Electronic Research Archive

ERA, 33(8): 4625–4657.
DOI: 10.3934/era.2025208
Received: 20 May 2025
Revised: 20 July 2025
Accepted: 29 July 2025
Published: 11 August 2025

https://www.aimspress.com/journal/ERA

Research article

A mirage search optimization algorithm that combines dual-chaos mapping and an opposition-based learning strategy

Yang Yang, Chaochuan Jia, Xiancun Zhou, Yu Liu and Maosheng Fu*

College of Electronic and Information Engineering, West Anhui University, Lu'an 237012, China

* **Correspondence:** Email: fums@wxc.edu.cn; Tel: +8615056408899.

Abstract: In this study, we propose a mirage search optimization algorithm that combines dual-chaos mapping and an opposition-based learning tactic, which we call the dual-chaos mapping opposition-based learning mirage search optimization (DMOMSO) algorithm. First, when initializing the population, a chaotic initialization method was used. This method takes advantage of the chaotic characteristics of iterative mapping to generate an initial population with high dispersion. Thus, the algorithm can more reliably reach the global optimum in a shorter time. Before the formal iteration process starts, a chaotic sequence is generated via chaotic mapping, which is used to update the iterative parameters, enabling them to vary dynamically throughout the iterative process. This helps the algorithm avoid becoming stuck in local optima, as well as enhancing its optimization performance and accelerating its convergence. In each iteration, the parameters are adjusted using the opposition-based learning method, which helps to cover the solution space more effectively, find good solutions faster, and explore a wider area. We conducted simulation experiments to assess the DMOMSO algorithm's effectiveness, in terms of its performance on the CEC2017 and CEC2022 benchmark function sets. In this way, we carried out a comparative analysis between the DMOMSO algorithm and seven other algorithms, namely, MSO, SSA, AO, FOX, WOA, HHO, and COA. The experimental results indicated that the DMOMSO algorithm outperforms the original mirage search optimization algorithm, being able to more effectively find optimal solutions and converging faster. The eight considered algorithms were also tested in the context of practical optimization problems, including the design of hydrostatic thrust bearings, the optimization of Himmel Blau's function, and industrial refrigeration system design. Based on the results obtained, DMOMSO is more effective for solving complex engineering problems.

Keywords: mirage search optimization algorithm; chaos mapping; opposition-based learning;

hydrostatic thrust bearing; Himmel Blau’s Function

1. Introduction

With the increasing complexity of industrial engineering and the growing demand for solving high-dimensional problems, intelligent optimization algorithms have become a core tool for addressing complex optimization problems, thanks to their advantages such as not relying on the mathematical properties of problems and strong global search capabilities. Their development process can be roughly divided into different stages, and the representative algorithms of each period have laid an important foundation for subsequent research.

In the early stage of the development of intelligent optimization algorithms, basic algorithms inspired by natural phenomena and biological laws gradually took shape [1]. The genetic algorithm (GA) [2] imitates the operations of selection, crossover, and mutation occurring in the process of biological evolution of a population and has achieved remarkable results in many fields. For instance, when attempting to solve the traveling salesman problem (TSP), scholars have used the genetic algorithm to effectively reduce the complexity of path planning and successfully find a better path [3]. In the scenario of oil transportation scheduling, a genetic algorithm based on multi-layer coding has been used to address the production and transportation collaborative scheduling problem considering marginal wells in low-permeability oilfields. It performed much better than the exact algorithm, providing strong support for reducing transportation costs while improving production efficiency [4]. The simulated annealing algorithm (SA) [5], designed based on the principle of physical annealing, has demonstrated unique value in many fields. In [6], a double-population genetic algorithm was successfully applied in the context of the lightweight design of a cross-sea bridge. As a result, the material utilization rate increased by 18%, the structure’s natural frequency rose by 12%, and steel consumption was reduced by over 200 tons. For example, using simulated annealing, Zhejiang Chuang Xin optimized the main pattern’s process window for the construction of SRAF rules in the integrated circuit lithography process, significantly improving the accuracy and efficiency of the pattern design [7,8]. In another study, considering the design stage for the scan chain of a three-dimensional on-chip system, the cores were reasonably allocated to each layer of wafers according to the length of the scan chains inside the cores determined using the simulated annealing algorithm. Through the comprehensive consideration of factors, such as the test time and TSV bonding cost, the total test cost was effectively reduced [9].

In recent years, with the development of the swarm intelligence theory, optimization algorithms centered on “group collaboration” have become a hotspot for research. The particle swarm optimization (PSO) [10] algorithm mimics the foraging behaviors of bird flocks and has enabled remarkable achievements in the context of machine learning parameter tuning. In [11], the researchers proposed a new multi-objective particle swarm optimization (PSO) algorithm and used it to optimize the intelligent parameters of the air-floating piston, which is the core component of domestic frictionless pneumatic actuators. An improved particle swarm optimization (PSO) algorithm integrating “Gaussian mutation” and “fuzzy theory” was proposed to achieve ultra-high-precision control of the pneumatic force servo system in [12]. The researchers in [13] propose a “hybrid Gaussian mutation particle swarm optimization algorithm” to realize the intelligence and efficiency of a cylinder design. In [14], in the field of radiation therapy, researchers have employed the multi-objective particle swarm optimization (MOPSO) algorithm. This algorithm has increased the dose coverage rate of the tumor target area, reduced the maximum dose received by the spinal cord, and significantly shortened the design time of the treatment plan.

Inspired by the pheromone-releasing behaviors of ants during foraging, the ant colony optimization (ACO) [15] algorithm has been widely used in contexts such as vehicle path planning. For example, researchers have applied this algorithm to real-time data collected using sensors along urban main roads for the dynamic optimization of signal timing and path planning [16]. In engineering design, the gray wolf optimization algorithm [17] has been used in the design of mechanical structures, thus enhancing mechanical performance. In economics, the artificial bee colony (ABC) algorithm [18] has been utilized to optimize investment portfolios, allowing for the effective allocation of resources. For instance, Ren Tong, the chief analyst of quantitative analysis and fund evaluation at China Merchants Securities, has used the genetic algorithm to reduce the dimensionality and standardize stock data, design a fitness function, and build a portfolio, achieving good results.

Although algorithms have achieved results in many fields, they have limitations such as slow convergence speed and a tendency to fall into local optima when dealing with complex problems with high dimensionality, multiple peaks, and strong constraints. Against this background, the mirage search optimization (MSO) [19] algorithm, proposed in 2025, has emerged as a representative of new intelligent optimization algorithms. This algorithm simulates the optical refraction phenomenon of "mirages" in deserts and balances global exploration and local exploitation through a dynamic virtual-real search mechanism, demonstrating unique advantages in problems involving asymmetric solution spaces. However, when handling ultra-complex problems, MSO has issues such as insufficient dispersion of the initial population and limited convergence accuracy in the later stages of iteration, which urgently require targeted improvements.

Here, we focus on the mirage search optimization (MSO) algorithm and propose a mirage search optimization algorithm (DMOMSO) that combines a dual-chaos mechanism with an opposition-based learning strategy. In the population initialization stage, chaotic initialization is introduced. By leveraging the chaotic characteristics [20] of iterative mapping to generate an initial population with high dispersion, the probability of the algorithm finding the global optimal solution is increased. Before the start of iterations, a Fuch mapping [21] is used to generate a chaotic sequence, which is then incorporated into the update of iterative parameters. This enables the parameters to change more flexibly during iterations, preventing the algorithm from falling into local optima and improving the algorithm's search capability and convergence speed. In each iteration, Beta opposition-based learning [22] is introduced to adjust parameters, enhancing the population's opposition search capability in the solution space, accelerating the convergence speed of high-quality solutions, further expanding the search range, and improving the algorithm's ability to find the global optimal solution. These three improvement strategies are organically integrated and dynamically balanced, forming a synergistic optimization effect.

The remainder of this article is structured as follows: In Section 2, the MSO algorithm is introduced. In Section 3, we explain the three enhancement methods used to develop DMOMSO in detail. In Section 4, we describe the experimental analysis. In Section 5, we detail the analyses involving optimization applications. In Section 6, we provide the conclusions.

2. Mirage search optimization algorithm

The MSO algorithm, inspired by mirages, is a type of meta-heuristic optimization algorithm. Its mathematical model is detailed in the following subsections:

2.1. Generating initial solutions

The original set of solutions is formulated as a matrix X that represents the initial position distribution, where each element within it corresponds to a particular solution's coordinate in a certain dimension,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{p1} & \cdots & x_{pd} \end{bmatrix}. \tag{1}$$

Here, p represents the number of initial solutions (or observation positions) and d represents the number of variables. The initial position values for each observation position in every dimension are randomly generated within the upper and lower bounds of the considered variables. The formula for initializing the position is as follows:

$$x_i = lb_i + r \cdot \left(ub_i - lb_i\right), i = 1, 2, \cdots d \tag{2}$$

where x_i denotes the i-th component of a position vector; lb_i and ub_i are the lower and upper bounds of the i-th variable, respectively, and r is a uniform random number within [0, 1].

2.2. Superior mirage method

A superior mirage causes viewers to see distant objects as virtual images. This phenomenon mostly results from reflection in the optical path.

First, the refractive index stratification line is simulated. Considering angle *A* between the reflected light and the horizontal reference line, angle *B* between the incident light and the reflected light, angle *C* of the incident light relative to the refractive index stratification line, and angle *D* of the incident light relative to the normal of the horizontal reference line, the formula for the exploration point's position is derived as follows:

$$\Delta x_{lower} = \frac{sinB \cdot h \cdot sinC}{sinDsinA} \tag{3}$$

where Δx_{lower} represents the horizontal offset of the target point from its origin, while h indicates the vertical distance from the starting point to the refractive boundary. As increasing h can aid in escaping from local optima, h is defined as the gap between the current and optimal positions:

$$h_{ij} = | gbest_j - x_{ij} | \cdot rand + 1 \tag{4}$$

where h_{ij} represents the value at the jth initial position in the ith dimension (to ensure that the individual update step size is appropriate, we set the minimum value h_{ij} to be no less than 1), $gbest_j$ denotes the optimal value in the jth dimension, and $rand$ is a random value within [0, 1]. In particular,

h_{ij} satisfies Eq (5):

$$1 \le h_{ij} \le 10 \cdot arctanh\left(-\frac{t}{t_{max}}+1\right)+1 \tag{5}$$

where t represents the present iteration, while t_{max} is the maximum number of iterations. By maintaining h as non-zero and within a wide range, the global optimization ability of the proposed strategy is preserved, and relevant settings can be made according to t and t_{max}. The step size of a single individual within an iteration is determined by two values: α and β. Here, β denotes the tilt of the refractive index layer relative to the ground, while α is the angle of incident light relative to the horizontal normal. For the incident light to hit the refractive layer interface and produce reflected light, the condition $\frac{\pi}{2}-\beta>\alpha$ must be met. In this work, the values of α and β are generated using Eq (6):

$$\begin{cases} \alpha = rand \cdot \dfrac{\pi}{9} \\ \beta = rand \cdot \left(\dfrac{\pi}{4}-\dfrac{\alpha}{2}\right) \end{cases}. \tag{6}$$

When $\beta<2\pi$, the incident light hits the right side of the refractive stratification line at an angle of $\alpha=\beta$. As the angle between the incident light and the line is 2π, the reflected light will be perpendicular to the incident light.

In the MSO algorithm, three scenarios of incident light are considered, with $\alpha=\beta$ and α=2π serving as the boundaries for differentiation. In addition, as the cases of $\beta<2\pi$ and $\beta>2\pi$ are similar, for sake of simplify, in the case of $\beta>2\pi$, $a=\pm1$ is set to indicate whether the current situation is to the left/right of the line's normal direction.

Situation 1: When the incident light meets the condition of being on the left side of the normal of the horizontal reference line, considering Eqs (3)–(6) comprehensively, the step size of an individual in the mirage-based search is calculated as follows:

$$\Delta x_{lower} = a \cdot \frac{sin(\pi-2\alpha-2\beta)\cdot h \cdot sin\left(\frac{\pi}{2}+\beta\right)}{sin(\alpha-\beta)sin(\alpha-2\beta)}. \tag{7}$$

Situation 2: When the incident light is right of the reference line's normal and satisfies certain conditions (i.e., $\beta<\alpha<\frac{\pi}{2}$), the step-size calculation in the superior mirage search, based on Eqs (3)–(6), is as follows:

$$\Delta x_{lower} = a \frac{sin(\pi-2\alpha+2\beta)\cdot h \cdot sin\left(\frac{\pi}{2}-\beta\right)}{sin(\pi-\alpha+\beta)sin(\alpha-2\beta)}. \tag{8}$$

Situation 3: When the incident light is right of the reference line's normal and satisfies certain conditions (i.e., $\beta<\alpha<\frac{\pi}{2}$), we can use Eqs (3)–(6) to calculate the step size of an individual in the superior mirage search:

$$\Delta x_{lower} = a\frac{\sin(-\pi+2\alpha-2\beta)\cdot h\cdot\sin\left(\frac{\pi}{2}-\beta\right)}{\sin(\pi-\alpha+\beta)\sin(\pi-\alpha+2\beta)}. \tag{9}$$

Finally, based on Eqs (7)– (9), the adjusted formula for the mirage's origin point is as follows:

$$x_{ij}^{t+1} = x_{ij}^{t} + \Delta x_{lower} \tag{10}$$

where t denotes the current iteration.

The superior mirage strategy has improved exploration capabilities. This approach enables the detection of distant targets using the principle of light reflection, which enables individuals to more effectively perform global update operations. In the MSO algorithm, to realize this design idea, the superior mirage is divided into three cases corresponding to Eqs (7)– (9), respectively. According to these formulas, the positions of all individuals in the algorithm can be updated globally, and they can reach any position.

