
The calculated p-values from the signed-rank test for the CCMWOA versus other optimizers.

Function	DECLS	CGPSO	ALCPSO	BLPSO
F1	1.73E−06	1.73E−06	1.73E−06	1.73E−06
F2	6.8E−08	1.73E−06	1.72E−06	1.44E−07
F3	1.73E−06	1.73E−06	1.73E−06	9.13E−07
F4	1.73E−06	1.73E−06	1.73E−06	1.73E−06

to more high-quality solutions. The Gaussian mutation also accelerates the convergence trends of proposed CCMWOA in finding high-quality solutions. The CCMWOA also inherits the core advantages of WOA over variants of PSO and DE.

4.1.5. *The impact of parameter* m *on CCMWOA*

In experiments, we monitored the impact of parameter m on CCMWOA based on the results obtained by CCMWOA on four benchmark functions when m is set to 100, 500, 1000, 1500, 2000 and 2500, respectively. The experimental results showed that the optimal solutions found by the CCMWOA with different values of parameter m are almost the same. This illustrates that m has little impact on CCMWOA's performance. Finally, we fixed $m = 1500$ as the parameter setting for CCMWOA according to the literature [26].

4.2. *Engineering problems*

In this section, the CCMWOA was adopted to realize and study three constrained engineering design problems including welded beam design, tension/compression spring design, and pressure vessel design cases [36]. These problems, often with equality and inequality constraints, require a constraint-handling approach. As is described in [37], CCMWOA with a death penalty approach was employed to realize three constrained problems.

4.2.1. *Tension/compression spring design case*

In this case, we are interested to optimize (minimize) the weight of a target TCS. In this problem, we have some unknown variables: wire diameter (d), mean coil diameter (D), and the number of active coils (N). The mathematical model can be explained as follows:

$$\text{Consider } \vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} d & D & N \end{bmatrix}$$

