Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3025

A MULTI-STRATEGY DEVELOPED MARINE PREDATOR ALGORITHM AND ITS APPLICATION IN PRESSURE VESSEL

Kai Wang*, HuanHuan Zou, Ming Fang
The First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, Zhejiang, China.
Corresponding Author: Kai Wang, Email: wangkai2@wmu.edu.cn

Abstract: In order to enrich the population diversity of the original marine predator algorithm (MPA), balance the phased exploration and development effect of the algorithm, and enhance the ability of leaping out of the local optimal solution, so as to improve the effect of solving engineering optimization problems, a multi-strategy developed marine predator algorithm (MSDMPA) is proposed. The algorithm first initializes the population through sine chaotic mapping and uses a phased position update mechanism with nonlinear convergence factor to guide individual position updates. At the same time, the adaptive monotonically decreasing update strategy for step size control is introduced. Finally, the algorithm is improved by combining the strategy of the Levy flight. The simulation experiment is based on 13 benchmark test functions to verify the effectiveness of various improvement strategies. The convergence analysis and Wilcoxon rank sum test are performed on the optimization results of the improved algorithm and the comparison algorithms, proving that MSDMPA has good optimization performance and robustness. Finally, MSDMPA is applied to the pressure vessel of a large hospital's liquid oxygen tank, further verifying its effectiveness and reliability in solving practical problem.
Keywords: Marine predator algorithm; MPA; MSDMPA; Convergence factor; Monotonically decreasing; Pressure vessel

1 INTRODUCTION

Advances in engineering have engendered optimization conundrums whose solutions via conventional algorithms like dynamic programming and branch and bound become computationally intractable with scale escalation. Compared with the traditional precision algorithm, the heuristic algorithm has a series of advantages: the model is flexible, the solution precision is high, and it can be applied to some complex cases. Meta-heuristic algorithm is a kind of heuristic algorithm, which is suitable for some complicated optimization fields, such as the constrained extreme value problem often encountered in engineering applications. In recent decades, researchers in relevant institutes and universities have innovated various meta-heuristic algorithms based on bionic evolution and natural laws[1-4]. However, these intelligent optimization algorithms have some disadvantages in different applications: low optimization precision and convergence efficiency [5]. Therefore, according to corresponding practical applications, various improved algorithms have emerged [6, 7].

In 2020, Faramarzi et al. introduced the Marine Predator Algorithm, an innovative addition to the realm of swarm intelligence methodologies [8]. The design idea of the algorithm comes from the marine higher organism's preys on the lower organisms in the sea, following the biological principle of the jungle. The algorithm simulates the differential movement patterns between species under different speed ratios to improve optimization efficiency through parallel architecture. Compared with other algorithms, it has many advantages, such as good local exploration effect, equal emphasis on development and exploration, and is not easy to fall into traps. Therefore, the algorithm has been applied by relevant researchers in some aspects, such as the multilevel image separation problem [9], heat-power system [10], parameter extraction of diodes photovoltaic models [11], metal correlation prediction [12]. Although the MPA algorithm has excellent optimization performance, it can still be further improved to meet more precise requirements of engineering application [13]. Fan et al. enhanced the Marine Predator Algorithm through opposition-based learning and self-adaptive updates [14]. Yang et al. refined MPA with adaptive mechanisms and chaotic strategies for robust optimization [15]. Sadiq et al. proposed a set of nonlinear functions to change the search mode of MPA which is applied to the problem of fair distribution of network efficiency [16]. Elaziz et al. integrated fuzzy entropy into MPA, advancing COVID-19 CT segmentation efficacy [17]. Shaheen et al. combined PSO and MPA algorithms using the minimum adjustable parameters to improve algorithm [18].

According to the analysis and summary of the above related literature, although the MPA algorithm has excellent solving performance, it still exhibits shortcomings, notably in initial population diversity, exploration-exploitation balance, and escaping local optima. Therefore, this paper introduces the developed Marine Predator Algorithm, enhanced with multiple strategies. At the initial stage, sine chaotic mapping of population distribution was adopted to make population distribution more uniform. Then nonlinear convergence factor for the phased position update mechanism is used to update prey positions in the high-speed ratio stage, the equal-speed ratio stage and the low-speed ratio stage respectively. The adaptive montonic decrement update strategy is used to update the step size control P. Finally, the Levy flight operator strategy is combined to improve the probability of the population jumping out of the local optimal solution.

This manuscript delineates the developed Marine Predators Algorithm presenting its foundational concept in Section 2. Subsequent to this, Section 3 details the novel multi-strategy developed MPA, articulating four innovative augmentation strategies. Section 4 documents comparative assessments of MSDMPA against existing algorithms, while Section 5 applies it to a pressure vessel design challenge, juxtaposing its performance with alternative approaches. The closing section encapsulates our findings and contemplates prospective research avenues.

2 ORIGINAL MARINE PREDATOR ALGORITHM

2.1 Population Initialization

Like other meta-heuristic algorithms, MPA first generates prey populations by randomly locating N individuals in a M -dimensional problem space, the formula is as follows:

$$W_{i,j} = LB_j + k \times (HB_j - LB_j) \tag{1}$$

Where $i \in [1, N]$, $j \in [1, M]$, $W_{i,j}$ represents the current position of the i -th prey in the j - dimension, $k \in [0,1]$, HB_j and LB_j are the highest and lowest limits of the j dimension search space respectively.

First of all, the initial prey is generated during initialization, that is, the prey matrix, according to which the predator will update its position. The formula is as follows:

$$W = \begin{bmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,M} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N,1} & W_{N,2} & \cdots & W_{N,M} \end{bmatrix}_{N \times M} \tag{2}$$

Where each row of elements in the matrix is a position vector for the prey.

The one that finds the best extreme value among all the groups is called the elite, and N copies of the optimal individual are used to construct the elite matrix. The formula is as follows:

$$O = \begin{bmatrix} W_{1,1}^1 & W_{1,2}^1 & \cdots & W_{1,M}^1 \\ W_{2,1}^1 & W_{2,2}^1 & \cdots & W_{2,M}^1 \\ \vdots & \vdots & \ddots & \vdots \\ W_{N,1}^1 & W_{N,2}^1 & \cdots & W_{N,M}^1 \end{bmatrix}_{N \times M} \tag{3}$$

Where each row of elements in the matrix is a position vector for the top predator. In fact, both predators and prey act as searchers, because any individual in the population is at risk of being preyed upon. If the fitness value of the new individuals is smaller than those of the previous elite after each update, then this individual becomes the next elite.

2.2 Staged Update Phase

In Marine Predator Algorithm, prey-predator velocity ratios delineate three phases: the high-velocity ratio stage(H-VRS), the equal-velocity ratio stage(E-VRS) and the low-velocity ratio stage(L-VRS). Under guidance of the predator matrix, the solution space is searched by using Levy walk and Brownian motion.