2.3. Inferior mirage strategy

The inferior mirage presents an enlarged virtual image of an object, which is mainly due to refraction in the light path. Based on this idea, in this work, a local search mode according to the formation mechanism of the inferior mirage is developed.

First, a reference line is set at a fixed distance from the starting point. Next, considering the distance between the horizontal and layered reference lines, a refractive index gradient $2h$ is established. Furthermore, an observation point and its reference line are placed at a defined distance from the starting position. During the search process, the observation points act as a light source. The refracted light's intersection with the horizontal line marks the real object, while the extended incident light's intersection represents the mirage. An inferior mirage is observed nearby with a wide field of view, aiding in local optimization. Its distance h and direction relative to the current solution can derived from Eqs (11) or (12).

Case 1: When a member is not the group's top performer:

$$\begin{cases} h_{ij} = |gbest_j - x_{ij}|\cdot rand \\ D = \dfrac{(gbest_j - x_{ij})\cdot rand}{h_{ij}} \end{cases}. \tag{11}$$

Case 2: When an individual is the best among the group at present:

$$\begin{cases} h_{ij} = |\pm 0.05\cdot rand| \\ D = \dfrac{\pm 0.05\cdot rand}{h_{ij}} \end{cases}. \tag{12}$$

When the value of D is 1, the current solution approaches the optimal solution's upper limit; moreover, when the value of D is −1, the solution approaches the optimal solution's lower bound.

Regarding the inclination between the line of refractive index layering and the reference layering line, its value ranges from 0 to $\frac{\pi}{2}$. This angle is initialized and set using Eq (13):

$$\gamma = \frac{\pi(t_{max} - t)}{2t_{max}} \cdot rand \tag{13}$$

where γ denotes the angle between the refractive stratification line and the reference line.

For the angle φ between the light direction and surface normal, there are two limiting conditions. When the refracted ray first returns to its starting position, the associated angle is the minimum-limit angle. When the refracted ray first reaches the junction of the refractive index layering line and the horizontal reference line, this angle represents the maximum-limit angle. The angle is set initially using Eq (14):

$$\varphi = \left(\arctan\left(\frac{1}{2\tan\gamma} \right) - \arctan\left(\frac{\sin\gamma\cos\gamma}{1+\sin^2\gamma} \right) \right) \cdot rand + \arctan\left(\frac{\sin\gamma\cos\gamma}{1+\sin^2\gamma} \right) \tag{14}$$

Based on the Fresnel reflection principle, using the refraction angle ω formed between the incoming light and the perpendicular line at the refractive edge, we can determine the angle of incidence and the refractive indices on both sides of the boundary:

$$\omega = \arcsin\left(\frac{n_2}{n_1} \sin(\varphi + \gamma) \right) \tag{15}$$

where the refractive index below the separation line is indicated as n_1 and the refractive index above it is denoted as n_2, with $n_1 < n_2$. Based on this, the shift in the current position can be calculated using Eqs (11)–(15):

$$\Delta x_{upper} = \frac{h}{\tan\gamma} - \frac{\left(\frac{h}{\sin\gamma} - \frac{h\sin\varphi}{\cos(\varphi+\gamma)}\right)\cos\omega}{\cos(\omega-\gamma)} \tag{16}$$

Through a comprehensive analysis and calculation of Eqs (11), (12), and (16), we can obtain the new formula for the inferior mirage's starting point. The specific derivation process is based on the relevant principles and calculation logic of these formulas:

$$x_{ij}^{t+1} = x_{ij}^{t} + D\Delta x_{upper} \tag{17}$$

The inferior mirage strategy, based on the principle of light refraction, enables excellent local search capabilities. This is because the inferior mirage can show an enlarged virtual image of an object, which helps individuals to better search in a local area. In the MSO algorithm, the search process based on the inferior mirage is achieved through Eq (16). All individuals in the algorithm perform detailed exploration and development within a local range according to this formula.

3. Improved mirage search optimization algorithm

Similar to other meta-heuristic algorithms, the MSO algorithm has certain drawbacks. Its global optimization ability is not satisfactory, and its convergence speed is slow. Therefore, we aimed to improve the MSO algorithm. To address these issues, an optimization plan that combines multiple intelligent mechanisms is put forward.

First, when obtaining the initial population, chaotic initialization was utilized. Due to the chaotic features of iterative mapping, an initial population with high dispersion was created. This caused the population to be spread more evenly across the search space, avoiding the blind spots caused by traditional random initialization and increasing the chance that the algorithm discovers the global optimum.

Before the iterative process began, chaotic mapping was performed to generate a chaotic sequence, which was then used to improve the iterative parameters. This chaotic sequence was characterized by properties such as randomness, ergodicity, and regularity, and incorporating it into the iterative parameter update step enabled the parameters to change more flexibly throughout the iterative process. This helped prevent the algorithm from becoming trapped in local optima, improved its searching power, and enabled it to converge faster.

In each iteration, opposition-based learning was carried out to tweak the parameters. This strengthened the population's reverse search ability in the solution space, sped up the convergence to optimal solutions, widened the search area, and increased the algorithm's power to find the global optimum.

3.1. Iterative map

The traditional population initialization method used in MSO may lead to an uneven initial population distribution and insufficient randomness in the search process. This increases the probability of becoming stuck in local optima. Iterative chaotic mapping is a non-linear mapping function composed using trigonometric functions, which can generate complex chaotic sequences. In this study, this iterative mapping strategy was used to create a chaotic sequence during the population initialization stage. A small difference in the initial value can cause a large difference in the result after multiple iterations, presenting complex dynamic behavior that seems random but is determined by a formula. The formula for initializing the population with the iterative chaotic map is as follows:

$$x_{i+1} = \sin\left(\frac{a\pi}{x_i}\right) \tag{18}$$

where x_i represents the individual in iteration i.

When a is small (e.g., $a \to 0$), the mapping tends to be linear: $x_{i+1} \approx \frac{a\pi}{x_i}$, and the system may converge or diverge. As a increases, nonlinearity strengthens, and periodic oscillations or chaos may occur.

The range of values a is selected as: $a \in [0.1, 2.0]$ with a step size of $\Delta a = 0.001$. When $a < 0.5$, the system may converge to a stable point. When $a \in [0.5, 1.5]$, bifurcations and chaos may appear. The bifurcation diagram is shown in Figure 1. When $a > 1.5$, it may lead to numerical instability due to violent oscillations caused by $\frac{a\pi}{x_i}$.

The bifurcation diagram demonstrates a classic transition from periodic to chaotic behavior. At $0.5 \le a \le 0.6$, the diagram exhibits tree-like branching patterns (period-doubling bifurcations), characteristics of systems transitioning from order to chaos. At $a \approx 0.8$, a stable periodic orbit emerges abruptly within the chaotic regime, visible as discrete point clusters. At $a \approx 0.7$, the system resides in a continuous banded region, confirming chaotic dynamics. This parameter is ideal for optimization algorithms, as chaos enhances global search capability while avoiding local optima.

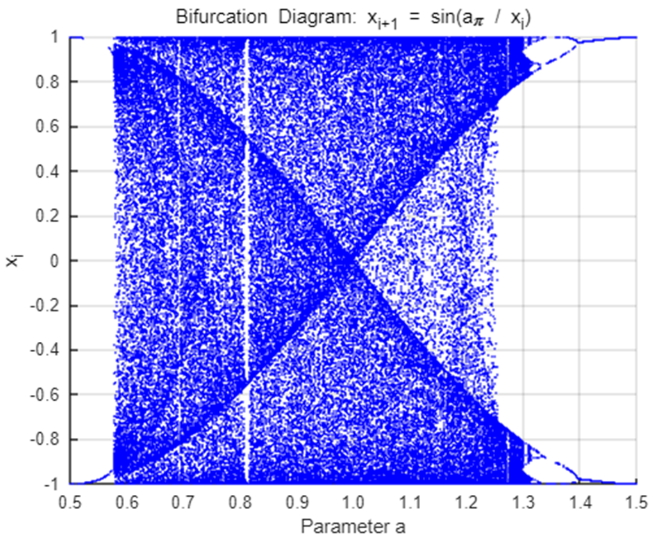


Figure 1. Bifurcation diagram.

3.2. Fuch map

The MSO algorithm lacks an adaptive adjustment mechanism in its parameter iteration process. In the early search stage, a larger search step might be needed to explore the search space more effectively; moreover, during the subsequent search phase, a smaller step size is required for a more detailed search. If these parameters are not adjusted dynamically according to the number of iterations or the search situation, it will affect the algorithm's convergence speed. Considering the above, the Fuch map strategy is introduced. Before the iteration starts, a chaotic sequence is generated based on the largest iteration number. At the start of the cycle, the iteration count is combined with the chaotic sequence to optimize the MSO parameters. This enables the parameters to change more flexibly during the iteration, prevents the algorithm from becoming stuck in local optima, and improves the algorithm's search efficiency and convergence capability. The Fuch map strategy formula is defined as follows:

$$x_{i+1} = \cos\left({}^{1}\!/\!{}_{x_i^{\,2}} \right) \tag{19}$$

where x_i is the individual in iteration i.

3.3. *Beta opposition-based learning strategy*

Randomly opposite learning with a beta distribution is an improved strategy designed for use in optimization algorithms. It aims to boost the efficiency of such algorithms, especially through increasing population diversity and avoiding local optima. This strategy is based on the concept of opposition-based learning (OBL). By generating opposite solutions for the current population of individuals, it expands the algorithm's search space. The beta distribution is also introduced, which is a flexible probability distribution that helps to generate opposite solutions more intelligently, thus increasing population diversity. This helps the MSO algorithm to better explore the search space while avoiding local optima.

Based on a random number, one of the strategies presented in Eqs (20) and (21) is chosen.

$$
\begin{aligned}
&\breve{x}_{i,j}(t)=(b_j-a_j)\cdot Beta(\alpha,\beta)+a_j\\
&\alpha=\begin{cases} spread\cdot peak, & if \quad mode<0.5\\ spread, & else\end{cases}\\
&\beta=\begin{cases} spread, & if \quad mode<0.5\\ spread\cdot peak, & else\end{cases}\\
&spread=\left(\frac{1}{\sqrt{normDiv}}\right)^{1+N(0,0.5)}\\
&peak=\begin{cases} \dfrac{(spread-2)mode+1}{spread(1-mode)}, & if \quad mode<0.5\\ \dfrac{2-spread}{spread}+\dfrac{spread-1}{spread\cdot mode}, & else\end{cases}\\
&mode=\frac{(a_j+b_j-x_{i,j}(t))-a_j}{b_j-a_j}
\end{aligned}
\tag{20}
$$

where $Beta(\alpha,\beta)$ is the statistical distribution of the beta kind, characterized by the parameters α and β; $N(\mu,\sigma)$ denotes a Gaussian distribution with mean μ and variance σ; and a_j and b_j stand for the lower and upper limits in the j th dimension, respectively, where $a_j\neq b_j$. The expansion factor $spread$ is calculated according to $normDiv$. For every individual, the mode value is calculated based on its coordinates in the search space. Then, the parameters α and β of the beta distribution are calculated from the mode value. Using the calculated α and β, a random number based on the beta distribution is generated and an opposite solution is obtained through a linear transformation. This opposite solution is more reasonably distributed in the search space, helping the algorithm to better explore new areas.

Using the midpoint ($mode=0.5$) as the boundary, the search interval $[a_j,b_j]$ is divided into the “upper sub–interval” (where $mode<0.5$ and solutions are close to b_j) and the "lower sub - interval" (where $mode\geq 0.5$ and solutions are close to a_j). This division serves two purposes. First, it covers all possible solution positions; no matter how a solution is distributed within the interval, it must belong to either the upper sub - interval or the lower sub - interval, with no ambiguous boundaries. Second, the midpoint is the most symmetric partition point of the interval, ensuring that a symmetric opposition - based learning strategy is applied to solutions near the upper or lower bounds.

The algorithm generates opposite solutions via the Beta distribution $Beta(\alpha,\beta)$, and its core lies in controlling the skewness of the distribution through the magnitude relationship between α and β. If $\alpha<\beta$, the distribution is left - skewed (with the peak close to the left endpoint a_j), so opposite solutions are more likely to appear near a_j. If $\alpha>\beta$, the distribution is right - skewed (with the peak close to the right endpoint b_j), so opposite solutions are more likely to appear near b_j.

When $mode<0.5$ (solutions are close to b_j), $\alpha=spread\cdot peak$ and $\beta=spread$, and it can be deduced that $\alpha<\beta$ (left - skewed), causing the opposite solutions to tend to explore near a_j

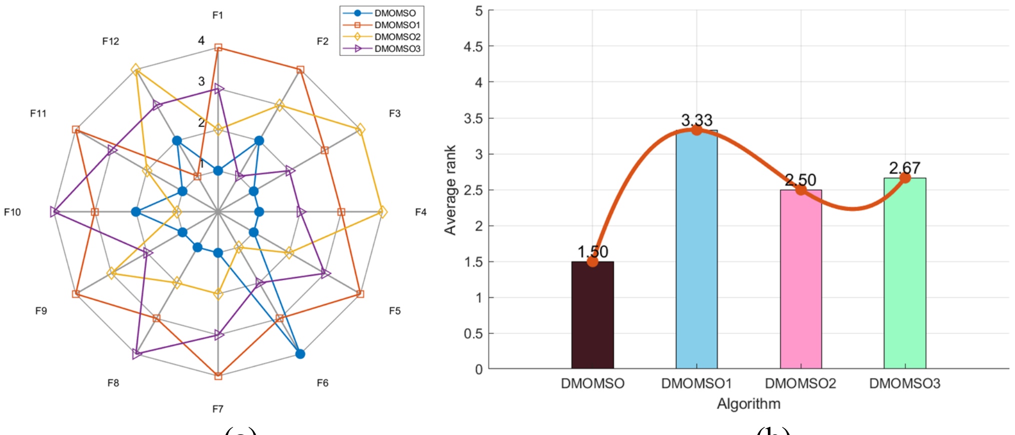
(filling the space opposite the upper bound). When $mode \geq 0.5$ (solutions are close to a_j), $\alpha = spread$ and $\beta = spread \cdot peak$, and it can be deduced that $\alpha > \beta$ (right - skewed), causing the opposite solutions to tend to explore near b_j (filling the space opposite the lower bound).

$$\begin{aligned}
&\breve{x}_{i,j}(t) = (b_j - a_j) \cdot Beta(\alpha, \beta) + a_j \\
&\alpha = \begin{cases} spread \cdot peak, & if \;\; mode < 0.5 \\ spread, & else \end{cases} \\
&\beta = \begin{cases} spread, & if \;\; mode < 0.5 \\ spread \cdot peak, & else \end{cases} \\
&spread = 0.1\sqrt{normDiv} + 0.9 \\
&peak = \begin{cases} \dfrac{(spread - 2)mode + 1}{spread(1 - mode)}, & if \;\; mode < 0.5 \\ \dfrac{2 - spread}{spread} + \dfrac{spread - 1}{spread \cdot mode}, & else \end{cases} \\
&mode = \frac{x_{i,j}(t) - a_j}{b_j - a_j}.
\end{aligned} \tag{21}$$

Integrating these multiple mechanisms, the improved DMOMSO algorithm is obtained. In the following sections, simulation experiments and comparative analyses with existing algorithms are detailed to validate the improved performance of the proposed approach.

