

Article

Not peer-reviewed version

---

# MRBMO: An Enhanced RBMO Algorithm For Solving Numerical Optimization Challenges

---

[Junhao Wei](#) , [Yanzhao Gu](#) , Yuzheng Yan , Zikun Li , [Baili Lu](#) , Shirou Pan , [Ngai Cheong](#) <sup>\*</sup> , Ruishen Zhou

Posted Date: 4 February 2025

doi: 10.20944/preprints202502.0188.v1

Keywords: RBMO; HHO; good nodes set; LIOBL; levy flight



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

## Article

# MRBMO: An Enhanced RBMO Algorithm For Solving Numerical Optimization Challenges

Junhao Wei <sup>1</sup>, Yanzhao Gu <sup>1</sup>, Yuzheng Yan <sup>1</sup>, Zikun Li <sup>2</sup>, Baili Lu <sup>3</sup>, Shirou Pan <sup>3</sup>,  
Ngai Cheong <sup>2,\*</sup> and Ruishen Zhou <sup>1</sup>

<sup>1</sup> Faculty of Applied Sciences, Macao Polytechnic University, Macao, 999078 China;

<sup>2</sup> School of Economics and Management, South China Normal University, Guangzhou, 510006 China;

<sup>3</sup> College of Animal Science and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou, 510225 China

\* Correspondence: ncheong@mpu.edu.mo

**Abstract:** To solve the limitation of easily getting trapped in local optima and slow convergence rates of Red-billed Blue Magpie Optimization algorithm (RBMO), an enhanced RBMO algorithm (MRBMO) was proposed. MRBMO was improved by integrating Good Nodes Set Initialization, an Enhanced Search-for-food Strategy, a newly-designed Siege-style Attacking-prey Strategy and Lens-Imaging Opposition-Based Learning (LIOBL). The experimental results show that MRBMO is superior to traditional algorithms in terms of convergence speed and solving accuracy, especially in high-dimensional search space. This paper designs two types of simulation experiments to test the practicability of MRBMO. First, MRBMO is used along with other heuristic algorithms to solve four engineering design optimization problems, aiming to verify the applicability of MRBMO in engineering design optimization. Then, to overcome the shortcomings of metaheuristic algorithms in antenna S-parameter optimization problems, such as time-consuming verification processes, cumbersome operations, and complex modes, this paper adopts a test suite specifically designed for antenna S-parameter optimization, with the goal of efficiently validating the effectiveness of metaheuristic algorithms in antenna S-parameter optimization. The results show that MRBMO demonstrates significant advantages in both engineering design optimization and antenna S-parameter optimization.

**Keywords:** RBMO; HHO; good nodes set; LIOBL; levy flight

## 1. Introduction

In the field of optimization, metaheuristic algorithms have received widespread attention due to their effectiveness and applicability in solving complex multimodal problems. Metaheuristic algorithms are an improvement of heuristic algorithms and are a combination of random algorithms and local search algorithms. These algorithms are used to solve complex optimization problems by performing global search and local exploration to find the optimal or near-optimal solution. The core of metaheuristic algorithms lies in exploration and exploitation. Exploration refers to the process of thoroughly exploring the entire search space since the optimal solution may exist at any position within the space. Exploitation, on the other hand, focuses on utilizing the available information as much as possible. In most cases, there is a certain correlation between the optimal solutions, and by exploiting these correlations, the algorithm can gradually adjust and evolve from an initial solution to an optimal solution. The main advantage of metaheuristic algorithms is their ability to handle complex, nonlinear problems without requiring assumptions about the specific model of the problem. Although metaheuristic algorithms cannot guarantee a global optimal solution for an optimization problem, they are capable of quickly finding or approximating the optimal solution within a given timeframe.

Over the past few decades, various metaheuristic algorithms have been developed. The Simulated Annealing (SA) algorithm, proposed by Metropolis in 1953, is based on the analogy between the process of solving an optimization problem and the thermal equilibrium problem in statistical thermodynamics.

The aim is to find the global or near-optimal solution by simulating the annealing process of a high-temperature **object** [24]. The Genetic Algorithm (GA), first proposed by John Holland in 1975, is based on Darwin's theory of evolution and Mendel's genetics. It uses methods such as reproduction, mutation, and competition among individuals in a population to exchange information and perform a survival-of-the-fittest approach, gradually approaching the optimal solution [2]. The Ant Colony Optimization (ACO) algorithm, proposed by Dorigo et al. in 1991, is a random search algorithm that simulates the food foraging process of real ants in nature [3]. In 1995, American psychologist Kennedy and electrical engineer Eberhart proposed the Particle Swarm Optimization (PSO) algorithm, inspired by the foraging behavior of birds [1]. Due to their excellent optimization ability and versatility, metaheuristic algorithms have been widely applied in various fields such as robot path planning, job shop scheduling, neural network parameter optimization, and feature selection. However, many traditional metaheuristic algorithms, such as PSO, GA, and ACO, often face issues such as getting trapped in local optima and slow convergence when dealing with complex problems. To address these challenges, researchers have attempted to integrate various improvement strategies into basic metaheuristic algorithms, including hybrid algorithms, enhanced exploration-exploitation balance methods, and the introduction of new biological behavior models.

In 2018, Guojiang Xiong et al. proposed the improved whale optimization algorithm (IWOA) with a novel search strategy for solving solar photovoltaic model parameter extraction problems [36]. In 2019, Qiang Tu et al. introduced the Multi-strategy Ensemble Grey Wolf Optimizer (MEGWO) by incorporating global best-guided strategies and adaptive collaborative strategies into the Grey Wolf Optimizer (GWO), aiming to overcome the limitations of the single search strategy in traditional GWO for solving various function optimization problems [37]. In 2023, Ya Shen et al. proposed an improved whale optimization algorithm based on multi-population evolution (MEWOA). The algorithm divides the population into three sub-populations based on individual fitness and assigns different search strategies to each sub-population. This multi-population cooperative evolution strategy effectively enhances the algorithm's search capability [38]. In 2024, Ying Li et al. proposed the Improved Sand Cat Swarm Optimization algorithm (VF-ISCSO) based on virtual forces and a nonlinear convergence strategy. VF-ISCSO demonstrated significant advantages in enhancing the coverage range of wireless sensor networks [39]. These outstanding algorithms, by integrating various novel improvement strategies, offer new insights into the enhancement of metaheuristic algorithms.

The Red-Billed Blue Magpie Optimization (RBMO) algorithm is a novel metaheuristic algorithm inspired by the foraging behavior and social cooperation characteristics of the red-billed blue magpie [5]. RBMO demonstrates significant advantages in global search, population diversity, and ease of implementation. However, it still exhibits limitations in solution accuracy and convergence speed, particularly when dealing with complex multimodal problems, where it struggles to quickly approach the optimal solution. To address these challenges, this paper proposes an enhanced RBMO algorithm (MRBMO). MRBMO was improved by integrating Good Nodes Set Initialization, an Enhanced Search-for-food Strategy, a newly designed Siege-style Attacking-prey Strategy, and Lens-Imaging Opposition-Based Learning (LIOBL). These strategies enhance the local search ability of RBMO and accelerate its convergence speed.