1) H-VRS: When $0 \le t \le \frac{T}{3}$, population is completely in the process of space exploration. In this stage, velocity value of prey is greater than predator, and its mathematical model is shown in equation (4):

$$\begin{cases} s_i = B \otimes \left(O_i - B \otimes W_i\right) \\ W_{i+1} = W_i + PR \otimes s_i \end{cases} \tag{4}$$

Where s_i is the walking step size; Vector B following the principle of Brownian walking; $\otimes$ is an item-by-item multiplication method; P is 0.5, also called step size control in this paper; R ranges from 0 to 1.

2) E-VRS: When $\frac{T}{3} < t \le \frac{2T}{3}$, the algorithm is in the transition stage between exploration and development. In this stage, predator's movement velocity is the same as that of the prey, and half of the population is developed through Levy flight, while the other half of the population is explored through Brownian motion. The mathematical model is:

$$\begin{cases} s_i = L \otimes \left(O_i - L \otimes W_i\right) \\ W_{i+1} = W_i + PR \otimes s_i \end{cases} \tag{5}$$

$$\begin{cases} s_i = B \otimes \left(B \otimes O_i - W_i \right) \\ W_{i+1} = O_i + PC_F \otimes s_i \end{cases} \tag{6}$$

Where Vector L following the principle of Levy flight distribution; C_F is adaptive parameters for controlling the movement of predators.

3) L-VRS: When $\frac{2T}{3} < t \le T$, the predator uses the Levy strategy for development, and its mathematical model is as follows:

$$\begin{cases} s_i = L \otimes \left(L \otimes O_i - W_i \right) \\ W_{i+1} = O_i + PC_F \otimes s_i \end{cases} \tag{7}$$

2.3 FADS Effect

Random factors around the ocean will change the behaviour of the marine organism called FADS effect, which can be likened to fleeing local extrema. Random factors can be regarded as favorable for the algorithm to find the optimal solution. Therefore, MPA takes into account the FADS effect, the formula is as follows:

$$W_{i+1} = \begin{cases} W_i + C_{\mathrm{F}} \left[LB_{rep} + R \otimes \left(HB_{rep} - LB_{rep} \right) \right] \otimes G, a \le F_{\mathrm{FADs}} \\ W_i + \left[F_{\mathrm{FADs}}(1-a) + a \right] \left(W_{r1} - W_{r2} \right), a > F_{\mathrm{FADs}} \end{cases} \tag{8}$$

Where $F_{\mathrm{FADs}} = 0.2$ represents the probability of reducing local extreme effects on marine predators ; G is a binary vector ; $a \in [0,1]$; W_{r1} and W_{r2} are two random preys from population. HB_{rep} and LB_{rep} represents the matrix of HB and LB respectively.

3 THE MULTI-STRATEGY DEVELOPED MARINE PREDATOR ALGORITHM

The original MPA algorithm is characterized by uneven individual distribution in the early stage of exploration, quickly converged in the later stage, but prone to fall into the dilemma of local extreme value. In order to appropriately increase the population richness in the early stage, coordinate exploration with development in each stage, improve convergence efficiency, and avoid premature maturation, improvements are made to it. This article mainly proposes four improvement strategies from three aspects: increasing population diversity, balancing global exploration and local development, and enhancing the ability to escape the local optimal dilemma.

3.1 Using Sine Chaotic Mapping for Initial Population

The original MPA used random initialization to determine the position distribution of the initial population, which usually generates different initial populations each time, making it more convenient to use. But there are also drawbacks, such as the uneven distribution of initial individuals in the solution space, often encountering individuals in local areas that are too dense, while some initial individuals in certain areas are too sparse. This situation is very unfavorable for the early exploration of MPA, and chaos initialization can effectively improve the quality of the original population and expand search rang .

The use of Sine function for chaotic mapping is one of the chaotic mapping methods . Improving the population quality of MPA algorithm has effectively expanded the scope of comprehensive exploration of the algorithm. Chaos initialization has the characteristics of randomness, traversal, and regularity. It traverses the search space within a certain range according to its own laws without repetition. This generates an initial population that has significant improvements in solving accuracy and convergence speed.

The sine chaos formula is below:

$$Z_{i+1} = R \times \sin(\pi \bullet Z_i), R \in (0,1) \tag{9}$$

The sine chaotic image is shown in Figure 1:

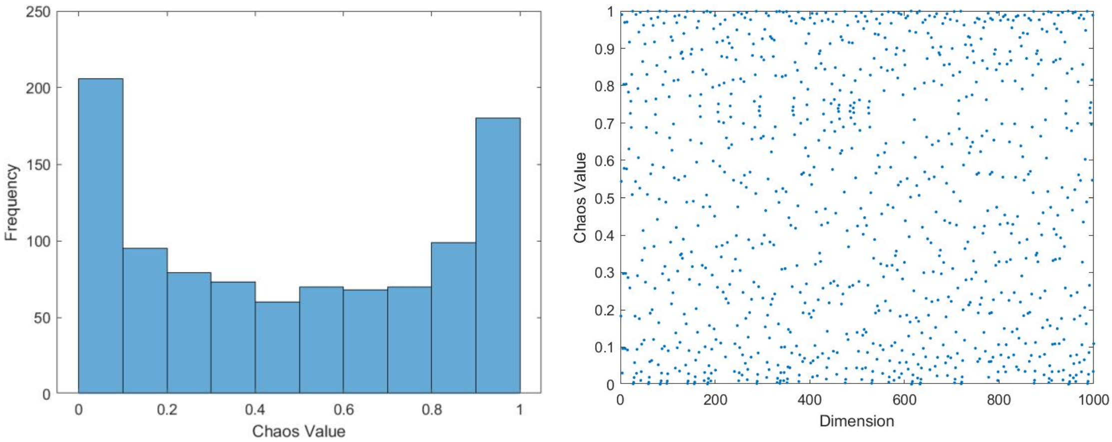

Figure 1 Sine Chaotic Image

The mathematical model for initializing the population using sine chaotic mapping is:

$$W = W_{\min} + \text{sine} \times (W_{\max} - W_{\min}) \tag{10}$$

3.2 Nonlinear Convergence Factors μ for Staged Position Updating Mechanism

The original MPA algorithm suffers from an imbalance between global exploration and local development when solving complex optimization problems, such as insufficient exploration space in the H-VRS and E-VRS, and premature convergence before reaching the optimal value in the E-VRS and L-VRS, leading to the problem of premature convergence.