3.4. Ablation study of DMOMSO

We conducted an ablation study to identify the contribution of each newly added strategy. Using MATLAB simulation software, we tested four algorithms on the CEC2022 [23] standard dataset: DMOMSO without Iterative Map (named DMOMSO1), DMOMSO without Fuch Map (DMOMSO2), DMOMSO without Beta Opposition-Based Learning Strategy (DMOMSO3), and the original DMOMSO. The experimental settings were as follows: A population size of 30, a maximum of 300 iterations, and each algorithm running independently 30 times for each test function.



(a) (b)

Figure 2. Radar chart (a) and ranking chart (b) of the ablation study.

In Figure 2(a), the radar chart of DMOMSO shows the smallest area variation. It ranked first in eight functions and second in three. By contrast, the radar chart areas of the other three algorithms were significantly larger, which demonstrates that the three new strategies have a positive impact on algorithm performance. Notably, in Figure 2(b), DMOMSO achieved the lowest average fitness value (1.50) and performed best in the comprehensive test. The ranking chart showed that DMOMSO1 came in fourth, DMOMSO2 second, and DMOMSO3 third—indicating that Fuch Map played the most important role in improving the algorithm.

3.5. Pseudo-code of DMOMSO

```
Input: $popsize, t_{\max}, Dim$
Initialize the positions of solutions using the chaotic strategy in Eq (18).
Evaluate the fitness value of every member of the current population.
Order all present individuals from highest to lowest fitness value.
While ($t < t_{\max}$) do
    Generate a chaotic sequence according to Eq (19) to improve the number of individuals n as an iterative parameter.
    Form an array $popn$ comprising the current optimal solution and n sub-optimal solutions.
    For $i = 1$ to $popn$ /* High-level mirage search* /
        Obtain $h$, $\alpha$, and $\beta$ according to Eqs (4)–(6).
        Using the values of $h, \alpha$, and $\beta$, perform random updates using Eqs (7)–(10).
        Perform boundary handling on the renewed individuals.
        If $f(gbest) < f(X_i)$
            $gbest = X_i$ and $f(gbest) = f(X_i)$
        End if
    End for
    Top performers $popsize$ create the next population.
    For $i = 1$ to $popsize$ /* Low-level mirage search */
        Obtain $h$, $D$, $\gamma$, and $\varphi$ according to Eqs (11)–(14).
        Using the values of $h$, $D$, $\gamma$ and $\varphi$, perform random updates using Eqs (15)–(17).
        Perform boundary handling on the adjusted members.
        If $f(gbest) < f(X_i)$
            $gbest = X_i$ and $f(gbest) = f(X_i)$
        End if
    End for
    Generate new solutions using beta opposition-based learning according to Eqs (20) and (21).
    The pop best individuals form a new population.
End while
Provide the optimal solution $X$ and function value $f$.
```

3.6. Flowchart of DMOMSO

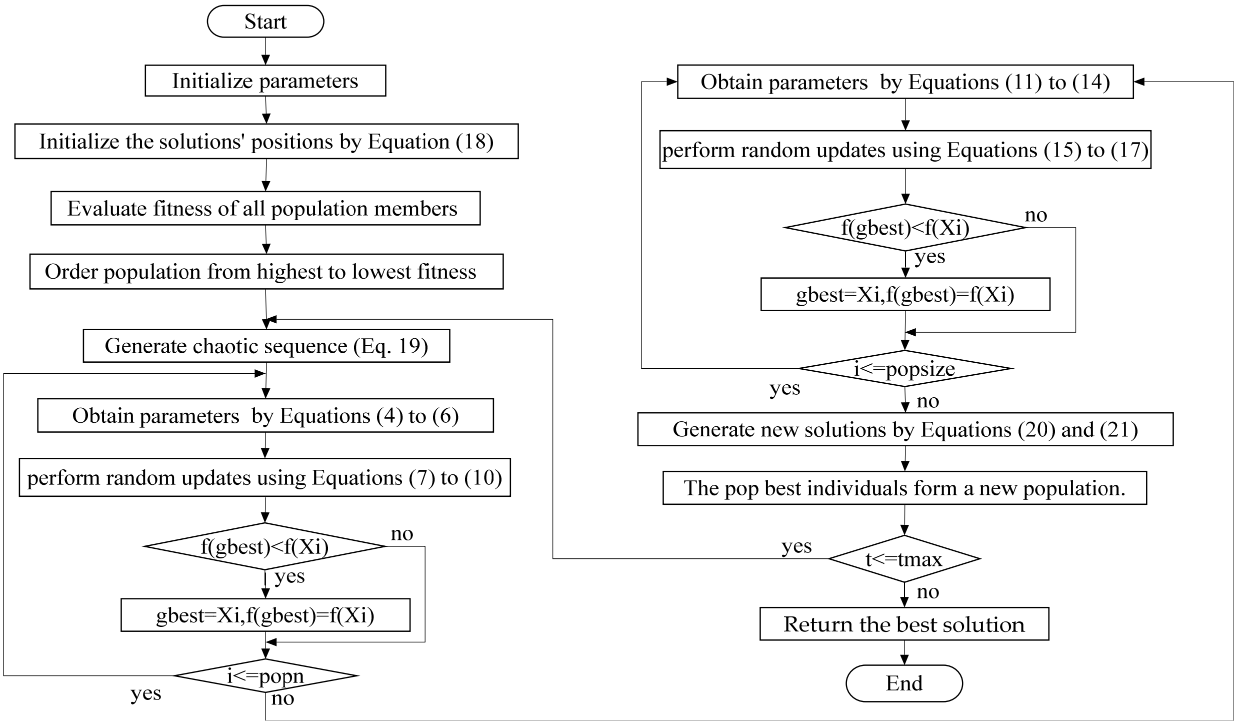

Start
Initialize parameters
Initialize the solutions' positions by Equation (18)
Evaluate fitness of all population members
Order population from highest to lowest fitness
Generate chaotic sequence (Eq. 19)
Obtain parameters by Equations (4) to (6)
perform random updates using Equations (7) to (10)
f(gbest)<f(Xi)
no
yes
gbest=Xi,f(gbest)=f(Xi)
i<=popn
yes
no
Obtain parameters by Equations (11) to (14)
perform random updates using Equations (15) to (17)
f(gbest)<f(Xi)
no
yes
gbest=Xi,f(gbest)=f(Xi)
i<=popsize
yes
no
Generate new solutions by Equations (20) and (21)
The pop best individuals form a new population.
yes
t<=tmax
no
Return the best solution
End

Figure 3. Flowchart of DMOMSO.

4. Simulation experiments and comparative analyses

To test the practicality of the DMOMSO algorithm, we compared it with seven algorithms on the CEC2017 [24] and CEC2022 benchmark function sets, namely, MSO, SSA [25], AO [26], FOX [27], WOA [28], HHO [29], and COA [30]. Through an analysis of the optimization results, convergence curves, box plots, and outcomes of the tests based on the ranked sums of these algorithms on the benchmark functions, we can fully demonstrate the performance advantages of the DMOMSO algorithm. For each of the algorithms, simulation experiments were performed 30 times. Each algorithm used a population of 30 agents, ran for 300 iterations at most, and operated in a 10-dimensional space; other parameters were set as per the algorithms' respective references. Table 1 offers a comprehensive overview of the algorithms employed in this research, along with their respective control parameters.

4.1. Experiments on CEC2017

According to the optimization results in Table 2, the different algorithms performed quite differently on each benchmark function. For example, on the F1 function, the average value of the DMOMSO was 1.80E+03, the MSO algorithm was 2.72E+03, and that of the SSA algorithm was 4.57E+03. For most functions (F1–3, F5–27, F29–F30), the DMOMSO algorithm obtained better minimum and average values, showing clear advantages over the other algorithms. This means that the DMOMSO algorithm has a good ability to find optimal solutions in a wide range of function optimization problems and can come very close to the theoretical optimal value.

Table 1. Parameter settings.

Algorithm	Name of parameters	Value of parameters
DMOMSO	a, Critical parameter	0.7
	Mode, Threshold of the decision boundary	0.5
MSO	--	--
SSA	Pd, Proportion of discoverers	0.2
	ST, Safety Threshold	0.8
AO	α,δ, Attack strategy parameter	0.1
FOX	c1, Low jump probability	0.18
	c2, High jump probability	0.82
WOA	b, Spiral parameter	1
HHO	--	--
COA	--	--

From the convergence curve graphs in Figure 4, the convergence speeds and trends of the different algorithms varied. For many functions (F1, F4–11, F15–18, F20–24, F26, F29–F30), the DMOMSO algorithm converged faster. Its curve dropped, and it quickly reached the optimal solution. In contrast, the other algorithms either converged slowly or presented large fluctuations. This indicates that the DMOMSO algorithm could locate superior solutions more rapidly during the optimization of the functions considered, demonstrating its improved efficiency.

Table 2. Comparative data for CEC2017.

F		DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
F1	min	1.03E+02	**1.01E+02**	2.37E+02	1.49E+07	5.28E+02	8.53E+06	1.81E+06	3.22E+09
	std	1.93E+03	3.52E+03	3.70E+03	2.35E+08	9.55E+08	2.00E+08	5.67E+06	4.35E+09
	avg	**1.80E+03**	2.72E+03	4.57E+03	2.12E+08	1.04E+09	2.12E+08	7.02E+06	1.11E+10
F3	min	**3.00E+02**	1.03E+03	3.07E+02	1.99E+03	8.48E+02	2.64E+03	1.15E+03	4.75E+03
	std	2.91E+02	5.13E+03	2.48E+02	1.27E+03	7.13E+03	9.06E+03	7.98E+02	2.73E+03
	avg	5.59E+02	7.57E+03	**5.20E+02**	4.55E+03	4.42E+03	1.14E+04	2.31E+03	1.16E+04
F4	min	**4.00E+02**	4.00E+02	4.00E+02	4.07E+02	4.01E+02	4.07E+02	4.01E+02	5.65E+02
	std	1.92E+01	1.20E+01	1.63E+01	2.32E+01	9.28E+01	5.11E+01	4.31E+01	3.41E+02
	avg	4.11E+02	**4.08E+02**	4.09E+02	4.36E+02	4.92E+02	4.62E+02	4.46E+02	1.02E+03
F5	min	**5.06E+02**	5.16E+02	5.13E+02	5.19E+02	5.40E+02	5.15E+02	5.18E+02	5.70E+02
	std	1.07E+01	9.27E+00	1.63E+01	1.33E+01	2.60E+01	2.33E+01	2.61E+01	1.68E+01
	avg	**5.19E+02**	5.30E+02	5.42E+02	5.38E+02	5.87E+02	5.58E+02	5.62E+02	5.92E+02
F6	min	**6.00E+02**	6.05E+02	6.00E+02	6.14E+02	6.34E+02	6.12E+02	6.19E+02	6.26E+02
	std	4.82E+00	5.72E+00	1.15E+01	6.24E+00	7.41E+00	1.54E+01	1.21E+01	8.24E+00
	avg	**6.04E+02**	6.16E+02	6.13E+02	6.25E+02	6.57E+02	6.42E+02	6.44E+02	6.45E+02