## 2. Research Work on Antenna Design

Adegboye et al. introduced the Honey Badger Algorithm (HBA) for antenna design optimization, demonstrating the algorithm's effectiveness in enhancing antenna performance through specific cases [12]. The improvements were evident in key metrics such as gain and bandwidth. They compared the performance of various algorithms in antenna design, highlighting the advantages of the new algorithm in addressing particular design challenges. Park et al. proposed a method for optimizing antenna placement in single-cell and dual-cell distributed antenna systems (DAS) to maximize the lower bounds of expected signal-to-noise ratio (SNR) and expected signal-to-leakage ratio (SLR) [1]. The results indicated that the DAS using the proposed gradient ascent-based algorithm outperforms

traditional centralized antenna systems (CAS) in terms of capacity, especially in dual-cell environments, effectively reducing interference and improving system performance. Jiang et al. designed a multi-band pixel antenna using genetic algorithms and N-port characteristic modal analysis, which operates effectively across the 900 MHz, 1800 MHz, and 2600 MHz frequency bands [13]. The effectiveness of the genetic algorithm in antenna design optimization was validated by monitoring changes in the objective function. Yang Zhao et al. proposed an optimization design method for dual-band tag antennas based on a multi-population genetic algorithm, overcoming the inefficiencies of traditional simulation and parameter tuning experiments in determining optimal size parameters [14]. The optimized UHF antenna achieved near-ideal input impedance at 915 MHz, resulting in good impedance matching with the chip. The size of the optimized dual-band tag antenna was significantly reduced compared to existing designs. Cai Jiaqi et al. presented a self-optimization method for base station antenna azimuth and downtilt angles based on the Artificial Bee Colony algorithm [15]. Experimental results provided the optimal azimuth and downtilt angles for base station antennas, improving coverage effectiveness for user devices, particularly in weak coverage areas. Fengling Peng et al. developed an antenna optimization framework based on Differential Evolution (DE), customized decision trees, and Deep Q-Networks (DQN) [16]. Experimental results showed that this hybrid strategy-based framework achieves superior antenna design solutions with fewer simulation iterations. Meta-heuristic algorithm plays an important role in antenna design.

### 3. Research Work on Engineering Design Optimization

In early engineering design, the concept of "optimization" was typically absent. Engineering design relied heavily on manual calculations and intuitive judgment. In the early 20th century, with the development of mathematical optimization theory, optimization tools gradually began to be introduced into engineering design. The initial optimization methods were based on classical mathematical analysis, such as calculus, to derive the optimal solution through analytical reasoning. In the 1940s, George Dantzig and others proposed Linear Programming (LP), which provided a mathematical foundation for optimization and was widely applied in economics, transportation, and resource allocation problems. With the emergence of nonlinear systems, engineering design faced more complex optimization problems, necessitating new mathematical tools. Numerical methods, such as Newton's method and gradient descent, were introduced and applied to engineering optimization. However, due to the lack of powerful computational tools, engineering design remained a time-consuming and complex process, mostly relying on simplified assumptions to solve problems. From the 1950s to the 1970s, advancements in computers drove the computerization of engineering design optimization. Engineers began to use computer programs, achieving breakthroughs particularly in Finite Element Analysis (FEA) and Dynamic Programming (DP). The Finite Element Method (FEM) helped address stress and deformation problems in complex structures, while Dynamic Programming was widely applied in control and scheduling problems. Despite these advancements, early computer-aided design still mainly used deterministic mathematical methods for solutions, which were applicable only to specific types of problems. When facing large-scale, complex engineering problems, the computational effort and difficulty of solving remained significant. By the 1980s, as the scale and complexity of engineering design problems increased, traditional mathematical optimization methods gradually became inadequate for handling high-dimensional, nonlinear, and complex-constrained optimization issues. Researchers shifted towards metaheuristic algorithms (such as Simulated Annealing, Genetic Algorithms, Particle Swarm Optimization, etc.). These algorithms could avoid local optima and offer better optimization solutions, especially for complex, irregular, or high-dimensional problems. The improvement in computer hardware performance also allowed these algorithms to handle larger-scale optimization problems. In the 2010s, the development of Artificial Intelligence (AI) and Deep Learning (DL) provided new options for engineering design optimization. Neural networks could assist design optimization by learning from large amounts of historical data. However, AI has the disadvantage of long training times and dependence on large datasets, which may yield poor results when data is

scarce. AI is also constrained by existing mathematical models, whereas metaheuristic algorithms, due to their robustness, can maintain high optimization performance across various complex problems without relying on mathematical models or derivative information, demonstrating strong adaptability. Therefore, metaheuristic algorithms remain the primary solution method for engineering design optimization.

#### 4. Arrangement of the Rest of the Paper

Chapter 5 primarily presents the main contributions of this study. Chapter 6 introduces the specific principles of the RBMO algorithm, along with its advantages and disadvantages. Chapter 7 introduces the proposed MRBMO. In Chapter 8, we will test the performance of MRBMO through a series of experiments. In Chapter 9, various metaheuristic algorithms and MRBMO will be simulated and tested on different engineering design optimization problems and antenna S-parameter optimization suites to verify the practicality of MRBMO.

#### 5. contributions of This Study

By integrating Good Nodes Set Initialization, an Enhanced Search-for-food Strategy, a newly designed Siege-style Attacking-prey Strategy and Lens-Imaging Opposition-Based Learning (LIOBL), we proposed a novel optimizer MRBMO for solving real-world challenges. By performing an ablation study, we evaluated the effectiveness of each strategy. By comparing MRBMO with other excellent metaheuristic algorithms on the classical benchmark functions, we validated the outstanding performance of MRBMO. In a subsequent series of simulation experiments, MRBMO demonstrated excellent optimization ability and good convergence, proving that MRBMO can be used in the real world to solve various numerical optimization problems.

#### 6. Red-Billed Blue Magpie Optimization Algorithm

The red-billed blue magpie is a bird that lives mainly in Asia and is common in China, India, and Myanmar, as shown in Figure 1, which depicts the red-billed blue magpie, which is characterized by its large size, bright blue plumage, and distinctive red beak. The red-billed blue magpie feeds mainly on insects, small vertebrates, plants, etc., and has relatively abundant hunting behavior. When foraging, red-billed blue magpies use a combination of jumping, ground walking and searching for food on branches.

Red-billed blue magpies show high activity levels in the early morning and evening, often gathering in small groups of 2-5 or even more than 10 individuals. They are involved in co-operative hunting. For example, a magpie may find fruit or an insect which it will then attract other members to share. This allows them to co-operate in catching large insects or small vertebrates, and group action can help them overcome the defence mechanisms of their prey. Magpies also store some food for later use. To prevent other birds or animals from stealing it, they hide food in places such as tree holes, branches and rock crevices.

Overall, red-billed blue magpies are flexible predators that acquire and store food in a variety of ways, as well as showing sociality and cooperation in their hunting behaviour. Inspired by this, Shengwei Fu et al. proposed a new metaheuristic algorithm, the Red-billed Blue Magpie Optimization algorithm (RBMO) in 2024 [4]. When RBMO deals with complex problems, each optimization problem has its own objective function, the solution space consists of the values of an objective function. The mission for RBMO is to search the optimal or suboptimal solution in the solution space. In each iteration, RBMO will randomly generate  $N$  individuals in the solution space. The individuals are called search agents, and they will move by imitating the behavior of the red-billed blue magpie in searching for prey, attacking prey or storing food. Meanwhile, they will update their position and the position is called a solution or the fitness of an objective function. After many iterations, they can find the optimal or suboptimal solution in the solution space.