By simplifying formula (4), update formula of the H-VRS and the levy flight in the E-VRS can be obtained as $W_{i+1} = W_i + \varepsilon$ .Where ε is defined as the integrated step length. Similarly, the formula of population renewal for the Brownian motion in the E-VRS and L-VRS is $W_{i+1} = O_i + \varepsilon$ .Therefore, regardless of the position update of the population at any stage, its population position update mechanism takes the position of prey and predators as the benchmark for the next position. Therefore, this paper proposes a staged position updating mechanism with nonlinear convergence factor μ to control the dependence of population on prey and elite positions during position updating. The mathematical formula is as follows:

$$W_{i+1} = \mu W_i + \varepsilon \tag{11}$$

$$W_{i+1} = \mu O_i + \varepsilon \tag{12}$$

According to the characteristics of each stage, the convergence factor μ should be designed into three parts, and the convergence factor μ of each stage should conform to the following characteristics to realize the implementation of positional differential updating in different stages, so as to achieve the achieve global exploration and local development parallel:

1) In the H-VRS, the convergence factor μ should maintain a dynamic adaptive process to reduce the dependence on the original prey position, so that the whole prey group can expand the global scope, optimize the individual quality traversal, and improve the overall exploration capability.

2) In the E-VRS, the population is in a situation of exploration and exploitation. At this time, the population will experience a transition period from exploration to exploitation, and the exploration scope should be gradually reduced. Therefore, the convergence factor μ should be gradually close to the elite, so that the exploration and exploitation ability of the rhythm optimization algorithm can gradually achieve a good transition in the E-VRS.

3) In the L-VRS, the population has fully entered the exploitation process, so the convergence factor μ should strengthen the dependence on the elite, so as to improve deep optimization capability.

Based on the above analysis, a mathematical model of nonlinear convergence factors μ for staged position updating mechanism is introduced. The formula is below:

$$\mu=\begin{cases}0.5-0.5\times\sin[(t-\frac{T}{3})\times\frac{3\pi}{T}+\frac{\pi}{2}] & 0\le t\le\frac{T}{3}\\ 1+\cos[(t-\frac{T}{3})\times\frac{3\pi}{2T}+\pi] & \frac{T}{3}< t\le\frac{2T}{3}\\ 1 & \frac{2T}{3}<t\le T\end{cases} \tag{13}$$

In order to clarify nonlinear convergence factor μ for staged position updating mechanism more clearly, its function image is drawn, where the value of T is 500, as shown in Figure 2:

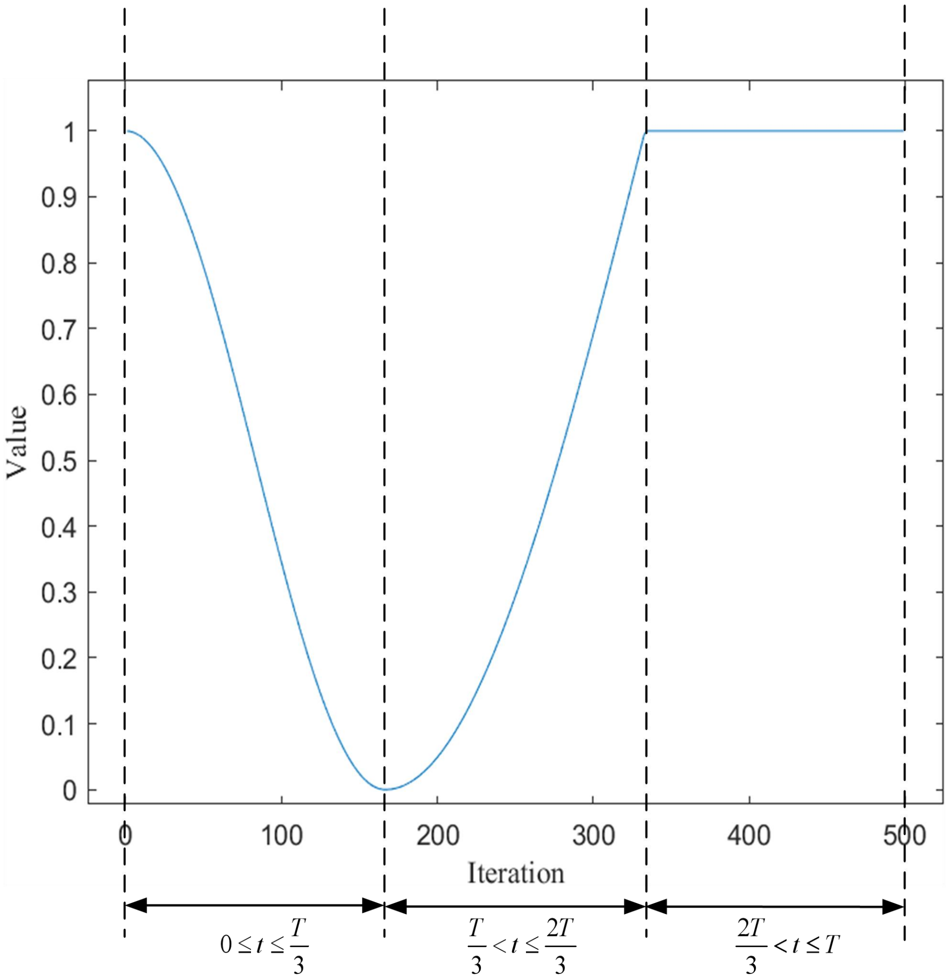

Figure 2 The curve of nonlinear convergence factor μ

According to Figure 2, when $0\le t\le\frac{T}{3}$, the convergence factor μ is monotonically decreasing, ensuring that the whole group can get rid of the prey position and expand the search range. When $\frac{T}{3}<t\le\frac{2T}{3}$, the convergence factor μ is monotonically increasing, ensuring that the whole population gradually approaches the elite. When $\frac{2T}{3}<t\le T$, the convergence factor μ is adjusted to 1, ensuring that the population is completely close to the elite and achieving local depth search. The above results quite meet the above requirements. The nonlinear convergence factors μ for staged position updating mechanism achieves a proper coordination between exploration and development in problem space by combining breadth and depth.

3.3 The Adaptive Monotonically Decreasing Update Strategy for Step Size Control P

As mentioned above, all population individual position updates are based on prey and predator movements. The step size control P is combined with the random number R or C_F to control the scale of the walking step size s_i. In the original MPA algorithm, P is 0.5, and its fixed value is obviously difficult to adapt to the step size requirements required for dynamic stage position update. Therefore, according to the optimization characteristics of the three stages, this paper proposes to update the P in the original algorithm by using the adaptive monotonically decreasing update strategy to further navigate the equilibrium between worldwide exploration and regional development. The specific analysis is as follows:

1) In the H-VRS, the scope of the search field should be expanded, otherwise the prey will always move around the self -boundary, unable to explore the whole area, and then it is easy to fall into the local optimal solution due to the failure to find the global optimal solution in the following the E-VRS and L-VRS. Therefore, the P in this stage should be maintained at a slightly higher level, and the downward trend should be maintained to lay the groundwork for the subsequent E-VRS.