Continued on next page

F		DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
F7	min	**7.15E+02**	7.16E+02	7.51E+02	7.34E+02	8.06E+02	7.49E+02	7.41E+02	7.57E+02
	std	1.14E+01	1.65E+01	2.00E+01	1.52E+01	6.01E+00	2.76E+01	2.01E+01	2.44E+01
	avg	**7.36E+02**	7.51E+02	8.04E+02	7.62E+02	8.16E+02	7.96E+02	8.01E+02	8.05E+02
F8	min	**8.06E+02**	8.08E+02	8.15E+02	8.14E+02	8.32E+02	8.20E+02	8.11E+02	8.33E+02
	std	8.19E+00	8.66E+00	1.08E+01	7.21E+00	1.70E+01	1.57E+01	1.21E+01	1.07E+01
	avg	**8.17E+02**	8.24E+02	8.33E+02	8.28E+02	8.51E+02	8.49E+02	8.35E+02	8.55E+02
F9	min	**9.00E+02**	9.02E+02	9.24E+02	9.56E+02	1.36E+03	1.16E+03	1.12E+03	1.12E+03
	std	5.44E+01	2.08E+02	3.27E+02	1.09E+02	1.71E+02	5.86E+02	2.01E+02	2.26E+02
	avg	**9.51E+02**	1.12E+03	1.52E+03	1.09E+03	1.77E+03	1.77E+03	1.56E+03	1.50E+03
F10	min	1.24E+03	1.72E+03	**1.17E+03**	1.35E+03	1.68E+03	1.45E+03	1.32E+03	2.10E+03
	std	2.27E+02	3.55E+02	3.45E+02	3.68E+02	4.28E+02	3.57E+02	3.24E+02	2.29E+02
	avg	**1.64E+03**	2.26E+03	2.06E+03	2.05E+03	2.45E+03	2.15E+03	2.03E+03	2.56E+03
F11	min	**1.10E+03**	1.11E+03	1.10E+03	1.15E+03	1.13E+03	1.13E+03	1.12E+03	1.20E+03
	std	1.61E+01	7.41E+01	5.78E+01	3.18E+02	2.21E+03	8.75E+02	8.30E+01	2.33E+03
	avg	**1.12E+03**	1.18E+03	1.15E+03	1.33E+03	2.36E+03	1.43E+03	1.20E+03	2.42E+03
F12	min	3.25E+03	1.65E+05	**2.38E+03**	3.07E+04	2.44E+04	7.87E+04	2.46E+04	4.16E+06
	std	8.95E+04	3.03E+06	9.65E+04	5.54E+06	1.81E+06	6.12E+06	3.34E+06	1.33E+08
	avg	6.52E+04	2.95E+06	**3.43E+04**	5.56E+06	1.40E+06	5.44E+06	3.64E+06	1.09E+08
F13	min	1.42E+03	1.45E+03	**1.39E+03**	3.13E+03	1.89E+03	2.60E+03	2.42E+03	2.18E+03
	std	9.08E+03	7.66E+03	9.52E+03	1.46E+04	1.08E+04	1.05E+04	1.34E+04	2.97E+05
	avg	1.16E+04	**7.56E+03**	8.64E+03	2.08E+04	1.41E+04	1.70E+04	1.86E+04	6.89E+04
F14	min	**1.42E+03**	1.47E+03	1.47E+03	1.49E+03	1.60E+03	1.53E+03	1.51E+03	1.45E+03
	std	1.16E+03	5.25E+03	4.73E+03	2.11E+03	6.99E+03	2.35E+03	1.10E+03	4.56E+01
	avg	2.31E+03	6.14E+03	4.92E+03	3.05E+03	8.10E+03	3.73E+03	2.37E+03	**1.53E+03**
F15	min	1.59E+03	1.57E+03	**1.51E+03**	1.83E+03	2.19E+03	1.63E+03	1.64E+03	1.82E+03
	std	5.85E+03	6.98E+03	6.76E+03	6.46E+03	3.08E+04	1.59E+04	4.72E+03	3.29E+03
	avg	**5.63E+03**	8.55E+03	6.92E+03	8.43E+03	4.04E+04	1.66E+04	8.87E+03	6.98E+03
F16	min	**1.60E+03**	1.60E+03	1.61E+03	1.65E+03	1.62E+03	1.69E+03	1.62E+03	1.70E+03
	std	1.51E+02	1.52E+02	1.67E+02	1.22E+02	2.68E+02	1.48E+02	1.66E+02	1.28E+02
	avg	**1.77E+03**	1.88E+03	1.81E+03	1.88E+03	2.21E+03	1.95E+03	1.96E+03	1.99E+03
F17	min	**1.71E+03**	1.72E+03	1.72E+03	1.75E+03	1.76E+03	1.75E+03	1.74E+03	1.76E+03
	std	2.87E+01	5.84E+01	5.54E+01	3.93E+01	2.26E+02	8.97E+01	7.58E+01	3.19E+01
	avg	**1.75E+03**	1.79E+03	1.79E+03	1.79E+03	2.03E+03	1.85E+03	1.82E+03	1.82E+03
F18	min	2.45E+03	**1.99E+03**	2.55E+03	4.83E+03	2.09E+03	3.74E+03	2.13E+03	2.29E+03
	std	7.92E+03	9.73E+03	1.20E+04	1.83E+04	1.25E+04	1.13E+04	1.38E+04	8.10E+05
	avg	**1.24E+04**	1.25E+04	1.92E+04	4.28E+04	1.31E+04	1.66E+04	1.74E+04	2.03E+05
F19	min	1.94E+03	**1.92E+03**	1.93E+03	2.08E+03	2.02E+03	2.38E+03	2.19E+03	1.94E+03
	std	5.98E+03	9.21E+03	9.21E+03	7.77E+04	1.27E+04	2.30E+05	2.13E+04	5.63E+03
	avg	7.77E+03	1.08E+04	8.96E+03	3.48E+04	1.40E+04	9.37E+04	1.99E+04	**6.44E+03**

Continued on next page

F		DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
F20	min	**2.00E+03**	2.04E+03	2.04E+03	2.05E+03	2.07E+03	2.06E+03	2.04E+03	2.07E+03
	std	6.08E+01	9.17E+01	6.33E+01	6.19E+01	1.43E+02	8.07E+01	7.67E+01	6.34E+01
	avg	**2.06E+03**	2.15E+03	2.12E+03	2.15E+03	2.27E+03	2.20E+03	2.20E+03	2.19E+03
F21	min	**2.01E+03**	2.21E+03	2.10E+03	2.22E+03	2.21E+03	2.23E+03	2.21E+03	2.24E+03
	std	5.08E+01	2.60E+01	6.93E+01	3.78E+01	5.72E+01	4.00E+01	6.34E+01	4.67E+01
	avg	**2.29E+03**	2.32E+03	2.31E+03	2.32E+03	2.36E+03	2.35E+03	2.33E+03	2.36E+03
F22	min	**2.22E+03**	2.24E+03	2.27E+03	2.24E+03	2.33E+03	2.29E+03	2.25E+03	2.43E+03
	std	2.86E+01	2.45E+02	2.60E+02	1.88E+01	6.32E+02	3.59E+02	2.80E+02	2.95E+02
	avg	**2.29E+03**	2.35E+03	2.37E+03	2.32E+03	2.80E+03	2.45E+03	2.37E+03	3.12E+03
F23	min	**2.61E+03**	2.61E+03	2.61E+03	2.62E+03	2.66E+03	2.62E+03	2.63E+03	2.64E+03
	std	1.02E+01	2.69E+01	1.42E+01	1.53E+01	8.31E+01	2.46E+01	3.06E+01	3.07E+01
	avg	**2.63E+03**	2.65E+03	2.64E+03	2.65E+03	2.79E+03	2.66E+03	2.69E+03	2.71E+03
F24	min	**2.50E+03**	2.50E+03	2.75E+03	2.51E+03	2.52E+03	2.75E+03	2.50E+03	2.79E+03
	std	7.92E+01	7.61E+01	1.94E+01	7.53E+01	1.15E+02	2.51E+01	1.06E+02	4.14E+01
	avg	**2.73E+03**	2.75E+03	2.78E+03	2.75E+03	2.90E+03	2.79E+03	2.81E+03	2.87E+03
F25	min	**2.90E+03**	2.90E+03	2.90E+03	2.90E+03	2.96E+03	2.95E+03	2.90E+03	3.00E+03
	std	2.40E+01	2.02E+01	2.65E+01	2.42E+01	2.68E+01	4.04E+01	5.24E+01	2.60E+02
	avg	**2.93E+03**	2.94E+03	2.93E+03	2.95E+03	2.99E+03	2.99E+03	2.96E+03	3.39E+03
F26	min	**2.60E+03**	2.60E+03	2.80E+03	2.71E+03	2.65E+03	2.79E+03	2.83E+03	3.70E+03
	std	1.36E+02	4.46E+02	3.90E+02	1.78E+02	6.99E+02	6.19E+02	5.96E+02	2.87E+02
	avg	**2.91E+03**	3.11E+03	3.24E+03	3.08E+03	3.94E+03	3.69E+03	3.72E+03	4.20E+03
F27	min	**3.09E+03**	3.09E+03	3.09E+03	3.10E+03	3.16E+03	3.10E+03	3.10E+03	3.11E+03
	std	3.20E+01	2.89E+01	3.52E+01	6.67E+00	9.44E+01	3.95E+01	5.69E+01	4.59E+01
	avg	3.12E+03	3.12E+03	3.13E+03	**3.11E+03**	3.27E+03	3.15E+03	3.19E+03	3.18E+03
F28	min	3.10E+03	**2.80E+03**	3.10E+03	3.19E+03	3.12E+03	3.07E+03	3.11E+03	3.44E+03
	std	1.28E+02	1.75E+02	1.97E+02	8.29E+01	1.52E+02	1.84E+02	1.73E+02	1.26E+02
	avg	3.32E+03	**3.29E+03**	3.35E+03	3.49E+03	3.55E+03	3.46E+03	3.43E+03	3.74E+03
F29	min	**3.16E+03**	3.16E+03	3.17E+03	3.17E+03	3.17E+03	3.17E+03	3.23E+03	3.24E+03
	std	5.99E+01	6.93E+01	1.01E+02	6.82E+01	1.85E+02	1.34E+02	1.08E+02	1.01E+02
	avg	**3.24E+03**	3.29E+03	3.32E+03	3.29E+03	3.56E+03	3.40E+03	3.37E+03	3.42E+03
F30	min	1.01E+04	4.69E+03	**7.67E+03**	4.08E+03	6.39E+03	3.96E+04	5.03E+04	1.84E+05
	std	5.59E+05	8.16E+05	5.00E+05	1.47E+06	1.21E+07	2.64E+06	3.17E+06	4.72E+06
	avg	**4.45E+05**	6.96E+05	5.01E+05	1.57E+06	8.15E+06	2.46E+06	2.08E+06	4.49E+06

Box plots (Figure 5) represent the stability of the algorithms visually. For functions F1, F3–10, F12–F13, F15–18, F20–F21, F23–26, and F29–F30, the DMOMSO algorithm had a more concentrated data distribution, with its box plot appearing shorter. This means that its solutions fluctuated less and were more stable. Some other algorithms had more dispersed data; hence, their stability was poor in comparison. This indicates that the DMOMSO algorithm's results were very consistent over multiple runs and were less affected by random factors.

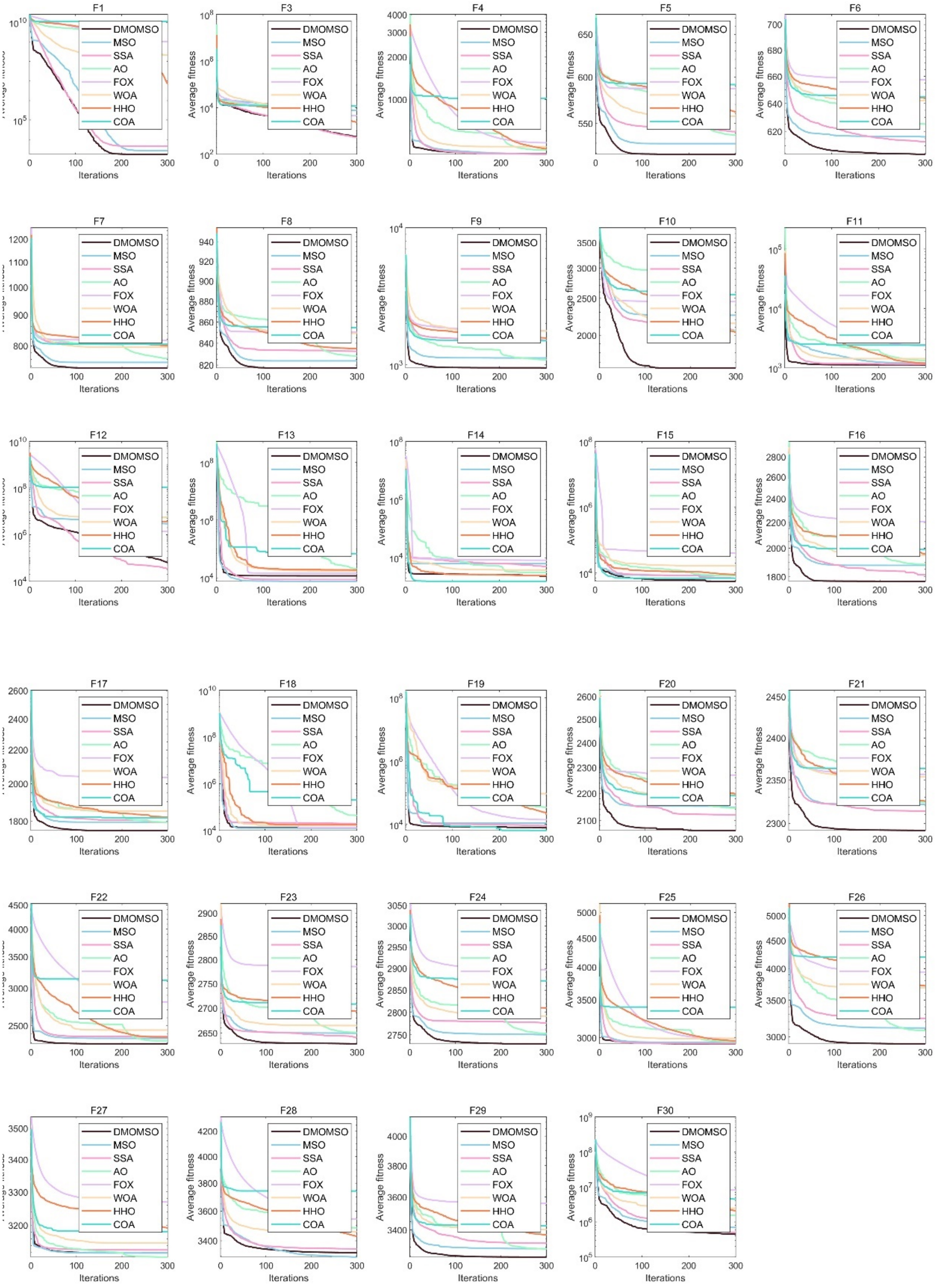


Figure 4. The average convergence curves for CEC2017.