**Figure 1.** a red-billed blue magpie

### 6.1. Search for Food

In red-billed blue magpies' search-for-food stage, they use a variety of methods such as hopping on the ground, walking or searching for food resources in trees. The whole flock will be divided into small groups of 2-5 individuals or in clusters of 10 or more to search for food.

RBMO imitates their search-for-food behavior in small groups as follows,

$$X^i(t+1) = X^i(t) + \left(\frac{1}{p} \cdot \sum_{m=1}^p X^m(t) - X^{rs}(t)\right) \cdot Rand_1 \quad (1)$$

where  $t$  represents the current iteration number;  $X^i(t)$  represents the location of  $t^{th}$  new search agent;  $p$  is a random integer between 2 and 5, representing the number of red-billed blue magpies in a population of 2 to 5 randomly selected from all searched individuals;  $X^M$  represents the  $m^{th}$  randomly selected individual;  $X^i$  represents the  $i^{th}$  individual; and  $X^{rs}$  represents the randomly selected search agent in the current iteration;  $Rand_1$  is a random number range from [0,1].

Also, RBMO imitates their search-for-food behavior in clusters as follows,

$$X^i(t+1) = X^i(t) + \left(\frac{1}{q} \cdot \sum_{m=1}^q X^m(t) - X^{rs}(t)\right) \cdot Rand_2 \quad (2)$$

where  $q$  is a random integer between 10 and  $n$ , representing the number of red-billed blue magpies in a population of 10 to  $n$  randomly selected from all searched individuals;  $Rand_2$  is a random number range from [0,1].

The whole Search-for-food Strategy is modeled below.

$$X^i(t+1) = X^i(t) + \left(\frac{1}{p} \cdot \sum_{m=1}^p X^m(t) - X^{rs}(t)\right) \cdot Rand_1, \quad rand < \varepsilon \quad (3)$$

$$X^i(t+1) = X^i(t) + \left(\frac{1}{q} \cdot \sum_{m=1}^q X^m(t) - X^{rs}(t)\right) \cdot Rand_2, \quad rand \geq \varepsilon \quad (4)$$

### 6.2. Attacking Prey

When attacking prey, red-billed blue magpies show a high degree of hunting proficiency and co-operation. Red-billed blue magpies use tactics such as rapid pecking, jumping to catch prey or flying to catch insects, and they usually move in small groups of 2-5 or in clusters of 10 or more to increase hunting efficiency.

RBMO imitates red-billed blue magpies' attacking-prey behavior in small groups as follows,

$$X^i(t+1) = X^{food}(t) + CF \cdot \left(\frac{1}{p} \cdot \sum_{m=1}^p X^m(t) - X^i(t)\right) \cdot Randn_1 \quad (5)$$

where  $X^i(t)$  represents the location of  $t^{th}$  new search agent;  $X^{food}$  represents the position of the food, which indicates the current optimal solution;  $p$  is a random integer between 2 and 5, representing the number of red-billed blue magpies in a population of 2 to 5 randomly selected from all searched individuals;  $Randn_1$  denotes the random number used to generate the standard normal distribution (mean 0, standard deviation 1);  $CF$  is the step control factor, calculated as in Eq 6.

$$CF = \left(1 - \frac{t}{T}\right)^{\left(2 \cdot \frac{t}{T}\right)} \quad (6)$$

where  $t$  represents the current number of iterations;  $T$  represents the maximum number of iterations.

RBMO imitates red-billed blue magpies' attacking-prey behavior in clusters as follows,

$$X^i(t+1) = X^{food}(t) + CF \cdot \left(\frac{1}{q} \cdot \sum_{m=1}^q X^m(t) - X^i(t)\right) \cdot Randn_2 \quad (7)$$

where  $q$  is a random integer between 10 and  $n$ , representing the number of red-billed blue magpies in a population of 10 to  $n$  randomly selected from all searched individuals;  $Randn_2$  denotes the random number used to generate the standard normal distribution (mean 0, standard deviation 1).

The Attacking prey is modeled below.

$$X^i(t+1) = X^{food}(t) + CF \cdot \left(\frac{1}{p} \cdot \sum_{m=1}^p X^m(t) - X^i(t)\right) \cdot Randn_1, \quad rand < \varepsilon \quad (8)$$

$$X^i(t+1) = X^{food}(t) + CF \cdot \left(\frac{1}{q} \cdot \sum_{m=1}^q X^m(t) - X^i(t)\right) \cdot Randn_2, \quad rand \geq \varepsilon \quad (9)$$

### 6.3. Food Storage

As well as searching for food and attacking food, red-billed blue magpies store excess food in tree holes or other hidden places for future consumption, ensuring a steady supply of food in times of shortage.

RBMO imitates the storing-food behavior of red-billed blue magpies. And the formula for storing food is shown in Eq 10.

$$X^i(t+1) = \begin{cases} X^i(t) & \text{if } fitness_{old}^i > fitness_{new}^i \\ X^i(t+1) & \text{else} \end{cases} \quad (10)$$

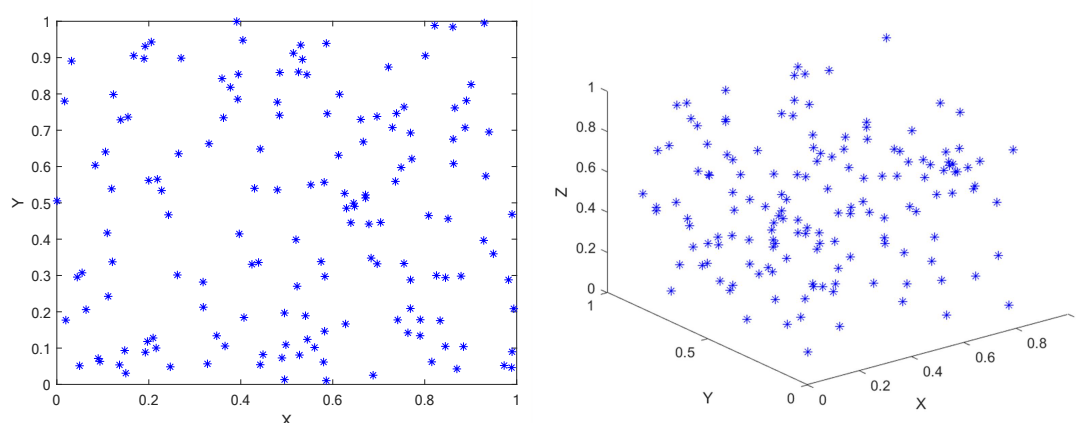
where  $fitness_{old}^i$  and  $fitness_{new}^i$  denote the fitness values before and after the position update of the  $i^{th}$  red-billed blue magpie respectively.

### 6.4. Initialization

Like most metaheuristic algorithms, RBMO uses Pseudo-random number initialization for population initialization. This approach, while simple and direct, often results in poor diversity and uneven distribution of solutions, which can lead to inefficiency in the search process. Figure 2 is the population initialized by Pseudo-random number method.

$$X_{i,j} = (ub - lb) \cdot Rand + lb \quad (11)$$

where  $X_{i,j}$  is randomly produced population;  $ub$  and  $lb$  are the upper limit and lower limit of the problem;  $Rand$  is a random number between 0 and 1.



**Figure 2.** Pseudo-random number initialization ( $N=150$ )

### 6.5. Workflow of RBMO and Its Analysis

The workflow of RBMO is provided in Figure 3.