$$\text{Object} \quad f(\vec{x})_{min} = x_1^2 x_2 x_3 + 2x_1^2 x_2$$

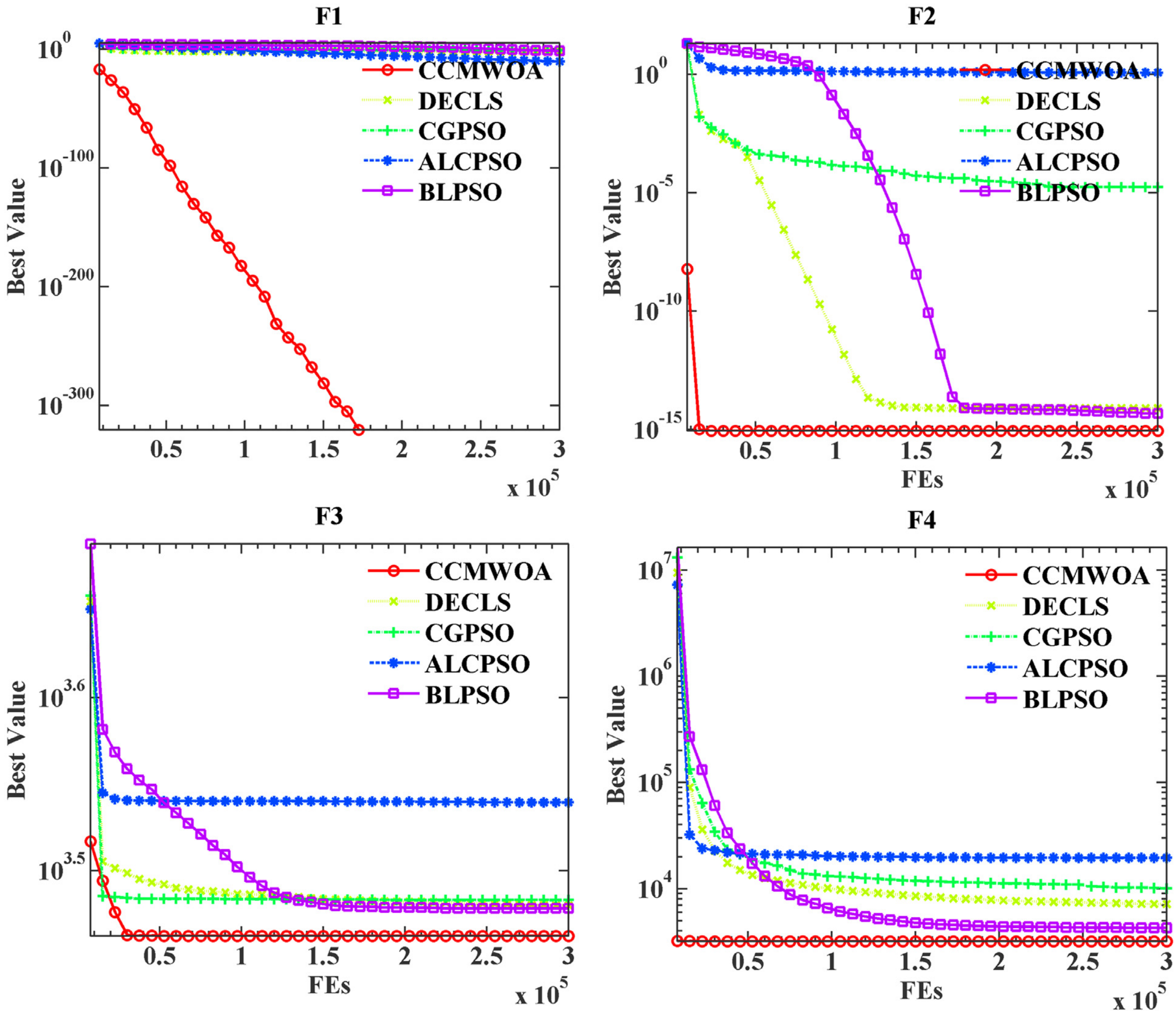

Fig. 4. Convergence trends for used problems.

Subject to

$$h_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$h_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} \le 0$$

$$h_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^3 x_3} \le 0$$

$$h_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variable ranges: $0.05 \le x_1 \le 2.00$, $0.25 \le x_2 \le 1.30$, $2.00 \le x_3 \le 15.0$

For this inequality constraint problem, there has been a lot of literature research on it. In [38], He and Wang employed a PSO-based solver to deal with this problem. The optimum weight they get is 0.0126747. In [12], the original WOA and Gravitational Search Algorithm (GSA) are used to solve this problem at the same time and the optimal results are 0.0126763 and 0.0127022 respectively. Moreover, Improved Harmony Search (IHS) [39], DE [40], and Ray Optimization (RO) [41] algorithms were also used to solve this problem.

Table 7 shows the results of CCMWOA on this problem and the results reported in other literature. From the table, it can be seen that CCMWOA yields the optimum weight of 0.0126660. The result obtained by CCMWOA is better than

Table 7
Comparison of CCMWOA optimization results with literature for tension/compression spring design problem.

Algorithm	Optimum variables			Optimum weight
	d	*N*	*D*	
CCMWOA	0.051843	11.07410	0.360444	**0.0126660**
WOA (Table 8 Section 4.1 p. 59 in [12])	0.051207	12.0043032	0.345215	0.0126763
GSA (Table 8 Section 4.1 p. 59 in [12])	0.050276	13.525410	0.323680	0.0127022
PSO (Table 3 Section 4.2 p. 95 in [38])	0.015728	11.244543	0.357644	0.0126747
RO (Table 5 Section 4.2.1 p. 292 in [41])	0.051370	11.762790	0.349096	0.0126788
IHS (Table 2 Section 3.1 p. 1573 in [39])	0.051154	12.076432	0.349871	0.0126706
DE (Table 4 Section 4.3 p. 348 in [40])	0.051609	11.410831	0.354714	0.0126702

that obtained by the original WOA, indicating that these three modes can effectively enhance the performance of the WOA. Compared with other algorithms, the solution obtained by CCMWOA is also the best, indicating that CCMWOA is an effective tool for this case.

4.2.2. Welded beam design case

For this case, we minimize the manufacturing cost of a welded beam. There are some constraints such as bucking load (P_c), shear stress (τ), bending stress (θ), and the deflection (δ). The cost has variables including the thickness (h), length (l), height (t), and the thickness (b) of the bar. The mathematical model of this case is as follows:

$$Consider\ \vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} = \begin{bmatrix} h & l & t & b \end{bmatrix}$$

$$Object\ \ f(\vec{x})_{min} = 1.10471 x_2 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

$$\begin{aligned} \text{Subject to} \quad & g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0 \\ & g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0 \\ & g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0 \\ & g_4(\vec{x}) = x_1 - x_4 \le 0 \\ & g_5(\vec{x}) = P - P_C(\vec{x}) \le 0 \\ & g_6(\vec{x}) = 0.125 - x_1 \le 0 \\ & g_7(\vec{x}) = 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0 \end{aligned}$$