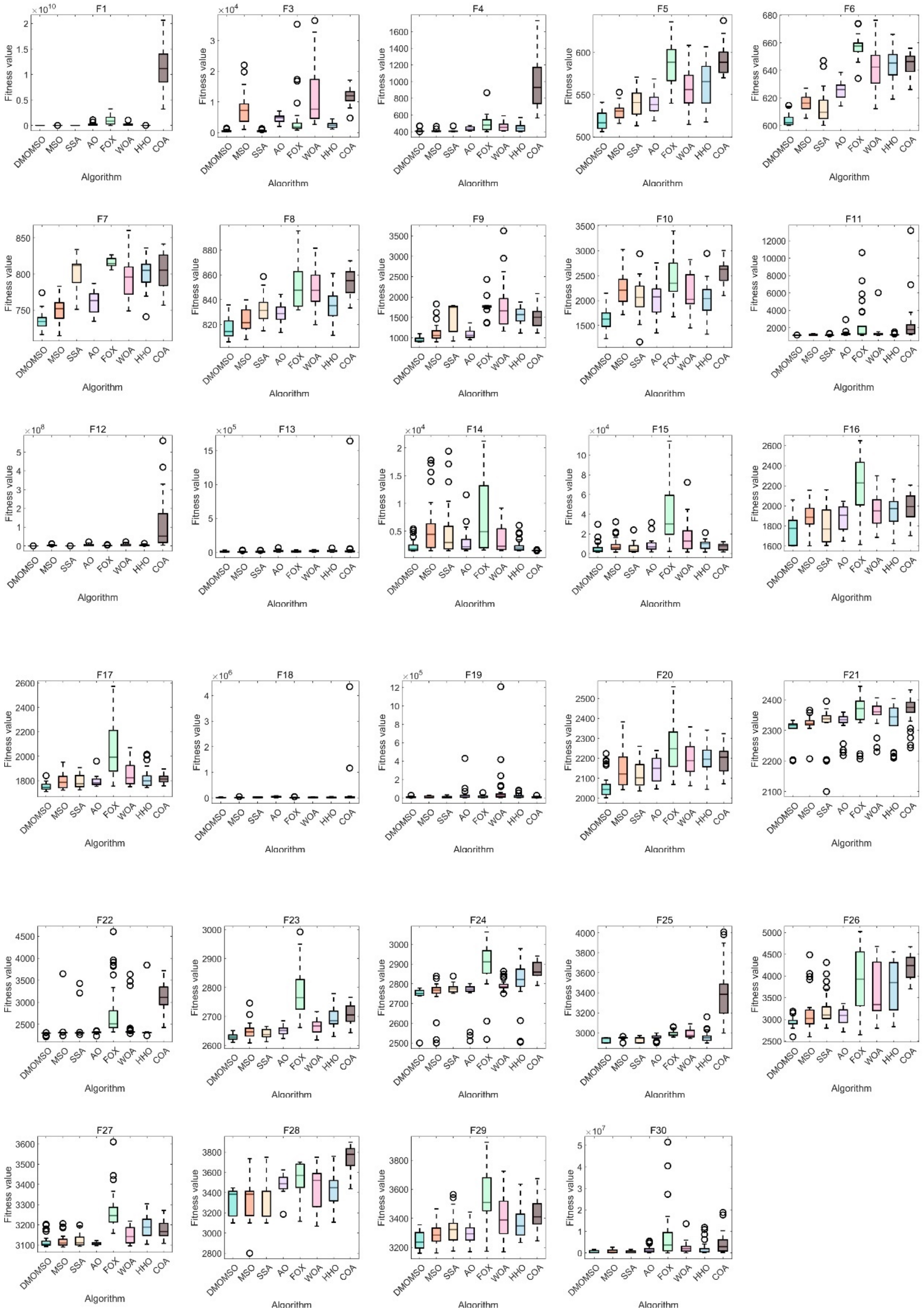


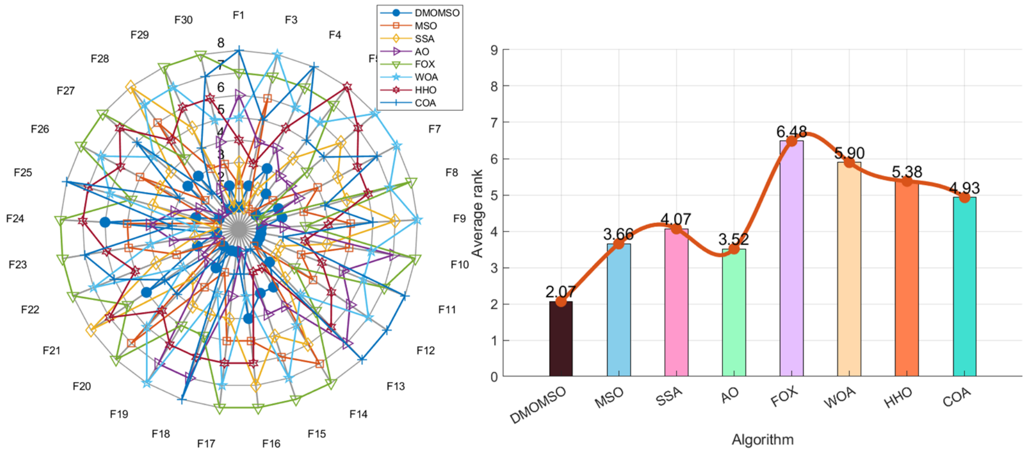
Figure 5. The box charts for CEC2017.

The Wilcoxon test (Table 3) showed significant algorithm differences (considering $p < 0.05$ to indicate statistically significant differences). For example, for F4, the p-value of the comparison between DMOMSO and MSO was 1.68E-04, that between DMOMSO and SSA was 7.04E-07, and that between DMOMSO and AO was 4.11E-07. For most functions, the p-values of DMOMSO in comparison to other algorithms were small. This indicates statistically significant performance differences, further proving the DMOMSO algorithm's effectiveness and advantages in the context of function optimization.

Table 3. Results of Wilcoxon test for CEC2017.

F	MSO	SSA	AO	FOX	WOA	HHO	COA
F1	6.10E-01	1.24E-03	3.02E-11	1.96E-10	3.02E-11	3.02E-11	3.02E-11
F3	4.08E-11	8.07E-01	3.02E-11	1.33E-10	3.02E-11	5.49E-11	3.02E-11
F4	8.07E-01	7.84E-01	3.65E-08	1.49E-06	5.00E-09	3.16E-05	3.02E-11
F5	1.68E-04	7.04E-07	4.11E-07	3.34E-11	4.18E-09	1.55E-09	3.02E-11
F6	5.97E-09	1.68E-04	3.34E-11	3.02E-11	4.50E-11	3.02E-11	3.02E-11
F7	1.58E-04	5.49E-11	3.65E-08	3.02E-11	8.15E-11	6.70E-11	4.98E-11
F8	6.97E-03	6.05E-07	7.74E-06	1.09E-10	3.47E-10	3.26E-07	3.34E-11
F9	3.32E-06	1.41E-09	9.06E-08	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F10	7.12E-09	2.32E-06	1.25E-05	9.76E-10	4.44E-07	3.09E-06	3.69E-11
F11	4.64E-05	1.30E-03	5.49E-11	1.21E-10	1.09E-10	2.44E-09	3.02E-11
F12	9.92E-11	1.70E-02	1.61E-10	1.10E-08	1.96E-10	1.41E-09	3.02E-11
F13	3.64E-02	4.51E-02	6.38E-03	4.20E-01	3.03E-02	3.39E-02	1.86E-01
F14	1.78E-04	4.23E-03	3.92E-02	1.64E-05	8.68E-03	2.97E-01	7.70E-04
F15	2.15E-02	4.73E-01	9.47E-03	2.02E-08	2.53E-04	1.37E-03	1.33E-02
F16	9.07E-03	1.91E-01	2.50E-03	2.02E-08	1.68E-04	4.94E-05	2.88E-06
F17	3.18E-03	2.38E-03	1.02E-05	2.67E-09	1.47E-07	7.74E-06	1.10E-08
F18	6.63E-01	3.92E-02	8.48E-09	5.20E-01	2.01E-01	3.87E-01	4.21E-02
F19	6.31E-01	5.40E-01	1.68E-03	6.15E-02	3.18E-04	1.22E-02	2.58E-01
F20	1.34E-05	1.68E-04	2.00E-06	4.18E-09	4.69E-08	7.69E-08	2.02E-08
F21	7.29E-03	5.61E-05	4.12E-06	3.35E-08	4.31E-08	8.15E-05	9.53E-07
F22	1.02E-05	2.15E-02	4.62E-10	3.02E-11	3.16E-10	1.29E-09	3.02E-11
F23	1.17E-04	8.56E-04	1.60E-07	3.02E-11	6.53E-08	1.21E-10	4.08E-11
F24	5.57E-03	1.53E-05	1.89E-04	5.00E-09	3.20E-09	3.81E-07	3.02E-11
F25	1.11E-03	3.26E-01	3.57E-06	3.02E-11	1.21E-10	5.87E-04	3.02E-11
F26	5.83E-03	5.86E-06	1.17E-04	8.48E-09	3.50E-09	5.09E-06	3.02E-11
F27	5.69E-01	2.28E-01	8.53E-01	1.96E-10	1.37E-03	3.81E-07	2.38E-07
F28	6.63E-01	6.52E-01	4.18E-09	6.01E-08	4.22E-04	2.50E-03	4.50E-11
F29	8.31E-03	1.17E-03	1.03E-02	1.17E-09	1.11E-06	1.11E-06	1.69E-09
F30	7.48E-02	8.65E-01	2.84E-04	5.60E-07	8.20E-07	3.56E-04	1.87E-07

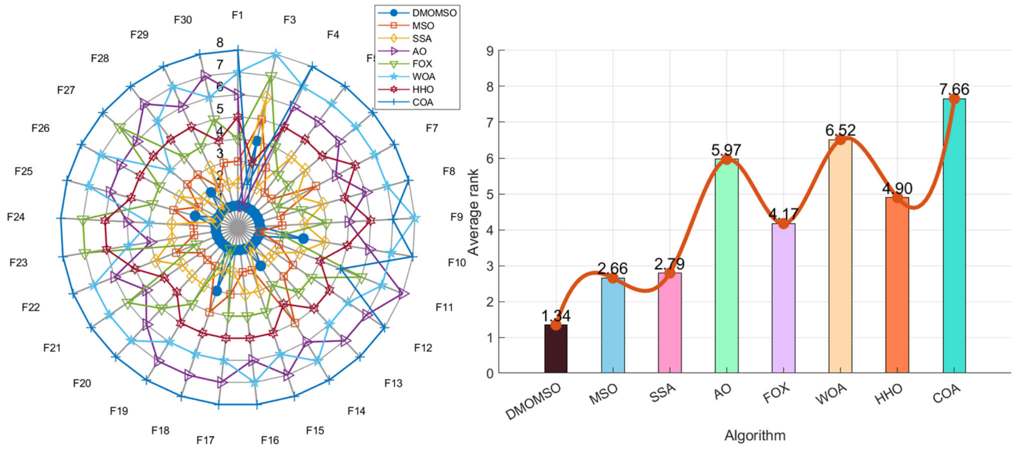
In Figure 6(a), for each dimension of the test functions, the closer the radar chart line was to the center, the better the algorithm performed with respect to that function. The DMOMSO algorithm (the blue line) was close to the center in many function dimensions, which means that it ranked high in accuracy for many functions. Specifically, DMOMSO ranked first for 12 functions, second for 9, and third for 5. Figure 6(b) presents a ranking chart showing the average fitness values of the different algorithms. From this figure, we can see that the DMOMSO algorithm delivered the optimal average results on 29 functions and, thus, was ranked first. This demonstrates that it is good at finding optimal solutions for these functions and can obtain good results in general. Its overall shape is quite distinct, indicating its excellent comprehensive performance and its ability to handle different types of function optimization tasks.



(a) (b)

Figure 6. Radar chart (a) and ranking chart (b) for CEC2017.

To verify that DMOMSO excels at handling high-dimensional complex problems, we set the dimension to 100. Moreover, the population size was configured as 70, and the number of iterations was set to 1000. Test results of high-dimensional functions are shown in Figure 7, where DMOMSO exhibits advantages.



(a) (b)

Figure 7. Radar chart (a) and ranking chart (b) for CEC2017(70-1000-100).

In Figure 7(a), the radar chart indicates that for each dimension of the test functions, an algorithm performed better on that function when its corresponding line was closer to the center. It was obvious that the DMOMSO algorithm was close to the center, which means it achieved a high accuracy ranking in numerous cases. Specifically, DMOMSO ranked first in 23 functions and second in 3 functions.

Figure 7(b) presents a ranking chart based on the average fitness values of different algorithms.

As shown in this chart, DMOMSO attained the optimal average results in 29 functions, thus securing the top position. This confirms its excellent performance in terms of scalability and robustness.

Table 4. Comparative data for CEC2022.