As a new biologically metaheuristic algorithm, RBMO has significant advantages in global search, population diversity and simplicity of implementation. The update mechanism of RBMO increases the breadth and diversity of the search by randomly selecting the mean values of multiple individuals for updating, which enables it to cover a larger search space, thus effectively avoiding falling into a local optimum. Secondly, the algorithm structure of RBMO is simple and easy to implement, which is suitable for the rapid solution of different fields and problems. In addition, by randomly selecting individuals and mean updating strategies, RBMO is able to adapt to different types and sizes of optimization problems, showing good stability and adaptability.

However, RBMO also has some limitations, especially in the local search ability and convergence speed. As the attacking-prey strategy of RBMO is relatively monotonous, which leads to its insufficient local search ability when facing complex and multi-peak problems, it is difficult to approach the optimal solution quickly. In addition, the convergence speed of RBMO is relatively slow, and more iterations are needed to find a better solution in the optimization process. These shortcomings limit the effectiveness and efficiency of RBMO application to some extent. In order to solve these problems, enhance the local search capability of RBMO and accelerate the convergence speed, we propose an enhance RBMO, which is called MRBMO.

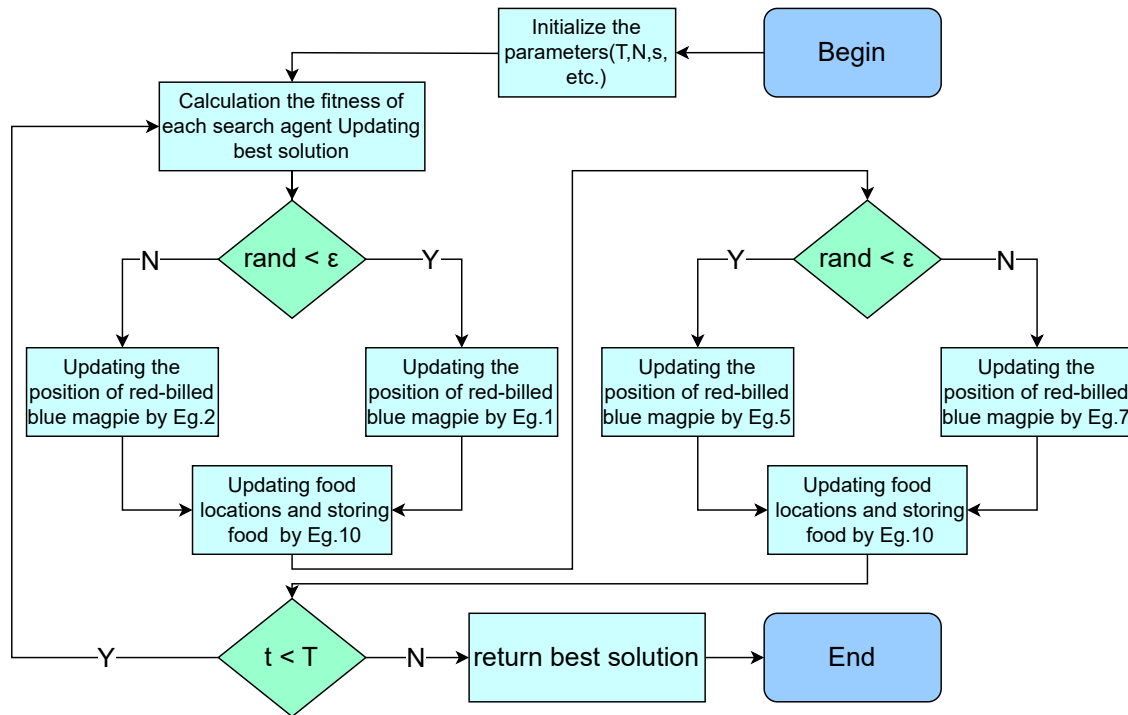


Figure 3. Workflow of RBMO

## 7. MRBMO

### 7.1. Good Nodes Set Initialization

The original RBMO uses pseudo-random number method to initialize the population, this method is simple, direct and random, but there are some drawbacks. the randomly generated population, as shown in Figure 2, is not uniformly distributed throughout the solution space, it is very aggregated in some areas and scattered in some areas, which leads to the algorithm's poor exploitation of the whole search space and low diversity of the population. Therefore, some experts proposed to use chaotic mapping, random wandering, Gaussian distribution and other methods to initialize the population. Later some scholars proposed to use the good nodes set initialization.

The theory of Good Nodes Set was first proposed by the famous Chinese mathematician Loo-keng Hua. Good Nodes Set is a method used to cover a multidimensional space uniformly, aiming to improve the quality of initialized populations. Compared with the traditional initialization method, Good Nodes Set initialization, as shown in Figure 4, can better distribute the nodes and improve the diversity of the population, thus providing better initial conditions for the optimization algorithm. This method is also effective in high dimensional spaces.

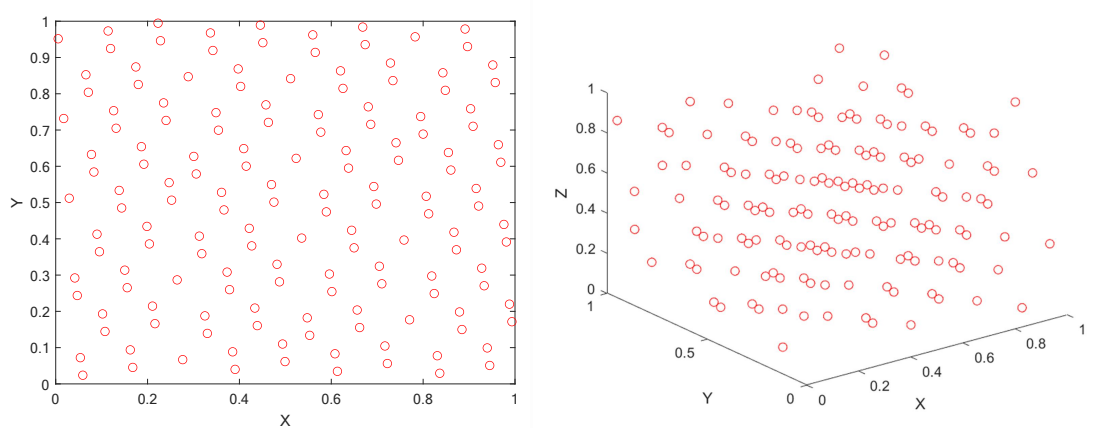
Assuming that  $U^D$  is a unit cube in the  $D$  dimensional Euclidean space, and assuming that  $r$  is a parameter, the set of canonical nodes  $P_r^M$  has the form shown in Eq 12:

$$P_r^M = \{p(k) = (\{kr\}, \{kr^2\}, \dots, \{kr^D\}) | k = 1, 2, \dots, M\} \quad (12)$$

where  $\{x\}$  represents the fractional part of  $x$ ;  $M$  is the number of points;  $r$  is a deviation parameter greater than zero; the constant  $C(r, \varepsilon)$  is associated only with  $r$  and  $\varepsilon$  is related to and is a constant greater than zero.