Variable ranges $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$, $0.1 \le x_4 \le 2$

Where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$ and $\tau' = \frac{P}{\sqrt{2}x_1x_2}$ $\tau'' = \frac{MR}{J}$ $M = P(L + \frac{x_2}{2})$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4 {x_3}^2}, \quad \delta(\vec{x}) = \frac{6PL^3}{E x_3^2 x_4}$$

$$P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 60001\text{b}, \ \ L = 14\text{in}.\delta_{max} = 0.25\ in.$$

$$E = 30\ \times 10^6 psi, \ \ G = 12\ \times 10^6 psi$$

$$\tau_{max} = 13600 psi, \ \ \sigma_{max} = 30000 psi$$

Table 8
Comparison of CCMWOA optimization results with literature for the welded beam design problem.

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
CCMWOA	0.19633	3.4272	9.0422	0.2057	**1.7054**
WOA (Table 10 Section 4.2 p. 60 in [12])	0.205396	3.484293	9.037426	0.206276	1.730499
GSA (Table 10 Section 4.2 p. 60 in [12])	0.182129	3.856979	10.00000	0.202376	1.879952
CBO (Table 1 Section 4.1 p. 22 in [42])	0.205722	3.470410	9.037276	0.205735	1.724663
RO (Table 7 Section 4.2.2 p. 292 in [41])	0.203687	3.528467	9.004233	0.207241	1.735344
IHS (Table 5 Section 3.3 p. 1575 in [39])	0.20573	3.47049	9.03662	0.20573	1.7248

Table 9
Comparison of CCMWOA optimization results with literature for the pressure vessel design problem.

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
CCMWOA	0.779661	0.385611	40.34738	199.6141	**5895.2039**
WOA (Table 12 Section 4.3 p. 61 in [12])	0.812500	0.437500	42.098209	176.638998	6059.7410
IHS (Table 4 Section 3.2 p. 1574 in [39])	1.125000	0.625000	58.29015	43.69268	7197.7300
PSO (Table 5 Section 4.3 p. 96 in [38])	0.812500	0.437500	42.091266	176.746500	6061.0777
ES (Table 9 Section 5.4 p. 463 in [43])	0.812500	0.437500	42.098087	176.640518	6059.7456
DE (Table 6 Section 4.4 p. 348 in [40])	0.812500	0.437500	42.098411	176.637690	6059.7340
ACO (Table 7 Section 5.4 p. 171 in [44])	0.812500	0.437500	42.098353	176.637751	6059.7258

Welded beam design case also has been studied in many papers. As reported in [12], the results obtained by WOA and GSA on this issue are 1.730499 and 1.879952, respectively. In [39], Mahdavi et al. applied the IHS to solve this problem and attained the optimum cost of 1.735344. Furthermore, both RO [41] and Colliding Bodies Optimization (CBO) [42] are used by Kaveh and co-workers to solve this problem.

The results CCMWOA has achieved on this issue are shown in Table 8, which also presents the results reported in the literature. From the table, we detected that CCMWOA provides an optimum design with a manufacturing cost of 1.7054. The manufacturing cost of the design provided by the original WOA is significantly higher than that of CCMWOA, indicating that CCMWOA has better optimization ability than WOA. It can also be seen that CCMWOA outperformed all the other algorithms.

4.2.3. Pressure vessel design case

The purpose of this case is to minimize the total cost of the material, forming, and welding of the cylindrical vessel. The cost has four variables: the thickness of the shell (T_s), the thickness of the head (T_h), the inner radius (R), and the length of the cylindrical part (L). The formulation of this case is shown as follows:

$$\text{Consider } \vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} = \begin{bmatrix} T_s & T_h & R & L \end{bmatrix}$$

$$\text{Object } \; f(\vec{x})_{min} = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$$

$$\begin{aligned} \text{Subjectto } \; & g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0 \\ & g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0 \\ & g_3(\vec{x}) = -\pi x_4 x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ & g_4(\vec{x}) = x_4 - 240 \le 0 \end{aligned}$$

Variable ranges $0 \le x_1 \le 99$, $0 \le x_2 \le 99$, $10 \le x_3 \le 200$, $10 \le x_4 \le 200$

Like the previous two problems, this engineering optimization problem has also been extensively studied. In [12], Mirjalili and Lewis employed WOA to deal with this problem and achieved the optimum cost of 6059.7410. In [39], Mahdavi et al. obtained the best cost of 7197.7300 on this problem by using the IHS algorithm. In [38], He and Wang used PSO and obtained the cost of 6061.07777. In addition, PSO [38], ES [43], DE [40] and ACO [44] are also used to solve this problem.