F		DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
	min	**3.05E+02**	7.19E+02	3.36E+02	2.05E+03	3.51E+03	6.42E+03	7.10E+02	3.47E+03
F1	std	6.97E+02	4.13E+03	1.34E+03	2.79E+03	7.38E+03	1.42E+04	1.01E+03	2.10E+03
	avg	**8.65E+02**	6.01E+03	1.68E+03	5.90E+03	1.23E+04	2.69E+04	3.11E+03	7.87E+03
	min	**4.00E+02**	4.00E+02	4.00E+02	4.11E+02	4.00E+02	4.06E+02	4.00E+02	5.27E+02
F2	std	2.34E+01	2.85E+01	2.60E+01	2.25E+01	1.31E+02	9.06E+01	7.29E+01	6.62E+02
	avg	**4.12E+02**	4.21E+02	4.15E+02	4.52E+02	5.62E+02	4.75E+02	4.83E+02	1.28E+03
	min	**6.00E+02**	6.06E+02	6.07E+02	6.11E+02	6.39E+02	6.25E+02	6.39E+02	6.21E+02
F3	std	4.62E+00	9.92E+00	1.02E+01	7.01E+00	1.40E+01	1.08E+01	9.11E+00	1.07E+01
	avg	**6.05E+02**	6.20E+02	6.29E+02	6.24E+02	6.63E+02	6.42E+02	6.62E+02	6.46E+02
	min	**8.06E+02**	8.10E+02	8.08E+02	8.07E+02	8.53E+02	8.13E+02	8.41E+02	8.32E+02
F4	std	8.72E+00	9.06E+00	9.26E+00	7.51E+00	1.26E+01	7.46E+00	7.16E+00	9.37E+00
	avg	**8.20E+02**	8.21E+02	8.23E+02	8.22E+02	8.79E+02	8.26E+02	8.55E+02	8.53E+02
	min	**9.00E+02**	9.07E+02	9.28E+02	9.28E+02	9.19E+02	1.11E+03	1.36E+03	9.76E+02
F5	std	5.66E+01	3.33E+02	1.34E+02	1.18E+02	3.99E+01	1.82E+02	1.50E+02	1.95E+02
	avg	**9.53E+02**	1.15E+03	1.14E+03	1.06E+03	9.57E+02	1.44E+03	1.77E+03	1.41E+03
	min	1.84E+03	**1.83E+03**	1.84E+03	4.49E+03	1.87E+03	2.05E+03	2.47E+08	3.16E+03
F6	std	1.63E+03	1.99E+03	1.84E+03	8.72E+04	2.32E+03	7.38E+03	7.27E+08	6.50E+06
	avg	**3.47E+03**	3.88E+03	3.39E+03	7.01E+04	4.55E+03	1.13E+04	1.07E+09	5.26E+06
	min	**2.00E+03**	2.02E+03	2.02E+03	2.03E+03	2.06E+03	2.03E+03	2.07E+03	2.05E+03
F7	std	1.91E+01	2.27E+01	1.75E+01	1.75E+01	3.81E+01	3.18E+01	2.24E+01	1.91E+01
	avg	**2.03E+03**	2.04E+03	2.05E+03	2.06E+03	2.11E+03	2.08E+03	2.12E+03	2.08E+03
	min	**2.21E+03**	2.21E+03	2.21E+03	2.23E+03	2.23E+03	2.23E+03	2.24E+03	2.22E+03
F8	std	2.71E+00	3.14E+01	2.33E+01	4.30E+00	4.35E+01	1.50E+01	1.76E+01	1.04E+01
	avg	**2.22E+03**	2.23E+03	2.23E+03	2.23E+03	2.27E+03	2.24E+03	2.27E+03	2.24E+03
	min	**2.53E+03**	2.53E+03	2.53E+03	2.58E+03	2.62E+03	2.59E+03	2.72E+03	2.68E+03
F9	std	1.47E+00	4.45E+01	6.31E+01	2.88E+01	5.00E+01	4.09E+01	5.21E+01	3.10E+01
	avg	**2.53E+03**	2.56E+03	2.57E+03	2.63E+03	2.70E+03	2.65E+03	2.79E+03	2.74E+03
	min	**2.50E+03**	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.53E+03	2.51E+03
F10	std	5.58E+01	6.40E+01	7.19E+01	6.23E+01	2.79E+02	1.50E+02	1.55E+02	2.37E+02
	avg	**2.54E+03**	2.56E+03	2.57E+03	2.58E+03	2.61E+03	2.64E+03	2.73E+03	2.80E+03
	min	2.90E+03	2.75E+03	**2.60E+03**	2.72E+03	2.83E+03	2.62E+03	2.88E+03	3.02E+03
F11	std	2.23E+02	5.74E+02	3.32E+02	1.20E+02	4.99E+02	1.52E+02	4.61E+02	4.51E+02
	avg	2.94E+03	3.04E+03	2.84E+03	**2.81E+03**	3.68E+03	2.86E+03	3.69E+03	3.78E+03
	min	**2.86E+03**	2.87E+03	2.86E+03	2.86E+03	2.89E+03	2.86E+03	2.89E+03	2.89E+03
F12	std	9.29E+00	1.85E+01	2.38E+01	9.95E+00	4.69E+01	5.46E+01	8.92E+01	5.12E+01
	avg	**2.87E+03**	2.89E+03	2.87E+03	2.87E+03	2.98E+03	2.94E+03	3.01E+03	2.96E+03

4.2. Experiments on CEC2022

As shown in Table 4, the DMOMSO algorithm performed differently across the functions in the CEC2022 benchmark function set. For Functions F1–5, F7–10, and F12, its minimum value was smaller than those obtained by the remaining seven algorithms. This demonstrates that it has an advantage in finding the optimal solution. Its smaller standard deviation means that its solutions fluctuated less, reflecting good stability. For these functions, the DMOMSO algorithm also presented the lowest average value, again outperforming the other algorithms.

From the convergence curves shown in Figure 8, the DMOMSO algorithm dropped fast and reached a stable state early. This demonstrates its swift convergence and efficient solution-finding ability and is particularly obvious for functions such as F1–7, F10, and F12.

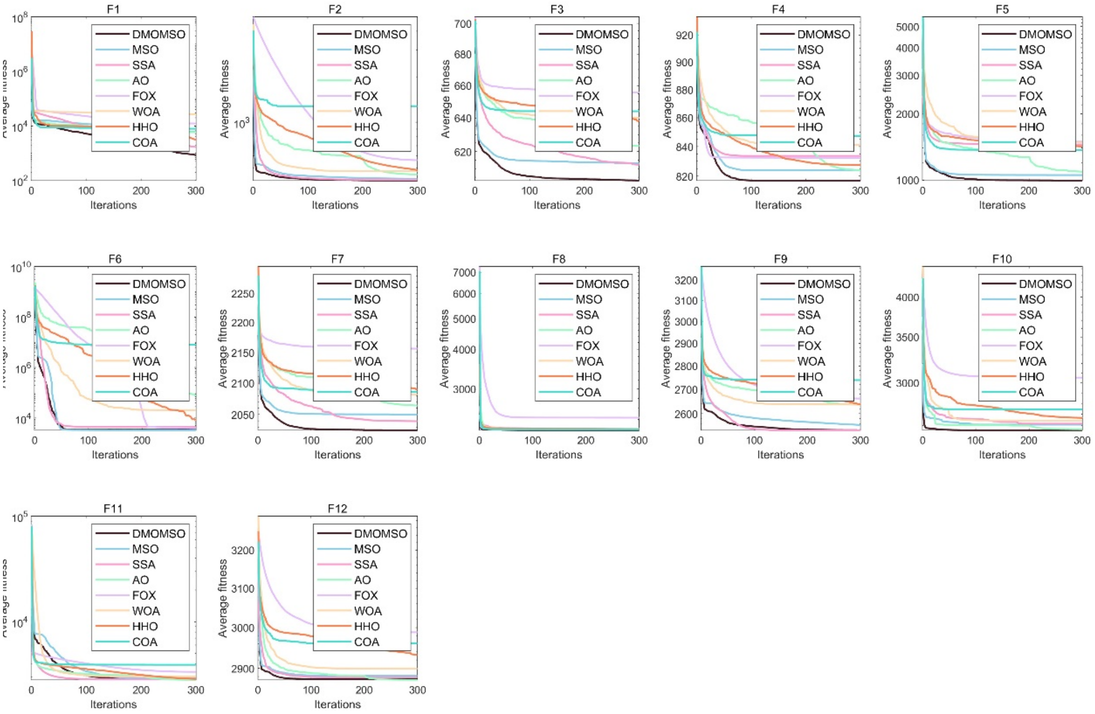


Figure 8. The average convergence curves for CEC2022.

The box plots in Figure 9 show the data's dispersion and stability. In the box plot for the DMOMSO algorithm, the data are concentrated, and the box is short, meaning that the solutions fluctuated little and were stable, especially for functions F1–5, F7–F8, and F12. For some functions, such as F1–F5, the DMOMSO algorithm's solutions were highly concentrated. This indicates that the algorithm's results were very consistent over multiple runs of these functions and were less affected by random factors.

Similar to the above, in the CEC2022 Wilcoxon test (Table 5), the DMOMSO algorithm was considered to have outperformed the other algorithms significantly when the p-value was less than 0.05. For Function F1, when compared with AO, WOA, HHO, and COA, every p-value was under 0.05, indicating that the DMOMSO algorithm had a comparative performance advantage on F1. For functions such as F3–5 and F7–12, in most cases, the p-values when comparing the DMOMSO algorithm with other algorithms were below 0.05, further proving its effectiveness and advantages with respect to these functions.

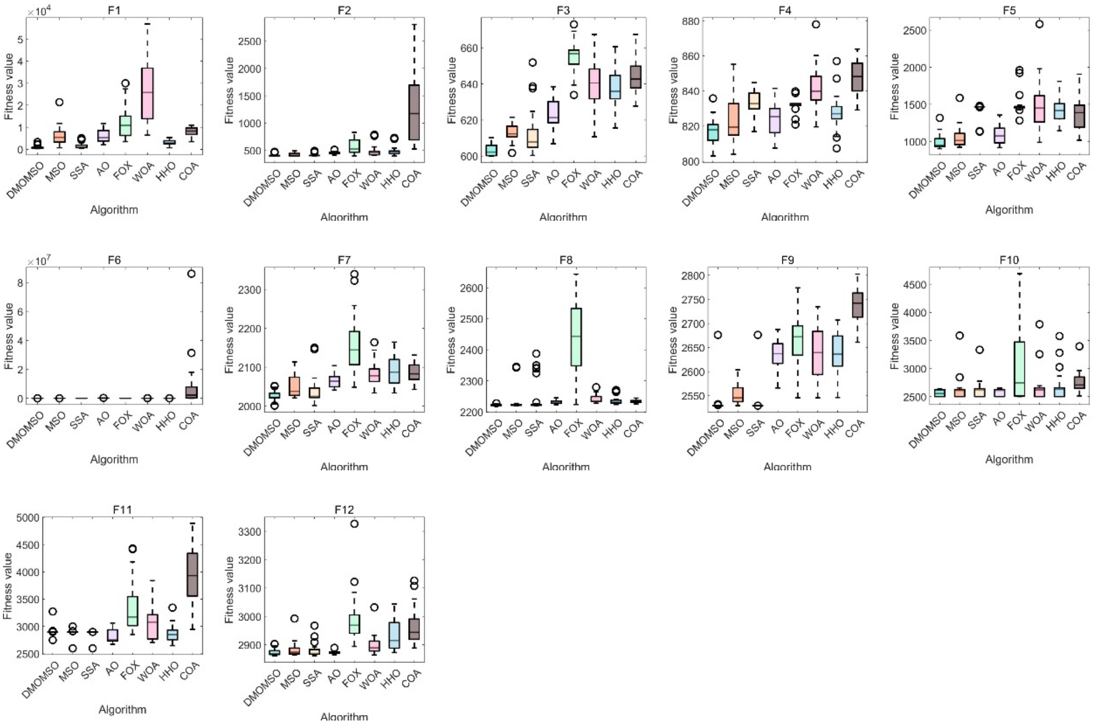
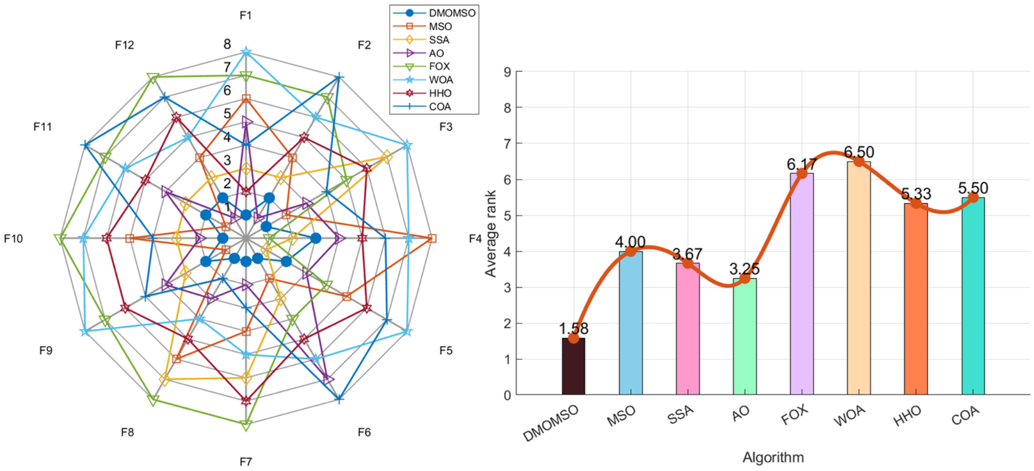


Figure 9. The box charts for CEC2022.

Table 5. Results of the Wilcoxon test for CEC2022.