2) In the E-VRS, because it has been in the stage of parallel exploration and development, the step size control P should be kept in a compromise state to avoid missing the global optimal solution, and at the same time avoid falling into the dilemma of local optimal. Therefore, the P at this stage should maintain a stable middle value.

3) In the L-VRS, the entire population is near the elite, and each individual is in the development stage. If the step length is too long at this time, it is easy to overstack with the FADS effect, resulting in easy jumping out of the global optimal solution and oscillating around the optimal solution. Therefore, the P in this stage should maintain a downward trend and eventually tend to a smaller value.

In summary, this paper proposes an adaptive monotone decreasing update strategy for P, and the formula is shown as follows:

$$P = \frac{1}{2} \times [1 + \tanh^3(2 - \frac{4t}{T})] \tag{14}$$

In order to clarify the adaptive monotonically decreasing update strategy for P more clearly, draw its function image, in which the value of T is 500, as shown in Figure 3:

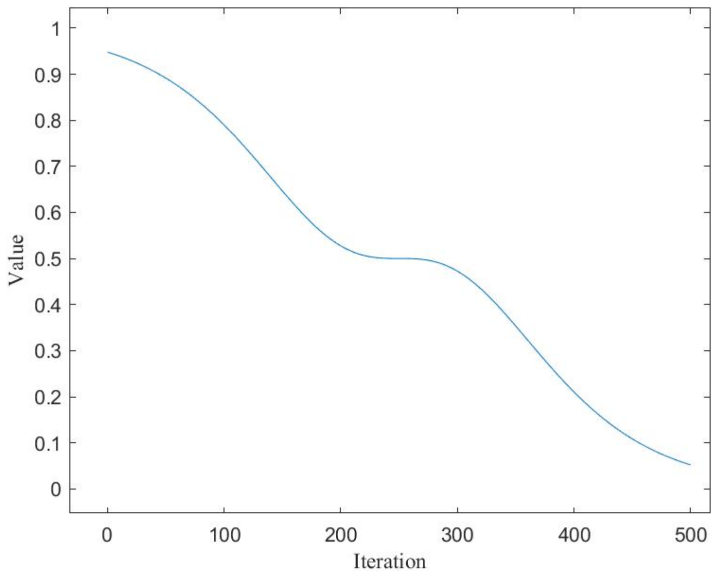

Figure 3 Adaptive Monotonically Decreasing P -Value Curve

This is obvious from the Figure 3 that in the H-VRS, the P is a large value, while maintaining a monotonically decreasing trend. In the E-VRS, although the P is still monotonically decreasing, the decline is small and maintains a relatively stable middle value. In the L-VRS, the P continues to decrease monotonically and eventually reaches a stable smaller value. Therefore, formula 14 fully meets the requirements of analysis.

3.4 The Levy Flight Operator Strategy

In the FADS effect mathematical model formula (8), when $r \le F_{\text{FADs}}$, the FADS effect facilitates the MPA individual's escape from local optimal. But its ability to break through local optima is limited, and the probability is entirely determined by the size of the set value. Therefore, in order to increase the probability of the population escaping from local optimal and achieve local quadratic breakthrough, this paper introduces the Levy flight operator strategy. When

$r > F_{\mathrm{FADs}}$, replacing the random number a in formula (8) with the Levy flight operator, which will randomly generate a step size and expand the search range . The stride formula for simulating the Levy flight process is as follows:

$$Levy(\beta) = \frac{\mu}{|v|^{-\beta}} \tag{15}$$

Where $Levy(\beta)$ is the Levy distribution with parameter β . $\mu \sim N(0, \sigma_\mu^2)$, $v \sim N(0, \sigma_v^2)$.
Therefore, the entire FADS formula is updated as follows:

$$W_{i+1} = \begin{cases} W_i + C_{\mathrm{F}}\left[LB_{rep} + R \otimes \left(HB_{rep} - LB_{rep}\right)\right] \otimes G, a \le F_{\mathrm{FADs}} \\ W_i + \left[F_{\mathrm{FADs}}(1 - m_{Levy}) + m_{Levy}\right]\left(W_{r1} - W_{r2}\right), a > F_{\mathrm{FADs}} \end{cases} \tag{16}$$

Where m_{Levy} is the random number of Levy flight operator.

3.5 Algorithm Step

MSDMPA solution steps are shown in Figure 4 below:

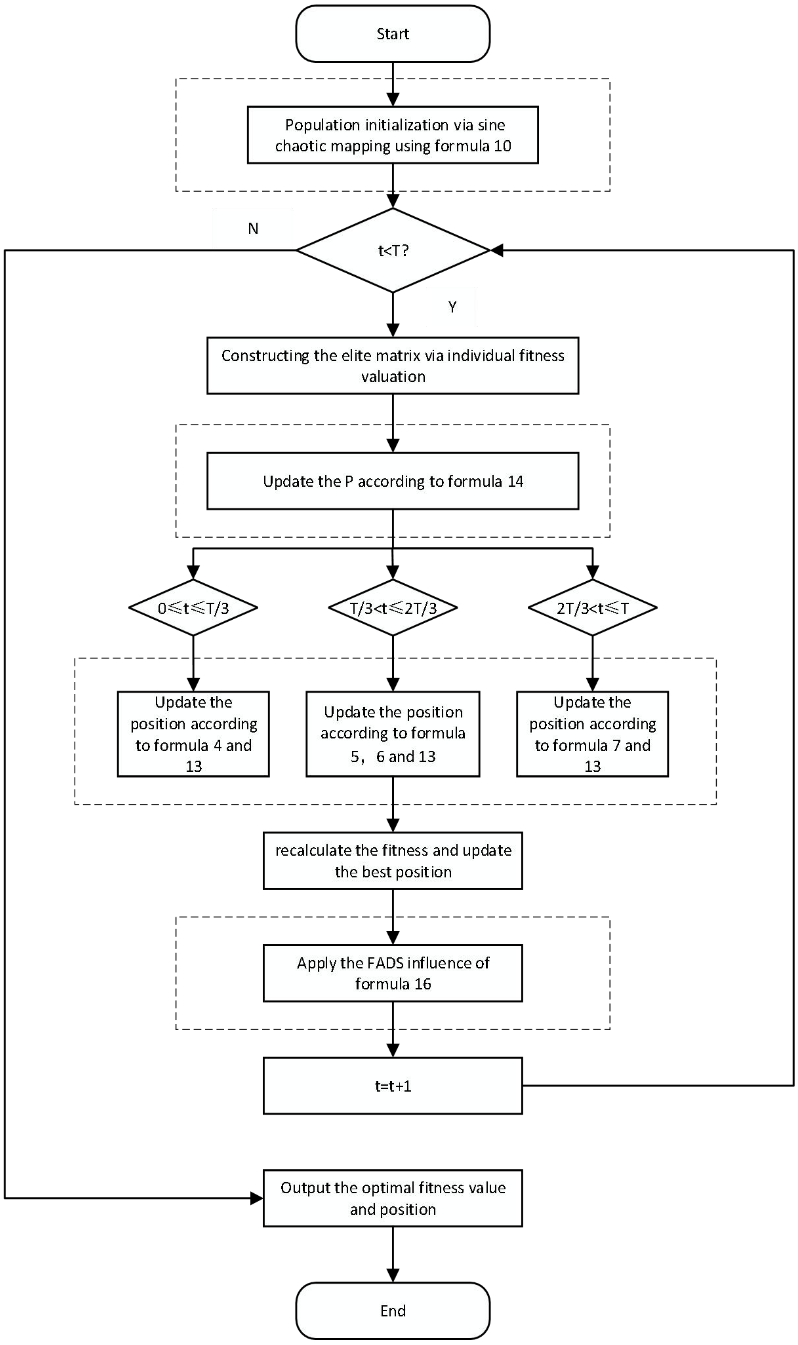

Figure 4 Flowchart of MSDMPA

Step 1: Population initialization via sine chaotic mapping using formula (10).
Step 2: Constructing the elite matrix via individual fitness valuation.
Step 3: Update the P according to formula (14), and use formulas (4-7) and (13) to update the position.
Step 4: After updating the position, recalculate the fitness and update the best position.
Step 5: Apply the FADS effect of formula (16).
Step 6: Evaluate the algorithm's termination criterion; upon fulfillment, conclude execution; otherwise, return to the Step 2.

4 EXPERIMENTAL DESIGN, SIMULATION AND ANALYSIS

4.1 Preliminary Preparation

In order to prove the optimization accuracy and convergence of the improved algorithm, 13 commonly used standard test functions are selected for testing. Category, name, expression, range and dimension are listed in Table 1.

Table 1 Test Functions

Category	Name	Expression	Range	Dimension
High dimensional unimodal function	F1	$f(m)=\sum_{i=1}^{n} m_i^2$	[-100,100]	30
	F2	$f(m)=\sum_{i=1}^{n}\lvert m_i\rvert+\prod_{i=1}^{n}\lvert m_i\rvert$	[-10,10]	30
	F3	$f(m)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} m_j\right)^2$	[-100,100]	30
	F4	$f(m)=\max_i\{\lvert m_i\rvert,1\le i\le n\}$	[-100,100]	30
	F5	$f(m)=\sum_{i=1}^{n} i m_i^4+\text{random}[0,1)$	[-1.28,1.28]	30
High dimensional multimodal function	F6	$f(m)=\sum_{i=1}^{n}\left[m_i^2-10\cos\left(2\pi m_i\right)+10\right]$	[-5.12,5.12]	30
	F7	$f(m)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} m_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi m_i\right)\right)+20+e$	[-32,32]	30
	F8	$f(m)=\frac{1}{4000}\sum_{i=1}^{n} m_i^2-\prod_{i=1}^{n}\cos\left(\frac{m_i}{\sqrt{i}}\right)+1$	[-600,600]	30
Fixed low dimensional multimodal function	F9	$f(m)=4m_1^2-2.1m_1^4+\frac{1}{3}m_1^6+m_1m_2-4m_2^2+4m_2^4$	[-5,5]	2
	F10	$f(m)=\left(m_2-\frac{5.1}{4\pi^2}m_1^2+\frac{5}{\pi}m_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos m_1+10$	[-5,10]x [0,15]	2
	F11	$f(m)=-\sum_{i=1}^{4} c_i\exp\left(-\sum_{j=1}^{6} a_{ij}\left(m_j-p_{ij}\right)^2\right)$	[0,1]	6

F12	$f(m) = -\sum_{i=1}^{5}\left[\left(m-a_i\right)\left(m-a_i\right)^T + c_i\right]^{-1}$	[0,10]	4
F13	$f(m) = -\sum_{i=1}^{10}\left[\left(m-a_i\right)\left(m-a_i\right)^T + c_i\right]^{-1}$	[0,10]	4

F1-F13 consists of 13 functions representing three types of functions. Through three different types of test functions, the optimization performance of MSDMPA algorithm is fully verified.

4.2 The Validity Of MSDMPA

In order to measure the level of MSDMPA algorithm, MSDMPA is compared with other swarm intelligent algorithms, including classical optimization algorithms WOA, GWO, SABO and original MPA. An ensemble of 30 individuals with termination at T=500 iterations undergo evaluation across various functions, with aggregate performance assessed by optimal, mean, and standard deviation metrics from trials.
In order to ensure the unity of the experiment, the experimental simulation of all algorithms uses the same hardware and software platform, and all experiments are implemented in MATLAB 2021a. Table 2 presents the outcomes of our empirical analysis:

Table 2 Optimization Results of Different Intelligent Algorithms

Function	Index	MSDMPA	MPA	WOA	GWO	SABO
F1	min	0.000E+00	4.951E-25	4.191E-84	3.182E-29	2.936E-200
	std	0.000E+00	1.289E-22	7.932E-71	1.163E-27	0.000E+00
	avg	0.000E+00	8.595E-23	1.955E-71	8.658E-28	1.802E-197
F2	min	2.523E-183	1.031E-14	1.047E-58	2.166E-17	4.992E-113
	std	0.000E+00	1.827E-13	1.555E-51	9.218E-17	8.134E-111
	avg	4.235E-179	1.941E-13	4.034E-52	1.029E-16	3.370E-111
F3	min	0.000E+00	2.827E-09	2.367E+04	1.429E-09	1.465E-82
	std	0.000E+00	4.646E-04	1.292E+04	2.264E-04	7.740E-41
	avg	2.769E-318	2.459E-04	4.924E+04	7.495E-05	1.413E-41
F4	min	1.387E-172	9.402E-10	1.550E-02	8.007E-08	9.218E-79
	std	0.000E+00	1.490E-09	2.713E+01	5.628E-07	7.195E-77
	avg	1.179E-167	3.328E-09	4.342E+01	6.238E-07	4.317E-77
F5	min	4.667E-06	2.769E-04	1.654E-04	2.351E-04	8.319E-06
	std	8.230E-05	6.863E-04	2.295E-03	1.016E-03	9.029E-05
	avg	9.340E-05	1.220E-03	2.671E-03	1.947E-03	1.205E-04
F6	min	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	std	0.000E+00	0.000E+00	1.442E-14	2.983E+00	0.000E+00
	avg	0.000E+00	0.000E+00	3.790E-15	2.300E+00	0.000E+00
F7	min	8.882E-16	3.633E-13	8.882E-16	7.550E-14	4.441E-15
	std	0.000E+00	1.278E-12	2.234E-15	1.454E-14	0.000E+00
	avg	8.882E-16	1.912E-12	4.915E-15	9.989E-14	4.441E-15
F8	min	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	std	0.000E+00	0.000E+00	3.839E-02	7.036E-03	0.000E+00
	avg	0.000E+00	0.000E+00	7.008E-03	3.338E-03	0.000E+00
F9	min	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00
	std	4.164E-16	4.554E-16	2.063E-09	1.530E-08	8.971E-03
	avg	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.026E+00
F10	min	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	std	1.788E-15	6.981E-15	6.185E-06	1.101E-06	6.542E-02
	avg	3.979E-01	3.979E-01	3.979E-01	3.979E-01	4.337E-01
	min	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.321E+00