The detailed results obtained by the above algorithms and our proposed CCMWOA on this problem are shown in Table 9. From the table, we can see that the result obtained by CCMWOA is significantly smaller than the results obtained by other algorithms. Compared with original WOA, CCMWOA has the ability to get a better design for this problem, which reveals proposed CCMWOA has a stronger global search capability and it can solve such constrained problems, effectively.

5. Conclusions and future works

In the study, we propose an improved CCMWOA method, which combines the chaotic initialization strategy, Gaussian mutation, and CLS to enhance the exploratory and exploitative leanings of the original WOA. In order to study the impact of these three strategies on the original WOA, four different improved WOAs and the original WOA were compared on representative benchmark functions. The experimental results expose that all three strategies play an indispensable role in the improvement of the original WOA's optimizing performance and the Gaussian mutation strategy is crucial. In the experiments for comparing the performance of CCMWOA with recent algorithms and state-of-the-art EAs, CCMWOA has shown its very competitive global search capability, indicating that the combining of the three strategies is effective. Moreover, the proposed algorithm was successfully applied to tackle welded beam design, tension/compression spring design, and pressure vessel design cases. The results of CCMWOA are superior to those obtained by other algorithms, indicating that the proposed algorithm has an improved exploration and exploitation capabilities and can be a good approach for effectively solving the constrained problems.

For future directions, there are many aspects that can be investigated. First, the proposed algorithm can be further studied, and its global search ability can be further enhanced by introducing a new mechanism. The second proposed algorithm can be used to combine optimization problems and feature selection problems. Moreover, the proposed algorithm can be applied to the machine learning model optimization problem to improve the learning efficiency and prediction accuracy of the machine learning model.

Acknowledgments

This research was funded by Zhejiang Provincial Natural Science Foundation of China (LY17F020012), Science and Technology Plan Project of Wenzhou of China (ZG2017019), Medical and Health Technology Projects of Zhejiang province (2019315504) and Zhejiang University Students Science and Technology Innovation Activity Plan (2018R429051).