F	MSO	SSA	AO	FOX	WOA	HHO	COA
F1	7.38E-10	1.44E-03	7.39E-11	3.02E-11	3.02E-11	8.89E-10	3.02E-11
F2	1.37E-01	9.05E-02	3.01E-07	1.87E-07	3.96E-08	3.08E-08	3.02E-11
F3	1.17E-09	1.64E-05	4.98E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F4	9.33E-02	1.29E-09	2.16E-03	8.89E-10	4.20E-10	1.34E-05	4.50E-11
F5	1.44E-02	5.49E-11	8.56E-04	3.34E-11	1.55E-09	8.15E-11	4.62E-10
F6	6.41E-01	5.75E-02	3.69E-11	5.79E-01	3.35E-08	7.74E-06	1.46E-10
F7	1.58E-04	5.94E-02	3.82E-10	3.69E-11	2.15E-10	2.87E-10	5.49E-11
F8	4.06E-02	3.03E-02	1.96E-10	3.69E-11	3.34E-11	6.70E-11	4.50E-11
F9	4.57E-09	2.23E-09	3.82E-10	1.33E-10	2.15E-10	3.47E-10	3.69E-11
F10	8.77E-02	2.07E-02	2.71E-02	8.66E-05	7.96E-03	5.56E-04	3.52E-07
F11	7.04E-07	7.70E-04	2.50E-03	2.92E-09	9.63E-02	9.71E-01	5.49E-11
F12	6.57E-02	6.52E-01	6.35E-02	4.08E-11	5.09E-06	2.60E-08	8.99E-11

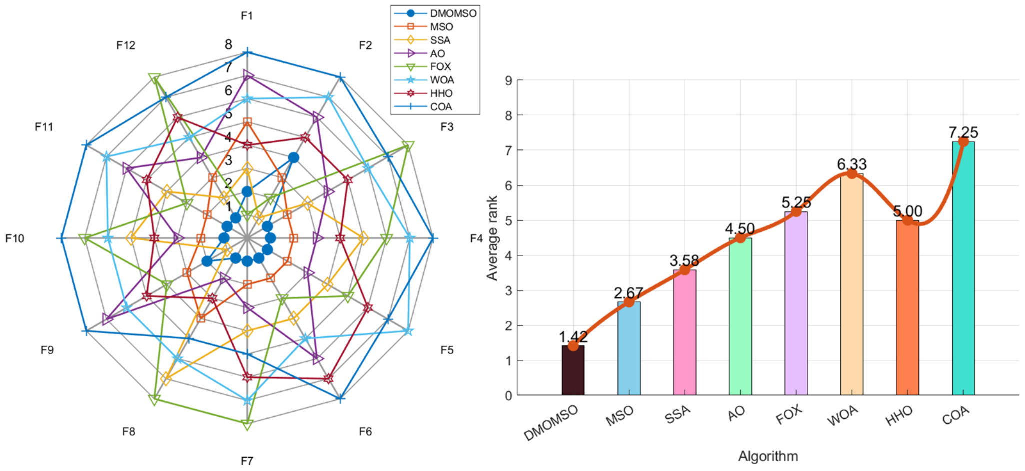
As seen from Figure 10(a), the DMOMSO algorithm (the blue line) was close to the center in many function dimensions, indicating that it ranked highly with respect to many functions. Specifically, it ranked first on 6 functions, second on 5 functions, and third on 1 function. From Figure 10(b), we can see that the DMOMSO algorithm had the best average performance on 12 functions and ranked first overall. This demonstrates its improved ability to find optimal solutions in these function optimizations, thus obtaining good results. Its overall shape is distinct, showing excellent comprehensive performance and an ability to deal with different types of function optimization tasks.



(a) (b)

Figure 10. Radar chart (a) and ranking chart (b) for CEC2022.

To confirm DMOMSO's superiority in tackling high-dimensional complex problems, the dimension was configured to 20. Moreover, the population was set at 70, and the number of iterations was set at 1000. Test results for high-dimensional functions are presented in Figure 11, and the DMOMSO shows obvious strengths.



(a) (b)

Figure 11. Radar chart (a) and ranking chart (b) for CEC2022(70-1000-20).

In Figure 11(a), the radar chart illustrates that for each dimension of the test functions, an algorithm performs better on that function when its corresponding line is closer to the center. It is evident that the DMOMSO algorithm lies near the center across function dimensions, reflecting its high accuracy ranking in many cases. Specifically, DMOMSO secured first place in 9 functions and second in 2. Figure 11(b) displays a ranking chart based on the average fitness values of various algorithms. From this chart, the DMOMSO algorithm achieved the optimal average results in 12 functions, thus claiming the top position. This confirms its excellent performance in terms of scalability and robustness.

5. Engineering applications

5.1. Design problem of hydrostatic thrust bearing

In hydrostatic thrust bearing design [31], the designer aims to minimize the power loss. This problem incorporates three parameters: The bearing radius R, flow rate Q, and recess radius R_0. The design must satisfy 7 non-linear constraints, relating to the inlet oil pressure, load capacity, and oil film thickness. The problem can be formulated as follows:

Minimize:

$$f(\bar{x}) = \frac{QP_0}{0.7} + E_f \tag{22}$$

Subject to:

$$g_1(\bar{x}) = 1000 - P_0 \le 0, \tag{23}$$

$$g_2(\bar{x}) = W - 101000 \le 0, \tag{24}$$

$$g_3(\bar{x}) = 5000 - \frac{W}{\pi(R^2 - R_0^2)} \le 0, \tag{25}$$

$$g_4(\bar{x}) = 50 - P_0 \le 0, \tag{26}$$

$$g_5(\bar{x}) = 0.001 - \frac{0.0307}{386.4P_0}\left(\frac{Q}{2\pi Rh}\right) \le 0, \tag{27}$$

$$g_6(\bar{x}) = R - R_0 \le 0, \tag{28}$$

$$g_7(\bar{x}) = h - 0.001 \le 0, \tag{29}$$

where:

$$W = \frac{\pi P_0}{2} \frac{R^2 - R_0^2}{ln\left(\frac{R}{R_0}\right)}, P_0 = \frac{6\mu Q}{\pi h^3} ln\left(\frac{R}{R_0}\right), \tag{30}$$

$$E_f = 9336Q \times 0.0307 \times 0.5\Delta T, \Delta T = 2(10^P - 559.7), \tag{31}$$

$$P = \frac{log_{10} log_{10}\left(8.122 \times 10^6 \mu + 0.8\right) + 3.55}{10.04}, \tag{32}$$

$$h = \left(\frac{2\pi \times 750}{60}\right)^2 \frac{2\pi\mu}{E_f} \left(\frac{R^4}{4} - \frac{R_0^4}{4}\right) \tag{33}$$

With bounds:

$$1 \le R \le 16, 1 \le R_0 \le 16, \tag{34}$$

$$1 \times 10^{-6} \le \mu \le 16 \times 10^{-6}, 1 \le Q \le 16. \tag{35}$$

The constraint violations are incorporated as a penalty term into the objective function to construct the fitness function:

$$F(\overline{x}) = f(\overline{x}) + \sum_{i=1}^{7} \lambda_i \cdot \max\left(0, g_i(\overline{x})\right)^2 \tag{36}$$

where λ_i is the penalty coefficient for the i th constraint (requires tuning, typically set to a large value such as 10^6). $\max\left(0, g_i(\overline{x})\right)$ represents the constraint violation (if $g_i(\overline{x}) \le 0$, the constraint is satisfied).

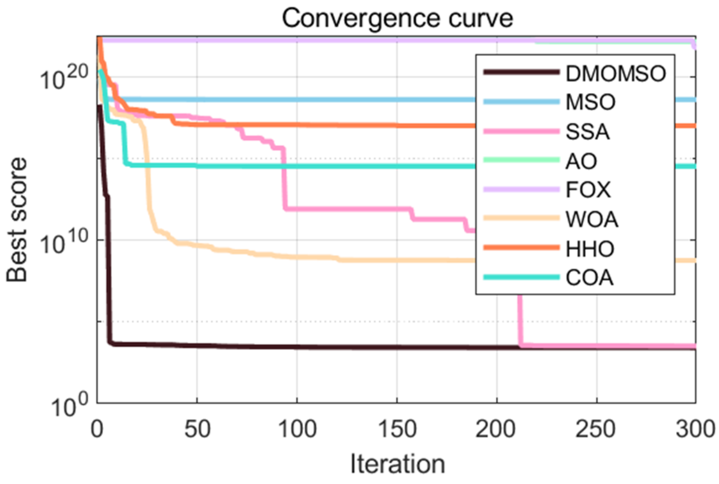


Figure 12. Convergence curves for the hydrostatic thrust bearing design test.

Table 6. Test results for Hydrostatic Thrust Bearing design optimization.

Algorithms	DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
min	2.26E+03	3.70E+03	2.26E+03	2.24E+15	3.06E+18	3.94E+03	4.16E+03	3.02E+03
std	3.42E+02	1.21E+19	1.38E+03	4.06E+22	9.95E+21	7.83E+08	2.13E+17	1.03E+15
avg	**2.65E+03**	3.84E+18	3.36E+03	1.35E+22	5.81E+21	5.65E+08	9.78E+16	3.24E+14

The convergence curve graph (Figure 12) shows how the 8 algorithms (DMOMSO, MSO, SSA, AO, FOX, WOA, HHO, COA) performed when applied to this hydrostatic thrust bearing design problem. For the DMOMSO algorithm, its curve dropped quickly and became stable after only a few iterations. This means that it converged extremely rapidly and reached the optimal solution, which is a significant advantage for improving design efficiency. As shown in Table 6, it also converged to a lower optimal value, demonstrating that it can find a high-quality design scheme with great precision when applied to the hydrostatic thrust bearing design optimization problem. Therefore, the DMOMSO algorithm is considered more suitable for solving this design problem. The other algorithms presented drawbacks in different ways and need to be improved.

5.2. Optimization of the Himmel Blau's function

Himmel Blau's function [32] is often applied to evaluate optimization algorithms; in particular, it is mainly used when analyzing optimization tasks restricted by non-linear conditions. This function contains 6 non-linear constraints and 5 parameters. The problem formulation is provided below:

Minimize

$$f\left(\vec{x}\right)=5.3578547x_3^2+0.8356891x_1x_5+37.293239x_1-40792.141 \tag{37}$$

subject to the constraints

$$\begin{aligned}
&g_1\left(\vec{x}\right)=-G1\leq 0,\\
&g_2\left(\vec{x}\right)=G1-92\leq 0,\\
&g_3\left(\vec{x}\right)=90-G2\leq 0,\\
&g_4\left(\vec{x}\right)=G2-110\leq 0,\\
&g_5\left(\vec{x}\right)=20-G3\leq 0,\\
&g_6\left(\vec{x}\right)=G3-25\leq 0,
\end{aligned} \tag{38}$$

where

$$\begin{aligned}
&G1=85.334407+0.0056858x_2x_5+0.0006262x_1x_4-0.0022053x_3x_5,\\
&G2=80.51249+0.0071317x_2x_5+0.0029955x_1x_2+0.0021813x_3^2, \quad \cdot\\
&G3=9.300961+0.0047026x_3x_5+0.00125447x_1x_3+0.0019085x_3x_4,
\end{aligned} \tag{39}$$

The values of the variables are constrained to the following ranges:

$$\begin{aligned} &78 \le x_1 \le 102, \\ &33 \le x_2 \le 45, \\ &27 \le x_3 \le 45, \\ &27 \le x_4 \le 45, \\ &27 \le x_5 \le 45. \end{aligned} \tag{40}$$

The convergence curves in Figure 13 present the performances of the 8 algorithms (DMOMSO, MSO, SSA, AO, FOX, WOA, HHO, COA) when optimizing Himmel Blau's function. The DMOMSO algorithm curve shows the steepest drop at the beginning of the iteration process. This demonstrates that it converged much faster than other algorithms and reached the optimal solution in fewer steps, saving optimization time. Table 7 demonstrates the DMOMSO algorithm's superior convergence close to an optimal value. This means that it can find a better combination of design parameters with high precision, thus improving its optimization performance.

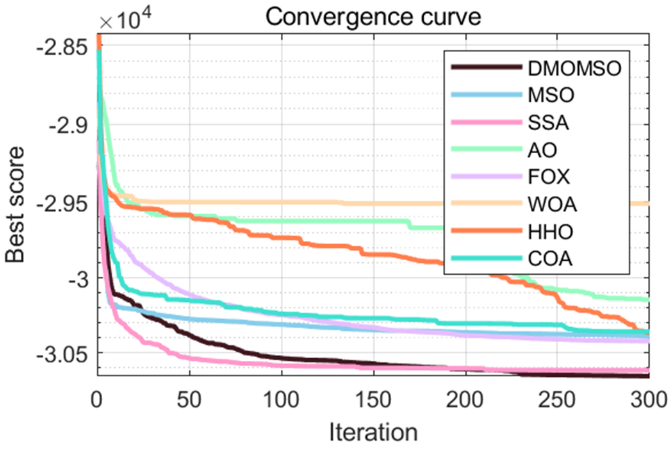


Figure 13. Convergence curves for Himmel Blau's function test.

Table 7. Test results for Himmel Blau's function optimization.

Algorithms	DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
min	-3.07E+04	-3.06E+04	-3.07E+04	-3.05E+04	-3.07E+04	-3.04E+04	-3.07E+04	-3.07E+04
std	1.39E+01	2.01E+02	1.52E+02	2.71E+02	2.17E+02	5.36E+02	1.72E+02	2.04E+02
avg	**-3.07E+04**	-3.04E+04	-3.06E+04	-3.01E+04	-3.04E+04	-2.95E+04	-3.04E+04	-3.04E+04

Overall, among the 8 considered algorithms, the DMOMSO algorithm excelled in terms of convergence speed, accuracy, and reliability. It had an overall advantage in optimizing Himmel Blau's function. The other algorithms were found to have flaws and may need further optimization for practical use.