F11	std	4.702E-13	5.451E-12	9.504E-02	8.620E-02	1.493E-01
	avg	-3.322E+00	-3.322E+00	-3.250E+00	-3.258E+00	-3.238E+00
F12	min	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-5.735E+00
	std	4.064E-12	8.147E-11	2.871E+00	1.922E+00	5.015E-01
	avg	-1.015E+01	-1.015E+01	-7.992E+00	-9.306E+00	-4.908E+00
F13	min	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-6.698E+00
	std	1.776E-12	6.233E-11	2.808E+00	1.093E-03	8.785E-01
	avg	-1.054E+01	-1.054E+01	-8.425E+00	-1.053E+01	-4.693E+00

By analysing the results in Table 2, it can be clearly seen that the performance of algorithm MSDMPA has great advantages over other algorithms: it solves 13 functions, a total of 39 indicators are optimal. For F1-F4, MSDMPA has obvious advantages in all indexes, which shows that MSDMPA has a strong ability to optimize the high dimensional unimodal test function. For F6 and F8, most of the algorithms show better solution result, which shows that the optimization of F6 and F8 test functions is not complicated. For F9-F12, MPA and MSDMPA have excellent optimization results, which shows the superiority of MPA series algorithms. However, MSDMPA still exceeds the MPA algorithm by a slight advantage, indicating that MSDMPA is better in the depth development of the global optimal solution.

4.3 Convergence Performance Analysis

The convergence curve can directly reflect the convergence, robustness and resistance to local stagnation of each algorithm. The convergence curves of each function obtained are shown in Figure 5.

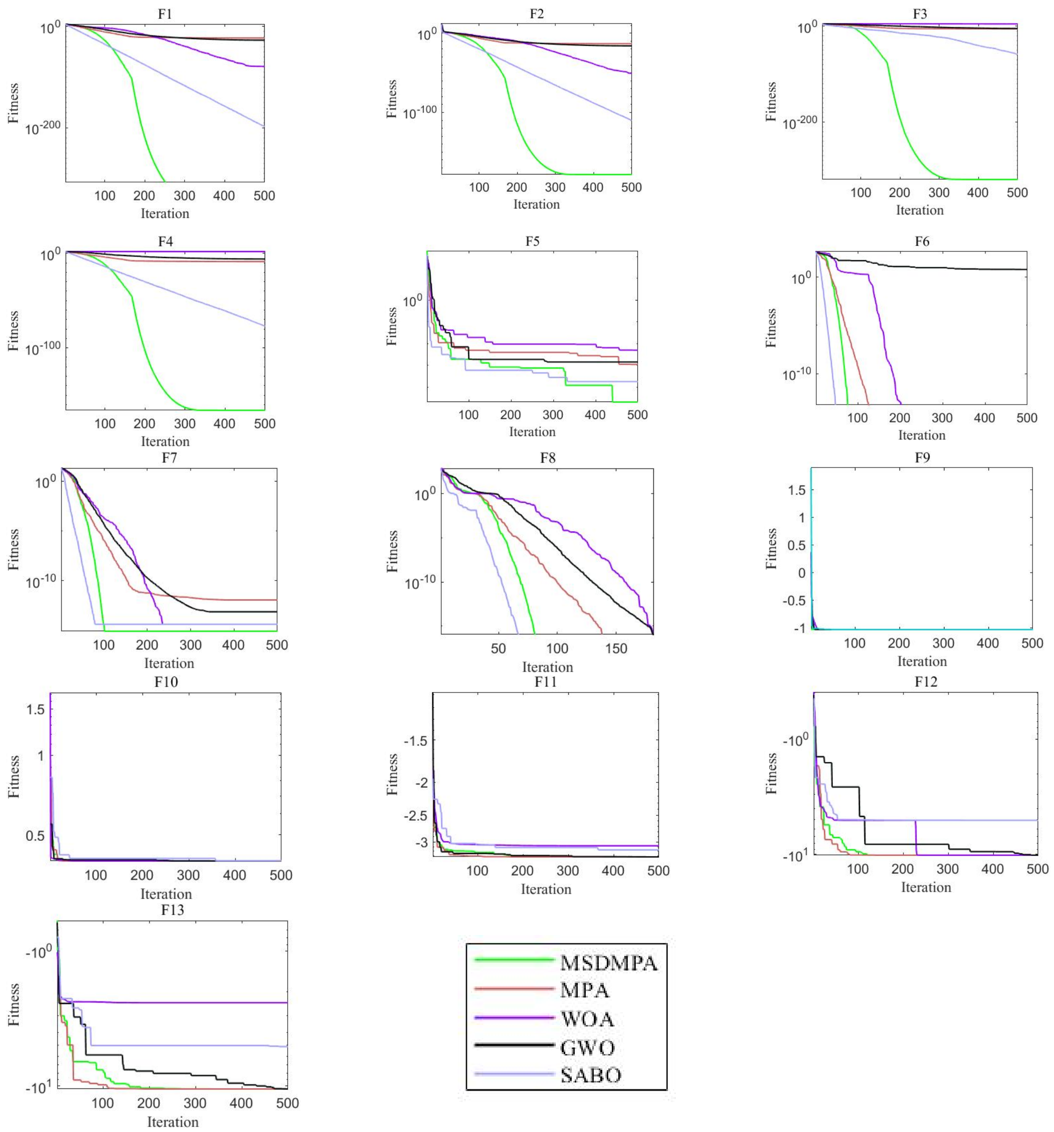

Figure 5 Convergence Diagram

According to the analysis in Figure 5, it is obvious that MSDMPA has more efficient optimization ability than other algorithms, especially in the test functions of F1-F5 and F7. In the first third stage, the MSDMPA search curve of most test functions will be slightly inferior to other algorithms, because when the convergence factor μ is in the first third stage, it mainly helps the population increase search distance. The focus is on expanding the range rather than solving accuracy. In the latter two thirds of the stage, the MSDMPA algorithm rapidly converges, because the convergence factor μ helps the population get rid of predators and approach the elite for domain search, and the convergence is instantly enhanced. Therefore, MSDMPA can significantly optimize the accuracy of the test functions.