This set  $P_r^M$  is called Good Nodes Set and each node  $p(k)$  in it is called a Good Node. Assume that the upper and lower bounds of the  $i^{th}$  dimension of the search space  $x_{max}^i$  are and  $x_{min}^i$ , then the mapping formula for mapping the Good Nodes Set to the actual search space is:

$$x_k^i = x_{min}^i + p_i(k) \cdot (x_{max}^i - x_{min}^i) \quad (13)$$



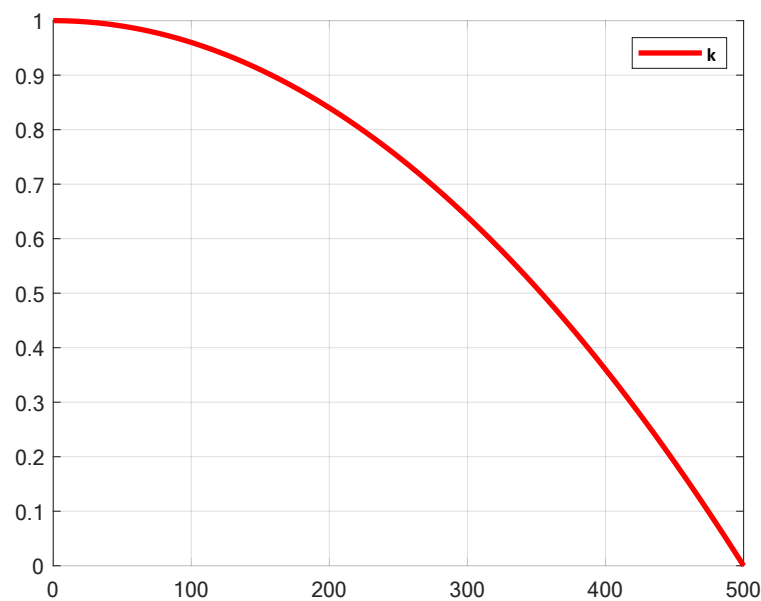
**Figure 4.** Good Nodes Set initialization ( $N=150$ )

### 7.2. Enhanced Search-for-food Strategy

In the original RBMO, the Search-for-food phase relies on the random number and lacks dynamic adjustment, which results in large random fluctuations between the population individuals. Particularly in the later stages of iteration, individuals may still explore with large step sizes, leading to a decrease in search efficiency and affecting convergence accuracy. Therefore, this paper introduces a nonlinear factor  $k$ , which enables more thorough exploration in the early stages and more refined development in the later stages. The variation process of  $k$  is shown in Figure 5. And the calculation of the nonlinear factor  $k$  is as follows:

$$k = 1 - \left( \frac{t}{T} \right)^2 \quad (14)$$

where  $t$  represents the current number of iterations;  $T$  represents the maximum number of iterations.



**Figure 5.** The variation process of  $k$

In the early iterations, the value of  $k$  is close to 1, which enhances the large step movements between the population individuals, thus improving global exploration ability. In the later iterations,

the value of  $k$  approaches 0, limiting the movement range of individuals and gradually transitioning towards local exploitation. By introducing the nonlinear factor  $k$ , the search intensity of RBMO dynamically decays, naturally balancing exploration and exploitation, thereby improving convergence stability. The Enhanced Search-for-food Strategy addresses the issue of excessive randomness in the search phase of the original algorithm. The Enhanced Search-for-food Strategy is modeled below.

$$X^i(t+1) = X^i(t) + k \cdot \left(\frac{1}{p} \cdot \sum_{m=1}^p X^m(t) - X^{rs}(t)\right), \quad rand < \varepsilon \quad (15)$$

$$X^i(t+1) = X^i(t) + k \cdot \left(\frac{1}{q} \cdot \sum_{m=1}^q X^m(t) - X^{rs}(t)\right), \quad rand \geq \varepsilon \quad (16)$$

### 7.3. Siege-Style Attacking-prey Strategy

#### 7.3.1. HHO

The original RBMO algorithm is prone to getting stuck in local optima because the average position induces a contraction effect on the dynamic range of the population, limiting further exploration of the search space. Furthermore, in the original RBMO algorithm, the strategy for attacking prey relies on the average position of the red-billed blue magpie individuals. This updating mechanism may lead to a decrease in population diversity, resulting in slower convergence and reduced accuracy and efficiency of the search, thus hindering further optimization. Therefore, inspired by the Harris Hawk Optimization (HHO) algorithm, this paper introduces the concept of HHO into the prey attack phase of RBMO, proposing the Siege-style Attacking-prey Strategy.

The Harris Hawk Optimization (HHO), introduced by Ali Asghar Heidari et al. in 2019, is a novel bio-inspired optimization algorithm. The HHO algorithm simulates the diverse hunting strategies of Harris hawks, allowing HHO to perform efficient global search in a larger solution space while reducing the likelihood of falling into local optima. At the development stage, the HHO algorithm fine-tunes the position of prey to perform local search, thereby finding better solutions in the local regions of the solution space. We draw inspiration from the following position updating strategy of HHO.

$$X(t+1) = \Delta X(t) - E \cdot |J \cdot X_{rabbit}(t) - X(t)| \quad (17)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (18)$$

where  $X$  is the positions of the Harris hawk at the  $t^{th}$  iteration;  $E$  is the energy of the prey;  $J$  is a prey's random step while escaping; and  $X_{rabbit}$  represents the position of the food, which indicates the current optimal solution.

The Siege-style Attacking-prey Strategy integrates the ideas of HHO, introducing the absolute difference between the prey's position  $X^{food}$  and the red-billed blue magpie individual's current position  $X^i(t)$ , combined with the step size  $CF$  to directly adjust the individual's position update step size. This mechanism helps to guide the red-billed blue magpie individuals more rapidly toward better solutions, refining the local development capability of RBMO in later stages, thereby improving solution accuracy and accelerating convergence speed. Additionally, through the combination of a random factor and nonlinear scaling, the Siege-style Attacking-prey Strategy maintains population diversity during the development phase. With the dynamically adjusted step size  $CF$ , the attacking behavior of the red-billed blue magpie individuals can adapt to the different search demands at various stages of iteration, enhancing exploration in the early stages and reinforcing exploitation in the later stages. This avoids premature convergence to a single solution and improves the robustness of the algorithm.

### 7.3.2. Levy Flight

The concept of Levy flight originates from the work of mathematician Paul Levy in the 1920s. Inspired by foraging behavior observed in nature and jump phenomena in complex systems, Levy flight combines short-range exploration with long-range jumps, resembling the foraging paths of predators such as sharks, birds, and insects. Levy flight is a stochastic walk model based on the Levy distribution, with a key characteristic of combining short-distance small steps and long-distance large jumps. This search pattern helps prevent individuals from getting trapped in local dead ends while maintaining the ability to explore the global space. The long jumps in Levy flight allow algorithms to quickly escape local regions, significantly improving the issue of local optima in complex problems. The higher proportion of short-distance small steps enables precise searching within local regions. Figure 6 shows a simulation of Levy flight. The step length  $L(s)$  of Levy flight follows the Levy distribution, calculated as follows:

$$L(s) = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (19)$$

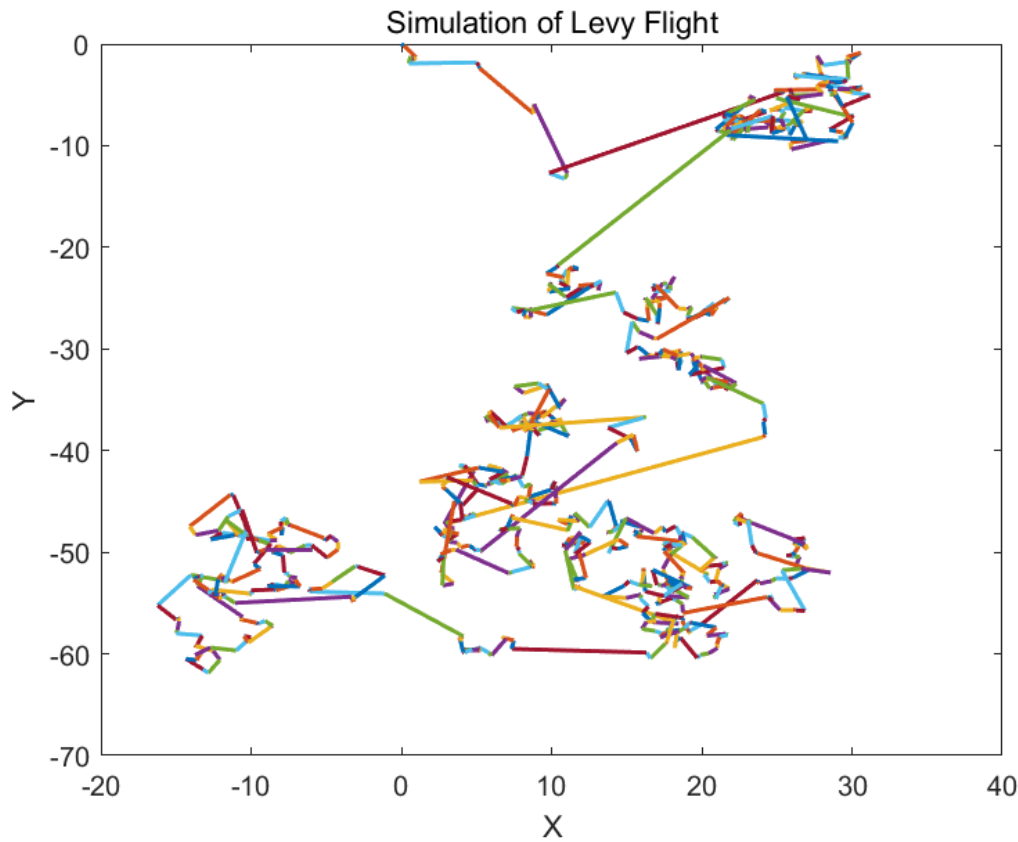
where  $u$  and  $v$  are normally distributed;  $\beta=1.5$ .

$$u \sim \mathcal{N}(0, \sigma_u^2) \quad (20)$$

$$v \sim \mathcal{N}(0, 1) \quad (21)$$

The calculation of  $\sigma_u$  is given by:

$$\sigma_u = \left( \frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (22)$$



**Figure 6. Simulation of Levy Flight**

One of the Siege-style Attacking-prey Strategy is modeled in Eq 23.

$$X^i(t+1) = \Delta X(t) - CF \cdot |r_1 \cdot X^{food}(t) - X^i(t)| \cdot L(s) \quad (23)$$

where  $X^{food}$  represents the position of the food, which indicates the current optimal solution;  $CF$  is the step control factor, calculated as in Eq 6;  $r_1$  is a random number between  $[0, 1]$ ;  $\Delta X(t)$  is calculated in Eq 18;  $L(s)$  is the step size of Levy flight.

### 7.3.3. Prey-Position-Based Enhanced Guidance

In the original attacking-prey phase, the movement of the red-billed blue magpie individuals relies on both the position of the prey and the average position of the randomly selected red-billed blue magpies. This movement strategy introduces some randomness and bias, which causes individuals to become trapped near suboptimal solutions and prevents them from fully utilizing information about the global optimum, hindering the local exploitation of the RBMO. Therefore, we propose Prey-position-based Enhanced Guidance. In this approach, we replace the average position of the randomly selected red-billed blue magpies with the position of the prey, directly guiding the red-billed blue magpie individuals towards the prey. This helps to reduce the gap between the individuals and the optimal solution. Prey-position-based Enhanced Guidance strengthens the dependency on the optimal position, mitigates the degradation of solution quality due to randomness, and enhances the concentration of local exploitation. Therefore, Prey-position-based Enhanced Guidance, one of the Siege-style Attacking-prey Strategy, is modeled in Eq 24.

$$X^i(t+1) = X^{food}(t) + CF \cdot (X^{food}(t) - X^i(t)) \cdot r_2 \quad (24)$$

where  $X^{food}$  represents the position of the food, which indicates the current optimal solution;  $CF$  is the step control factor, calculated as in Eq 6;  $r_2$  is a random number between  $[0, 1]$ . The whole Siege-style Attacking-prey Strategy is modeled below.

$$X^i(t+1) = \Delta X(t) - CF \cdot |r_1 \cdot X^{food}(t) - X^i(t)| \cdot L(s), \quad rand < \varepsilon \quad (25)$$

$$X^i(t+1) = X^{food}(t) + CF \cdot (X^{food}(t) - X^i(t)) \cdot r_2, \quad rand \geq \varepsilon \quad (26)$$

#### 7.4. Lens-Imaging Opposition-Based Learning

The food storage mechanism in RBMO passively retains better individuals, which may cause the algorithm to hover around local optima. However, the Lens-Imaging Opposition-Based Learning (LIOBL) strategy is more forward-looking. Therefore, we introduce the idea of LIOBL into the food storage mechanism of RBMO. LIOBL increases the diversity of solutions by generating multiple opposite solutions, thereby improving the exploration capability of RBMO in the later stages and enabling it to find solutions closer to the global optimum more quickly. The fundamental idea of Opposition-Based Learning (OBL) is to generate not only neighboring solutions of the current solution during the search process but also its opposite solutions, and then compare the current solution with the opposite solutions to select the better solution. The LIOBL strategy extends OBL by incorporating the concept of reflection, which is shown in Figure 7, where the opposite solutions generated by reflection are used to enhance the coverage of the solution space and improve the global exploration capability of the algorithm [30]. The opposite solutions generated by LIOBL are not only the opposites of the current solution but also include the reflected positions of the opposite solutions in the solution space. The opposition solution is calculated by Eq 27.

$$X'_i = \frac{ub + lb}{2} + \frac{ub + lb}{2\eta} - \frac{X_i}{\eta} \quad (27)$$

where  $X_i$  is the given solution;  $ub$  and  $lb$  are the upper and lower bounds of domain of definition respectively;  $\eta$  is the scaling factor of len imaging, which is set to 0.5.

Then we need to retain the better individuals through greedy meritocracy to the next generation of the population, increasing the proportion of elite individuals in the population and is shown in Eq 28.

$$X_{i+1} = \begin{cases} X_i, & \text{if } fitness(X_i) > fitness(X'_i) \\ X'_i, & \text{if } fitness(X_i) < fitness(X'_i) \end{cases} \quad (28)$$

where  $fitness(X_i)$  indicates the fitness value of  $X_i$ .