5.3. Optimization of industrial refrigeration systems

Due to the overuse of basic energy resources, energy efficiency and emission reductions are now vital in every industry. As the industrial refrigeration setups of businesses use a great deal of energy, optimizing their design is key to properly harmonizing their performance, cost, and effectiveness. This design problem [33] includes 14 parameters and 15 constraints. The formulation of the problem is provided below:

Design variables:

$$\vec{x} = \left[x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14}\right] \tag{41}$$

Objective function:

$$\begin{aligned} f\left(\vec{x}\right) &= 63098.88 x_2 x_4 x_{12} + 5441.5 x_2^2 x_{12} + 115055.5 x_2^{1.664} x_6 + 6172.27 x_2^2 x_6 \\ &+ 63098.88 x_1 x_3 x_{11} + 5441.5 x_1^2 x_{11} + 115055.5 x_1^{1.664} x_5 + 6171.27 x_1^2 x_5 \\ &+ 140.53 x_1 x_{11} + 281.29 x_3 x_{11} + 70.26 x_1^2 + 281.29 x_1 x_3 + 281.29 x_3^2 \\ &+ 14437 x_8^{1.8812} x_{12}^{0.3424} x_{10} x_{14}^{-1} x_7 x_9^{-1} + 20470 x_7^{2.893} x_{11}^{0.316} x_1^2 \end{aligned} \tag{42}$$

Constraint conditions:

$$\begin{aligned} &g_1\left(\vec{x}\right) = 1.524 x_7^{-1} \le 1 \\ &g_2\left(\vec{x}\right) = 1.524 x_8^{-1} \le 1 \\ &g_3\left(\vec{x}\right) = 0.07789 x_1 - 2 x_7^{-3} x_9 - 1 \le 0 \\ &g_4\left(\vec{x}\right) = 7.05305 x_9^{-1} x_1^2 x_{10} x_8^{-1} x_2^{-1} x_{14}^{-1} - 1 \le 0 \\ &g_5\left(\vec{x}\right) = 0.0833 x_{13}^{-1} x_{14} - 1 \le 0 \\ &g_6\left(\vec{x}\right) = 47.136 x_2^{0.333} x_{10}^{-1} x_{12} - 1.333 x_8 x_{13}^{2.1195} + 62.08 x_{13}^{2.1195} x_{12}^{-1} x_8^{0.2} x_{10}^{-1} - 1 \le 0 \\ &g_7\left(\vec{x}\right) = 0.04771 x_{10} x_8^{1.8812} x_{12}^{0.3424} - 1 \le 0 \\ &g_8\left(\vec{x}\right) = 0.0488 x_9 x_7^{1.893} x_{11}^{0.316} - 1 \le 0 \\ &g_9\left(\vec{x}\right) = 0.0099 x_1 x_3^{-1} - 1 \le 0 \\ &g_{10}\left(\vec{x}\right) = 0.0193 x_2 x_4^{-1} - 1 \le 0 \\ &g_{11}\left(\vec{x}\right) = 0.0298 x_1 x_5^{-1} - 1 \le 0 \\ &g_{12}\left(\vec{x}\right) = 0.056 x_2 x_6^{-1} - 1 \le 0 \\ &g_{13}\left(\vec{x}\right) = 2 x_9^{-1} - 1 \le 0 \\ &g_{14}\left(\vec{x}\right) = 2 x_{10}^{-1} - 1 \le 0 \\ &g_{15}\left(\vec{x}\right) = x_{12} x_{11}^{-1} - 1 \le 0 \end{aligned} \tag{43}$$

Value range:

$$0.001 \le x_i \le 5, i = 1, \ldots, 14 \tag{44}$$

The convergence curves shown in Figure 14 indicate how the 8 algorithms (DMOMSO, MSO, SSA, AO, FOX, WOA, HHO, COA) performed in the industrial refrigeration system design test. The DMOMSO algorithm's curve showed the steepest downward slope at the beginning of the iteration process, which means that it converged much faster than others and reached the optimal solution quickly, reducing the optimization time. Meanwhile, Table 8 shows that the algorithm converged to a relatively low optimal value, indicating that it can find better design parameters with high precision, thereby improving the performance of the refrigeration system. Overall, the DMOMSO algorithm performed very well in the aspects of convergence rate, exactness, and consistency, presenting significant advantages in terms of optimizing industrial refrigeration system designs. The other algorithms presented certain weaknesses and may need further improvement for practical applications such as those considered here.

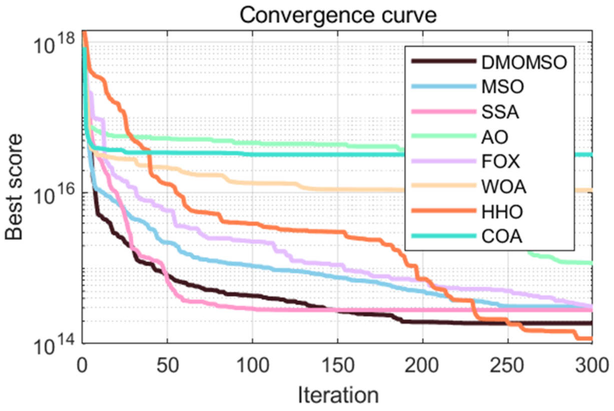


Figure 14. Convergence curves for industrial refrigeration system design tests.

Table 8. Test results for industrial refrigeration systems design optimization.

Algorithms	DMOMSO	MSO	SSA	AO	FOX	WOA	HHO	COA
min	3.75E-02	3.99E-02	3.34E-02	9.40E+14	5.56E+00	2.15E+00	4.41E+00	1.73E+15
std	3.95E+14	5.02E+14	4.52E+14	2.44E+14	7.20E+14	1.71E+16	3.71E+14	1.99E+16
avg	1.87E+14	3.09E+14	2.81E+14	1.18E+15	3.12E+14	1.10E+16	**1.17E+14**	3.25E+16

6. Conclusions

In this study, we focus on improving the MSO algorithm and propose a novel algorithm: The dual-chaos mechanism opposition-based learning mirage search optimization (DMOMSO) algorithm. By incorporating Iterative chaotic initialization, Fuch chaotic mapping, and Beta opposition-based learning strategies, the DMOMSO algorithm not only accelerates the convergence process to optimal solutions but also expands the search scope, significantly enhancing the algorithm's global search capability.

Comprehensive simulation experiments were conducted on the CEC2017 and CEC2022 benchmark function sets, comparing DMOMSO with 7 other algorithms including MSO, SSA, AO, FOX, WOA, HHO, and COA. The experimental results demonstrate that DMOMSO outperforms all other algorithms, including the original MSO, exhibiting superior search capability and faster convergence speed.

Furthermore, we applied these 8 algorithms to three engineering optimization problems: Hydrostatic thrust bearing design, Himmel Blau's function optimization, and industrial refrigeration system design to evaluate their performance in handling complex engineering challenges. The results show that DMOMSO achieves better performance than other algorithms across all three engineering problems, demonstrating its effectiveness in solving these difficult optimization tasks.

For future research, we plan to further integrate the MSO algorithm with other optimization techniques to obtain more accurate solutions and accelerate convergence. Additionally, we intend to introduce multi-objective optimization techniques (such as Pareto solution sets) to enable the algorithm to simultaneously identify multiple optimal solutions.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This study was supported by the University Key Research Project of Department of Education Anhui Province (2022AH051683 and 2024AH051994), the University Innovation Team Project of Department of Education Anhui Province (2023AH010078), and Scientific Research Foundation for High-level Talents of West Anhui University (WGKQ2023003 and WGKQ2022046).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. A. E. Smith, Swarm intelligence: from natural to artificial systems [Book Reviews], *IEEE Trans. Evol. Comput.*, **4** (2000), 192–193. https://doi.org/10.1109/TEVC.2000.850661
2. A. E. Eiben, J. Smith, From evolutionary computation to the evolution of things, *Nature*, **521** (2015), 476–482. https://doi.org/10.1038/nature14544
3. K. Dai, S. Lu, X. Jiang, Solving the multi-person traveling salesman problem based on genetic algorithm, *Comput. Eng.*, **30** (2004), 3. https://doi.org/10.3969/j.issn.1000-3428.2004.16.054
4. H. Nikravesh, A. Ranjbar, R. Azin, Optimizing the wells location using the Metaheuristic algorithms, as well as optimizing the drilling time of production and injection wells in one of the reservoirs in south west of Iran, *Earth Sci. Inf.*, **17** (2024). https://doi.org/10.1007/s12145-023-01204-3
5. S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680. https://doi.org/10.1126/science.220.4598.671
6. Z. Li, J. Yu, C. Qian, Research on layout optimization of defective rectangular parts based on dual population genetic algorithm, *Mach. Electron.*, **41** (2023), 7–12. https://doi.org/10.19850/j.cnki.2096-4706.1111
7. Y. Granik, Source optimization for image fidelity and throughput, *J. Micro/Nanolithogr. MEMS MOEMS*, **3** (2004), 509–522. https://doi.org/10.1117/1.1794708
8. J. Liu, S. Chen, Multi-objective optimization of oilfield logistics with marginal well consideration, *Appl. Soft Comput.*, **145** (2024), 110623. https://doi.org/10.1016/j.asoc.2024.110623
9. C. J. Shih, C. Y. Hsu, C. Y. Kuo, J. Li, J. C. Rau, K. Chakrabarty, Thermal-aware test schedule and TAM co-optimization for three-dimensional IC, *Act. Passive Electron. Compon.*, **2012** (2012), 763572. https://doi.org/10.1155/2012/763572
10. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
11. C. Pu, Y. Jia, Z. Zhang, H. Luo, M. Ren, J. Wang, et al., Intelligent optimization of air-floating piston core parameters for homemade frictionless pneumatic actuators based on a new multi-objective particle swarm optimization algorithm with Gaussian mutation and fuzzy logic, *Eng. Appl. Artif. Intell.*, **154** (2025), 111053. https://doi.org/10.1016/j.engappai.2025.111053

12. P. Qian, H. Luo, L. Liu, P. Lv, C. Pu, D. Meng, et al., A hybrid Gaussian mutation PSO with search space reduction and its application to intelligent selection of piston seal grooves for homemade pneumatic cylinders, *Eng. Appl. Artif. Intell.*, **122** (2023), 14. https://doi.org/10.1016/j.engappai.2023.106156.
13. P. Qian, C. Pu, L. Liu, H. Luo, J. Wu, Y. Jia, et al., Ultra-high-precision pneumatic force servo system based on a novel improved particle swarm optimization algorithm integrating Gaussian mutation and fuzzy theory, *ISA Trans.*, **152** (2024), 14. https://doi.org/10.1016/j.isatra.2024.06.024
14. D. Riofrio, S. Luan, J. Zhou, L. Ma, Particle swarm optimization for radiation therapy planning, in *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, (2015), 250–257. https://doi.org/10.1145/2808719.2808745
15. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, **26** (1996), 29–41. https://doi.org/10.1109/3477.484436
16. V. Subrahmanyam, O. S. Nagesh, Y. V. Reddy, A. P. Rao, M. Suresh, G. Neelima, Internet of Things (IoT) enabled intelligent traffic control management system by hybrid swarm intelligence (SI) algorithm, in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, **2** (2024). https://doi.org/10.1109/IATMSI60426.2024.10503563
17. M. Varshney, P. Kumar, M. R. Ali, Y. Gulzar, Using the grey wolf aquila synergistic algorithm for design problems in structural engineering, *Biomimetics*, **9** (2024), 54. https://doi.org/10.3390/biomimetics9010054.
18. T. Cura, A rapidly converging artificial bee colony algorithm for portfolio optimization, *Knowl.-Based Syst.*, **233** (2021), 107505. https://doi.org/10.1016/j.knosys.2021.107505
19. J. He, S. Zhao, J. Ding, Y. Wang, Mirage search optimization: application to path planning and engineering design problems, *Adv. Eng. Software*, **203** (2025). https://doi.org/10.1016/j.advengsoft.2025.103883
20. V. Elser, I. Rankenburg, P. Thibault, Searching with iterated maps, *Proc. Natl. Acad. Sci. U.S.A.*, **104** (2007), 418–423. https://doi.org/10.1073/pnas.0606359104
21. W. Y. Fu, G. G. Li, Y. Q. Wang, Reverse Chaotic Optimization Algorithm with Variable Interval Length, *Acta Electronica Sin.*, **47** (2019), 113–121.
22. W. Gong, A. Fialho, Z. Cai, H. Li, stochastic opposition-based learning using a beta distribution in differential evolution, *IEEE Trans. Cybern.*, **46** (2016), 2184–2196, https://doi.org/10.1109/TCYB.2015.2475736
23. A. Ahrari, S. Elsayed, R. Sarker, D. Essam, C. A. C. Coello, Problem definition and evaluation criteria for the CEC'2022 competition on dynamic multimodal optimization, in *Proceedings of the IEEE World Congress on Computational Intelligence (IEEE WCCI 2022)*, Italy, (2022), 18–23.
24. X. M. Liang, L. Y. Shi, W. Long, An improved snake optimization algorithm based on hybrid strategies and its application, *Comput. Eng. Sci.*, **46** (2024), 693–706.
25. J. K. Xue, B. Shen, A novel swarm intelligence optimization approach: Sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. https://doi.org/10.1080/21642583.2019.1708830
26. L. Abualigah, D. Yousri, M. A. Elaziz, E. G. Ewees, A. H. Gandomi, Matlab code of aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **2021** (2021). https://doi.org/10.1016/j.cie.2021.107250

27. H. Mohammed, T. Rashid, FOX: a FOX-inspired optimization algorithm, *Appl. Intell.*, **53** (2023), 1030–1050. https://doi.org/10.1007/s10489-022-03533-0
28. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawk's optimization: algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
29. H. Jia, H. Rao, C. Wen, S. Mirjalili, Crayfish optimization algorithm, *Artif. Intell. Rev.*, **56** (2023), 1919–1979. https://doi.org/10.1007/s10462-023-10567-4
30. J. Pierezan, L. D. S. Coelho, Coyote optimization algorithm: a new metaheuristic for global optimization problems, in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018. https://doi.org/10.1109/CEC.2018.8477769
31. A. Osyczka, S. Krenich, K. Karas, Optimum design of robot grippers using genetic algorithms, in *Proceedings of the Third World Congress 560 of Structural and Multidisciplinary Optimization (WCSMO)*, Buffalo, New York, (1999), 241–243.
32. A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, *Swarm Evol. Comput.*, **51** (2021), 100961. https://doi.org/10.1016/j.swevo.2021.100961
33. Y. Li, X. Liang, J. Liu, H. Zhou, Solving engineering optimization problems based on improved balance optimizer algorithm, *Comput. Integr. Manuf. Syst.*, (2023), 1–34. https://doi.org/10.13196/j.cims.2023.0169