4.4 Algorithm Statistical Test Results

Employing the Wilcoxon rank sum test enhances the intuitive evaluation of algorithmic disparities. This nonparametric method ascertained the statistical significance between the proposed MSDMPA and comparator algorithms. Calculation outcomes yielded the h value; the h value less than 0.05 denoted a significant departure of MSDMPA from its contenders, while the h value surpassing 0.05 suggested negligible variation. Table 3 encapsulates the rank sum test outcomes for functions F1-F13:

Table 3 Wilcoxon Rank Sum Test Results of Each Algorithm

Function	MPA	WOA	GWO	SABO
F1	1.21178E-12	1.21178E-12	1.21178E-12	1.21178E-12
F2	3.01986E-11	3.01986E-11	3.01986E-11	3.01986E-11
F3	1.78949E-11	1.78949E-11	1.78949E-11	1.78949E-11
F4	3.01986E-11	3.01986E-11	3.01986E-11	3.01986E-11
F5	3.01986E-11	7.38908E-11	3.68973E-11	0.176127545
F6	NaN	0.160741998	4.48798E-12	NaN
F7	1.21079E-12	8.96603E-11	1.16418E-12	1.6853E-14
F8	NaN	0.333710696	0.011035086	NaN
F9	0.0300607	6.31878E-12	6.31878E-12	6.31878E-12
F10	0.097602373	6.4697E-12	6.4697E-12	6.4697E-12
F11	3.4742E-10	3.01986E-11	3.01986E-11	3.01986E-11
F12	3.96477E-08	3.01986E-11	3.01986E-11	3.01986E-11
F13	5.0922E-08	3.01986E-11	3.01986E-11	3.01986E-11

According to the results in Table 3, only 3 test functions in MPA have h values greater than 0.05. Only 2 test functions in WOA have h values above 0.05. All h values in GWO are less than 0.05, and the MSDMPA algorithm completely shows significant differences. Finally, only 3 test functions in SABO have h values greater than 0.05. Therefore, it can be considered that the performance of MSDMPA algorithm is basically stronger, and the reliability and effectiveness of MSDMPA algorithm are statistically confirmed.

5 APPLICATION ISSUE OF PRESSURE VESSEL

A pressure vessel is a sealed equipment that carries a certain amount of pressure. The liquid oxygen station tank is a type of pressure vessel, with a common pressure range of 0.8-1.6Mpa, to meet the oxygen demand of the entire hospital. This article applies the MSDMPA algorithm to solve the tank problem of liquid oxygen stations in hospitals, which is essentially a pressure vessel design problem. As shown in Figure 6, simplify the liquid oxygen station tank of a large tertiary hospital. To minimize production costs in pressure vessel design, optimization of wall thickness d_1 , head thickness d_2, internal radius m_3, and cylinder length m_4 is imperative. At the same time, the results are compared with 11 algorithms including NRBO[19], SCA[20], FTTA[21], WOA[22], etc.

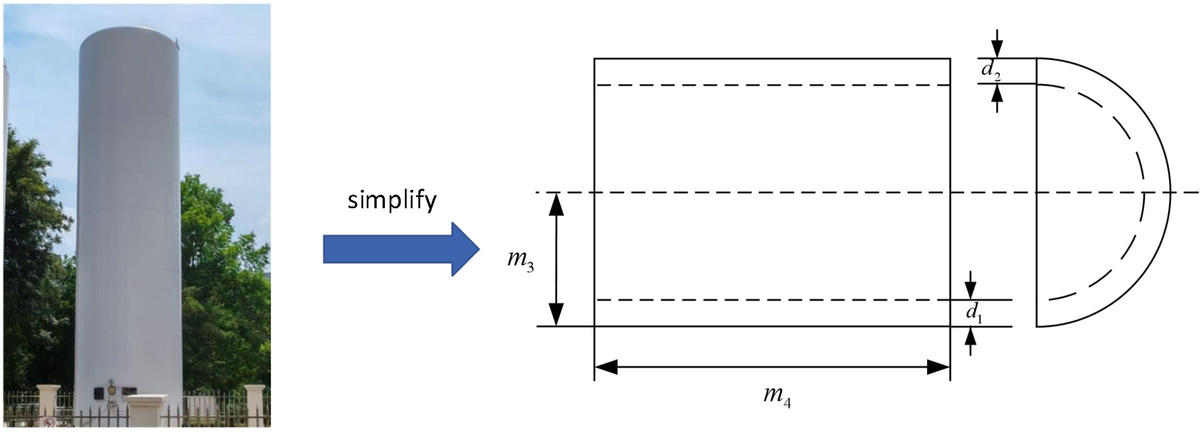

Figure 6 Pressure Vessel Design Model

The formula model is as follows [23]:
Minimize:

$$C_{\min} = 3.1661 d_1^2 m_4 + 1.7781 d_2 m_3^2 + 0.6224 d_1 m_3 m_4 + 19.84 d_1^2 m_3 \tag{17}$$

Subject to:

$$\begin{aligned} Q_1 &= 0.00954 m_3 - d_2 \le 0, \\ Q_2 &= 0.0193 m_3 - d_1 \le 0, \\ Q_3 &= m_4 - 240 \le 0, \\ Q_4 &= -\pi m_3^2 m_4 - \frac{4}{3}\pi m_3^3 + 1296000 \le 0. \end{aligned} \tag{18}$$

Where:

$$\begin{aligned} d_1 - 0.0625 x_1 &= 0, \\ d_2 - 0.0625 x_2 &= 0. \end{aligned} \tag{19}$$

With bounds:

$$\begin{aligned} &10 \le m_4, m_3 \le 200, \\ &1 \le x_1, x_2 \le 99 (\text{integer variables}) \end{aligned} \tag{20}$$

In the pursuit of experimental impartiality, T is fixed at 500 with a population size of 30, executing each algorithm 30 distinct times. Table 4 exhibits the optimization outcomes for MSDMPA and 11 comparator algorithms:

Table 4 Results of Pressure Vessel Design Problem

Algorithm	Worst	Std	Best	Mean
FTTA	7.31900E+03	4.24456E+02	5.88533E+03	6.28795E+03
BWO	1.28980E+04	1.32006E+03	7.13825E+03	9.32401E+03
CPO	5.96927E+03	2.10019E+01	5.89236E+03	5.91138E+03
FHO	9.72725E+03	7.10387E+02	7.08203E+03	8.58511E+03
GWO	7.22036E+03	4.88568E+02	5.89718E+03	6.27777E+03
HHO	7.87263E+03	3.63382E+02	6.25033E+03	6.90573E+03
NRBO	7.70891E+03	5.90666E+02	5.89013E+03	6.55175E+03
SCA	9.74744E+03	8.36443E+02	6.37127E+03	7.70263E+03
SGA	7.33360E+03	4.67630E+02	5.88823E+03	6.50263E+03
WOA	4.45680E+04	6.75507E+03	6.11888E+03	1.02608E+04
MPA	5.88548E+03	3.88065E-02	5.885334E+03	5.88538E+03
MSDMPA	5.88547E+03	2.94267E-02	5.885333E+03	5.88535E+03

According to the experimental results, the worst value of MSDMPA is 5.88547E+03, the standard deviation is 2.94267E-02, the best value is 5.885333E+03, and the mean value is 5.88535E+03, which are both the smallest among the algorithms. Therefore, it is obvious that MSDMPA algorithm has excellent optimization performance and good robustness in solving pressure vessel design problem.

6 SUMMARY AND PROSPECT

The paper introduces the multi-strategy developed marine predator algorithm, a novel approach to enhance the performance of heuristic optimization algorithms. By integrating sine chaotic mapping to initialize individuals, MSDMPA enhances population diversity. A nonlinear convergence factor μ refines the update mechanism, ensuring effective exploration and exploitation phases. The step size control P's adaptive decrement strategy along with Levy flight mechanics collectively mitigate premature convergence and foster global search local development capabilities. Empirical analyses demonstrate that MSDMPA exhibits notable improvements in convergence rates and precision of solutions. Application to a canonical pressure vessel design challenge corroborates MSDMPA's practical efficacy in complex engineering optimizations.

Although the MSDMPA proposed in this paper shows good performance in some test functions and engineering problems compared with the MPA algorithm, it is only compared with a few algorithms, and only tests the engineering optimization problems of pressure vessel design. Therefore, it can be compared with more excellent algorithms in the future, and more problems related to engineering fields can be tested to reflect the superiority of MSDMPA algorithm.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

The research leading to these results received funding from [Wenzhou Municipal Science and Technology Bureau] under Grant Agreement No[Y20240959].

ACKNOWLEDGMENTS

We would like to express our gratitude to all contributors to this article, with special thanks to the editors and reviewers.

REFERENCES

[1] Mohammadi-Balani A, Azar A, Taghizadeh-Yazdi M, et al. Golden Eagle Optimizer: A nature-inspired metaheuristic algorithm. Computers & Industrial Engineering, 2020, 152: 107050.
[2] Su H, Zhao D, Heidari AA, et al. RIME: A physics-based optimization. Neurocomputing, 2023, 532: 183-214.
[3] Trojovská E, M Dehghani, P Trojovský. Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm. Ieee Access, 2022, 10: 49445-49473.
[4] Xue J, B Shen. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. The Journal of Supercomputing, 2023. 79(7): 7305-7336.
[5] Jia H, Rao H, Wen C, et al. Crayfish optimization algorithm. Artificial Intelligence Review, 2023, 56(Suppl 2): 1919-1979.
[6] Xu D, J Yin. An Improved Black Widow Optimization Algorithm for Engineering Constrained Optimization Problems. IEEE Access, 2023, 11: 32476-32495.
[7] Ou Y, L Yu, A Yan. An Improved Sparrow Search Algorithm for Location Optimization of Logistics Distribution Centers. Journal of Circuits, Systems and Computers, 2023, 32(09).
[8] Faramarzi A, Heidarinejad M, Mirjalili S, et al. Marine Predators Algorithm: A nature-inspired metaheuristic. Expert systems with applications, 2020, 152: 113377.
[9] Elaziz MA, Mohammadi D, Oliva D, et al. Quantum marine predators algorithm for addressing multilevel image segmentation. Applied Soft Computing, 2021: 107598.
[10] Sun X, Wang G, Xu L, et al. Optimal performance of a combined heat-power system with a proton exchange membrane fuel cell using a developed marine predators algorithm. Journal of Cleaner Production, 2020: 124776.
[11] Ridha HM. Parameters extraction of single and double diodes photovoltaic models using Marine Predators Algorithm and Lambert W function. Solar Energy, 2020, 209: 674-693.
[12] Elaziz MA, Shehabeldeen TA, Elsheikh AH, et al. Utilization of Random Vector Functional Link integrated with Marine Predators Algorithm for tensile behavior prediction of dissimilar friction stir welded aluminum alloy joints. Journal of Materials Research and Technology, 2020, 9(5).
[13] Rai R, Dhal KG, Das A, et al. An Inclusive Survey on Marine Predators Algorithm: Variants andApplications. Archives of Computational Methods in Engineering, 2023, 30(5): 3133-3172.
[14] Fan Q, Huang H, Chen Q, et al. A modified self-adaptive marine predators algorithm: framework and engineering applications. Engineering with Computers, 2022.
[15] Yang W, Xia K, Li T, et al. A Multi-Strategy Marine Predator Algorithm and Its Application in Joint Regularization Semi-Supervised ELM. Mathematics, 2021, 9(3): 291.
[16] Sadiq AS, Amin, Mirjalili S, et al. Nonlinear marine predator algorithm: A cost-effective optimizer for fair power allocation in NOMA-VLC-B5G networks. Expert Systems with Application, 2022, 203: 117395-117395.
[17] Elaziz MA, Ewees AA, Yousri D, et al. An Improved Marine Predators Algorithm With Fuzzy Entropy for Multi -level Thresholding: Real World Example of COVID-19 CT Image Segmentation. IEEE Access, 2020(99): 1-1.
[18] Shaheen M, Yousri D, Fathy A, et al. A Novel Application of Improved Marine Predators Algorithm and Particle Swarm Optimization for Solving the ORPD Problem. Energies, 2020, 13(21).
[19] Sowmya R, M Premkumar, P Jangir. Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. Engineering Applications of Artificial Intelligence, 2024, 128.
[20] Mirjalili S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. Knowledge-Based Systems, 2016, 96.

[21] Tian Z, M Gai. Football team training algorithm: A novel sport-inspired meta-heuristic optimization algorithm for global optimization. Expert Systems with Applications, 2024, 245: 123088.
[22] Mirjalili, Lewis A. The Whale Optimization Algorithm. Advances in engineering software, 2016, 95: 51-67.
[23] Kumar, A., Wu G, Ali MZ, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm and Evolutionary Computation, 2020, 56: 100693.