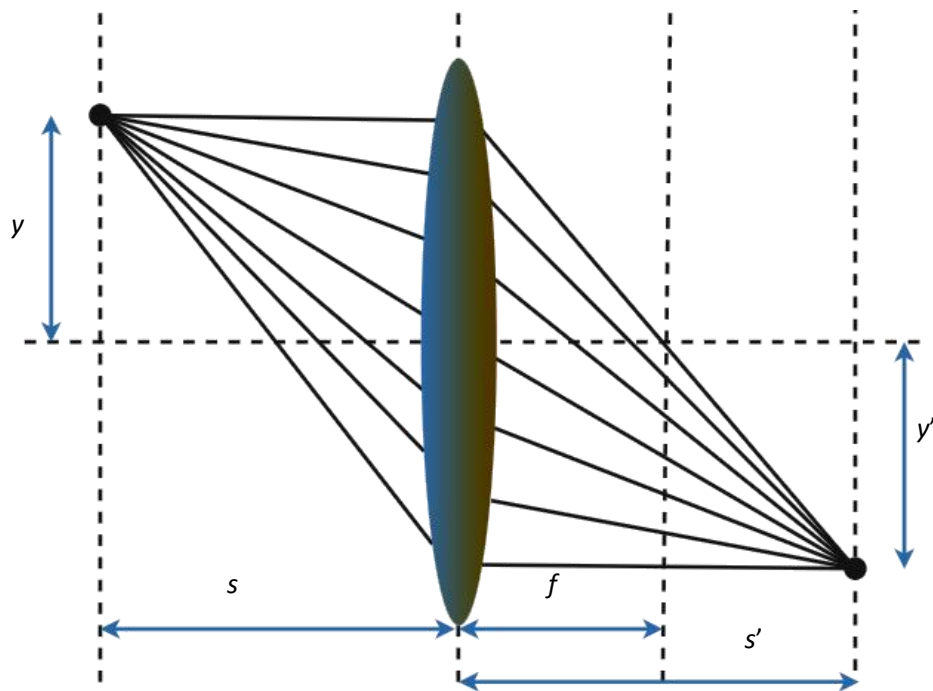


Figure 7. The concept of lens imaging

### 7.5. Time Complexity Analysis

Assume that the time complexity of initialization in RBMO is  $O(ND)$ . During each iteration, the time complexity of food storage is  $O(ND)$ , and the total time complexity of position updates is  $O(ND)$ . Therefore, the total time complexity per iteration is  $O(ND)$ . If the algorithm iterates  $T$  times, the total time complexity of RBMO is calculated as:

Total Time Complexity 1 = Initialization +  $T * (\text{the total time complexity per iteration}) = O(ND) + T * O(ND) = O(T * ND)$

Assume that the time complexity of initialization in MRBMO is  $O(ND)$ . During each iteration, the time complexity of food storage is  $O(ND)$ , the time complexity of LIOBL is  $O(ND)$ , and the total time complexity of position updates is  $O(ND)$ . Therefore, the total time complexity per iteration is  $O(ND)$ . If the algorithm iterates  $T$  times, the total time complexity of MRBMO is calculated as:

Total Time Complexity 2 = Initialization +  $T * (\text{the total time complexity per iteration}) = O(ND) + T * O(ND) = O(T * ND)$

In summary, the time complexity of MRBMO and RBMO are the same, both are  $O(T * ND)$ .

### 7.6. Workflow of MRBMO

The workflow of MRBMO is provided in Figure 8.

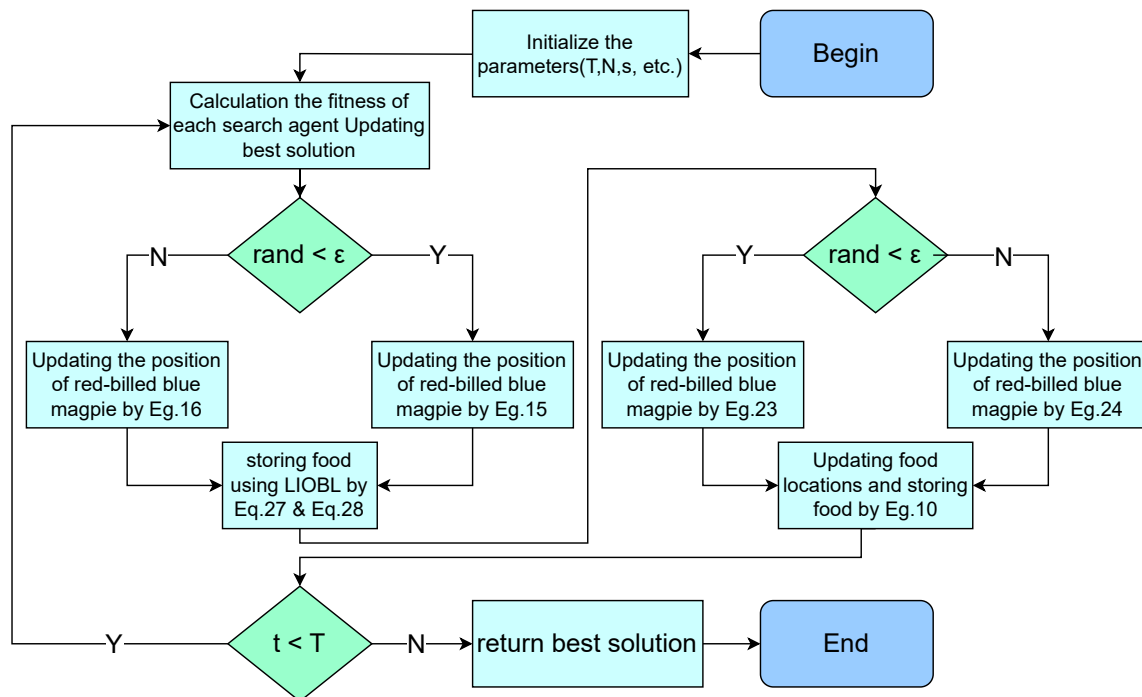


Figure 8. Workflow of MRBMO

## 8. Performance Test

The experimental environment for experiments was Windows 11 (64bit), Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz, 8GB running memory and the simulation platform is Matlab R2023a.

In order to validate the performance and effectiveness of MRBMO, the following two experiments are designed to test the algorithms on 23 classical benchmark functions and simulation experiment for engineering design optimization and antenna S-parameter optimization will be performed in next chapter:

- Each of the four improvement strategies is removed from MRBMO and an ablation study is performed on the 23 classical benchmark functions in Table 1;
- A qualitative analysis experiment was performed by applying MRBMO on the benchmark functions to comprehensively evaluate the performance, robustness and exploration-exploitation balance of MRBMO in different types of problems, by assessing convergence behavior, population diversity and exploration-exploitation capability;
- MRBMO, traditional RBMO and other outstanding metaheuristic algorithms are examined on the classical benchmark functions.