References

[1] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li, S. Mirjalili, Binary dragonfly optimization for feature selection using time-varying transfer functions, Knowl.-Based Syst. 161 (2018) 185–204.
[2] H. Faris, A.-Z. Ala'M, A.A. Heidari, I. Aljarah, M. Mafarja, M.A. Hassonah, H. Fujita, An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks, Inf. Fusion 48 (2019) 67–83.
[3] H.l. Chen, B. Yang, S.j. Wang, G. Wang, H.z. Li, W.b. Liu, Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy, Appl. Math. Comput. 239 (2014) 180–197.
[4] W. Deng, H. Zhao, X. Yang, J. Xiong, M. Sun, B. Li, Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment, Appl. Soft Comput. J. 59 (2017) 288–302.
[5] M. Wang, H. Chen, H. Li, Z. Cai, X. Zhao, C. Tong, J. Li, X Xu, Grey wolf optimization evolving kernel extreme learning machine: application to bankruptcy prediction, Eng. Appl. Artif. Intell. 63 (2017) 54–68.
[6] R. Abbassi, A. Abbassi, A.A. Heidari, S. Mirjalili, An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models, Energy Convers. Manag. 179 (2019) 362–372.
[7] Q. Zhang, H. Chen, A.A. Heidari, X. Zhao, Y. Xu, P. Wang, Y. Li, C. Li, Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers, IEEE Access, 10.1109/ACCESS.2019.2902306 (2019) 1–1.
[8] Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu, C. Li, Chaos enhanced bacterial foraging optimization for global optimization, IEEE Access 6 (2018) 64905–64919.
[9] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, D. Liu, Evolving support vector machines using fruit fly optimization for medical data classification, Knowl.-Based Syst. 96 (2016) 61–75.
[10] W. Deng, J. Xu, H. Zhao, an improved ant colony optimization algorithm based on hybrid strategies for scheduling problem, IEEE Access (2019), doi:10.1109/ACCESS.2019.2897580.
[11] J. Luo, H. Chen, Q. Zhang, Y. Xu, H. Huang, X. Zhao, An improved grasshopper optimization algorithm with application to financial stress prediction, Appl. Math. Model. 64 (2018) 654–668.
[12] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[13] I.N. Trivedi, J. Pradeep, J. Narottam, K. Arvind, L. Dilip, A novel adaptive whale optimization algorithm for global optimization, Indian J. Sci. Technol. 9 (2016).
[14] E. Emary, H.M. Zawbaa, M.A. Salam, A proposed whale search algorithm with adaptive random walk, in: Proceedings of the Thirteenth IEEE International Conference on Intelligent Computer Communication and Processing, ICCP, 2017, pp. 171–177. 2017.
[15] H. Hu, Y. Bai, T. Xu, Improved whale optimization algorithms based on inertia weights and theirs applications, Int. J. Circuits Syst. Signal Process. 11 (2017) 12–26.
[16] A. Kaveh, M.I. Ghazaan, Enhanced whale optimization algorithm for sizing optimization of skeletal structures, Mech. Based Des. Struct. Mach. 45 (2017) 345–362.
[17] Y. Ling, Y. Zhou, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for global optimization, IEEE Access 5 (2017) 6168–6186.
[18] M. Abdel-Basset, G. Manogaran, D. El-Shahat, S. Mirjalili, A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem, Fut. Gen. Comput. Syst. 85 (2018) 129–145.
[19] S. Khalilpourazari, S. Khalilpourazary, SCWOA: an efficient hybrid algorithm for parameter optimization of multi-pass milling process, J. Ind. Prod. Eng. 35 (2018) 135–147.
[20] D. Olivaa, A. Mohamed, A. Hassaniend, Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, Appl. Energy 200 (2017) 141154.
[21] H. Chen, Y. Xu, M. Wang, X. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, Appl. Math. Model. 71 (2019) 45–59.
[22] A.A. Heidari, I. Aljarah, H. Faris, H. Chen, J. Luo, S. Mirjalili, An enhanced associative learning-based exploratory whale optimizer for global optimization, Neural Comput. Appl. (2019), doi:10.1007/s00521-019-04015-0.
[23] W.N. Chen, J. Zhang, Y. Lin, N. Chen, Z.H. Zhan, S.H. Chung, Y. Li, Y.H. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2013) 241–258.
[24] X. Chen, H. Tianfield, C. Mei, W. Du, G. Liu, Biogeography-based learning particle swarm optimization, Soft Comput. 21 (2017) 7519–7541.

[25] D. Jia, G. Zheng, B. Qu, M.K. Khan, A hybrid particle swarm optimization algorithm for high-dimensional problems, Comput. Ind. Eng. 61 (2011) 1117–1122.
[26] D. Jia, G. Zheng, M.K. Khan, An effective memetic differential evolution algorithm based on chaotic local search, Inf. Sci. 181 (2011) 3175–3187.
[27] T. Bäck, H.P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evol. Comput. 1 (1993) 1–23.
[28] X. Cheng, M. Jiang, An improved artificial bee colony algorithm based on Gaussian mutation and chaos disturbance, Lect. Notes Comput. Sci. 7331 (2012) 326–333.
[29] C. Jena, M. Basu, C.K. Panigrahi, Differential evolution with Gaussian mutation for combined heat and power economic dispatch, Soft Comput. 20 (2016) 681–688.
[30] N. Higashi, H. Iba, Particle swarm optimization with Gaussian mutation, in: Proceedings of the 2013 Swarm Intelligence Symposium, SIS '03, 2013, pp. 72–79.
[31] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Fut. Gen. Comput. Syst. 97 (2019) 849–872.
[32] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, KanGAL report, 2005005, 2005.
[33] J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria For the CEC 2014 Special Session and Competition On Single Objective Real-Parameter Numerical Optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
[34] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.
[35] D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization, J. Heurist. 15 (2009) 617–644.
[36] A.A. Heidari, R.A. Abbaspour, A.R. Jordehi, An efficient chaotic water cycle algorithm for optimization tasks, Neural Comput. Appl. 28 (2017) 57–85.
[37] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (2002) 1245–1287.
[38] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (2007) 89–99.
[39] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.
[40] FZ Huang, L. Wang, H.E. Qie, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (2007) 340–356.
[41] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112–113 (2012) 283–294.
[42] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization: a novel meta-heuristic method, Comput. Struct. 139 (2014) 18–27.
[43] E. Mezura-Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, Int. J. Gen. Syst. 37 (2008) 443–473.
[44] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. 27 (2010) 155–182.