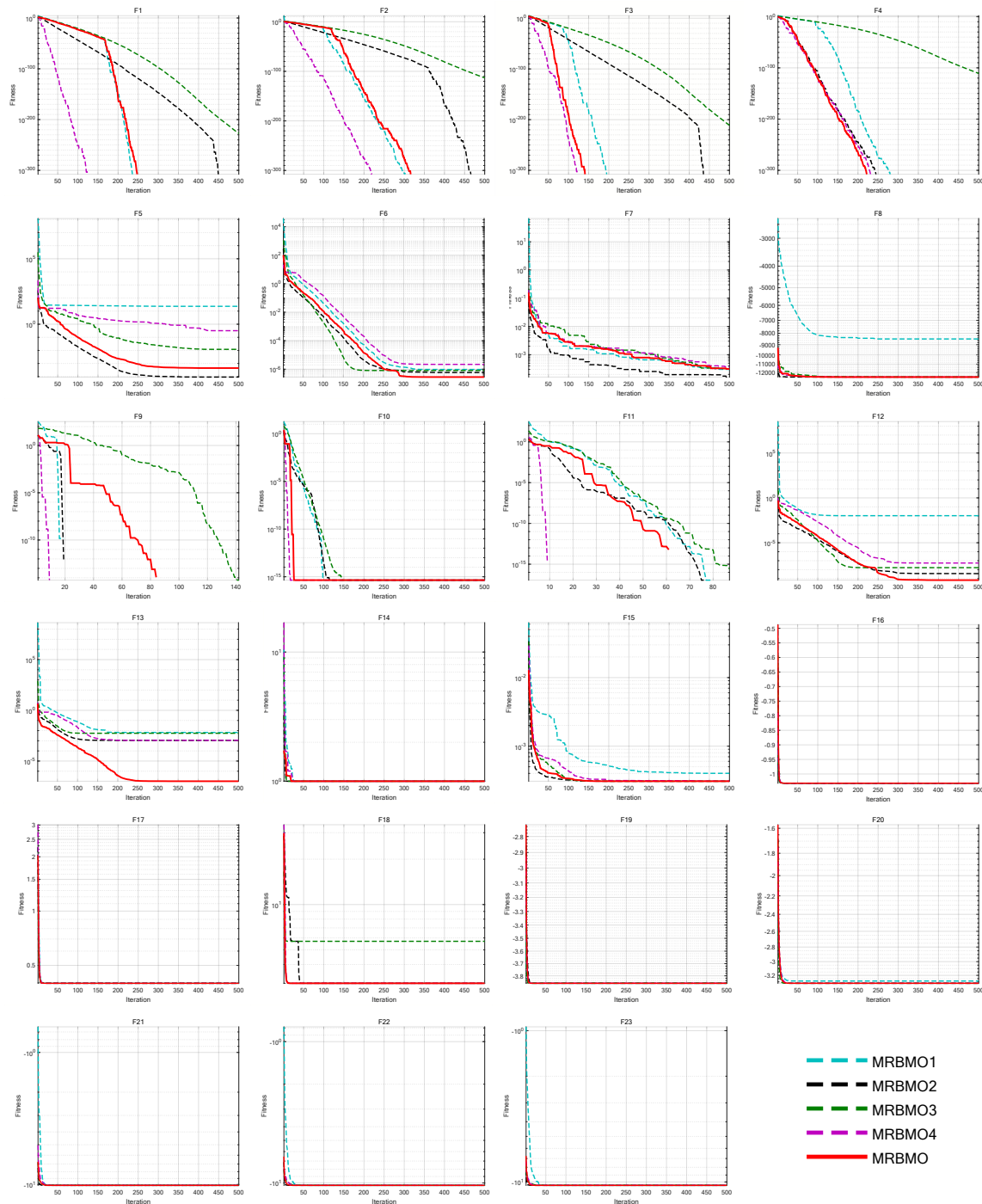
Table 1. Classical Benchmark Functions

Function	Function's Name	Type	Dimension	Best Value
F1	Sphere	Uni-modal	30	0
F2	Schwefel's Problem 2.22	Uni-modal	30	0
F3	Schwefel's Problem 1.2	Uni-modal	30	0
F4	Schwefel's Problem 2.21	Uni-modal	30	0
F5	Generalized Rosenbrock's Function	Uni-modal	30	0
F6	Step Function	Uni-modal	30	0
F7	Quartic Function	Uni-modal	30	0
F8	Generalized Schwefel's Function	Multi-modal	30	-12569.5
F9	Generalized Rastrigin's Function	Multi-modal	30	0
F10	Ackley's Function	Multi-modal	30	0
F11	Generalized Griewank's Function	Multi-modal	30	0
F12	Generalized Penalized Function 1	Multi-modal	30	0
F13	Generalized Penalized Function 2	Multi-modal	30	0
F14	Shekel's Foxholes Function	Multi-modal	2	0.998
F15	Kowalik's Function	Multi-modal	4	0.0003075
F16	Six-Hump Camel-Back Function	Composite	2	-1.0316
F17	Branin Function	Composite	2	0.398
F18	Goldstein-Price Function	Composite	2	3
F19	Hartman's Function 1	Composite	3	-3.8628
F20	Hartman's Function 2	Composite	6	-3.32
F21	Shekel's Function 1	Composite	4	-10.1532
F22	Shekel's Function 2	Composite	4	-10.4029
F23	Shekel's Function 3	Composite	4	-10.5364

8.1. Ablation Study

This paper designs an ablation study to evaluate the effectiveness of various improvement strategies on RBMO. We define the following variants: MRBMO1 is the MRBMO removes the Good Nodes Set initialization. MRBMO2 is the MRBMO which removes Enhanced Search-for-prey Strategy. MRBMO3 is the MRBMO which removes Siege-style Attacking-prey Strategy. And MRBMO4 is the MRBMO which removes LIOBL. To fairly compare the effectiveness of each strategy, we test these improved algorithms on 23 benchmark functions. We set the maximum iteration as  $T=500$  and the population size as  $N=30$ . We run each algorithm on the 23 functions for 30 times and the results are shown in Figure 9.

Experimental results show that each improvement strategy significantly enhances RBMO's performance. The Good Nodes Set Initialization distributes the population evenly in the solution space, improving quality and aiding in solving high-dimensional multi-modal functions like F5, F8, F12 and F13. As shown in F6, F7 and F13, the Enhanced Search-for-food Strategy sacrifices a slight reduction in convergence speed but improves the optimization ability for handling complex multi-modal functions. As shown in F1-F4 and F9-F13, the Siege-style Attacking-prey Strategy strengthens the exploitation phase, enhancing local search capability and convergence speed. Replacing the Food Storage mechanism with LIOBL allows population updates post-development, increasing diversity while preserving elite individuals, which boosts exploration and reduces local optima entrapment.



**Figure 9.** Iteration curves for MRBMOs in Ablation Study

## 8.2. Qualitative Analysis Experiment

In qualitative analysis experiment, we applied MRBMO on the benchmark functions recorded the search history of the red-billed blue magpie individuals, the exploration-exploitation percentage of MRBMO during the iterations and the population diversity of MRBMO. So that we could comprehensively evaluate the performance, robustness and exploration-exploitation balance of MRBMO in different types of problems.

In this experiment, the maximum number of iterations was set to  $T=500$  and the population size was  $N=30$ . The search history of the red-billed blue magpie individuals, the proportions of exploration and exploitation, population diversity, and iteration curves were recorded and are presented in Figure 10–13. From the figures, it is evident that the red-billed blue magpie individuals in MRBMO

demonstrate a well-distributed search within the solution space, indicating the effectiveness of the Good Nodes Set Initialization. For the F8, the global optimal solution is located in the upper-right corner of the solution space, posing significant challenges for the algorithm’s ability to escape local optima. The Siege-style Attacking-prey Strategy facilitates detailed exploration around the region with potential solution, ultimately leading to the identification of the optimal solution for F8. Additionally, the introduction of Levy flight allows MRBMO to consistently escape local optima and maintain high population diversity, even when addressing complex combinatorial problems such as F15-F23. For uni-modal functions, the results show that the exploitation proportion of MRBMO increases rapidly during the iterative process, demonstrating strong exploitation capabilities. For complex functions like F7, F8 and F15, the exploration proportion decreases gradually in the early iterations, reflecting MRBMO’s robust global exploration ability. In the later stages of the iterations, the exploitation proportion increases significantly, indicating strong local exploitation capabilities.

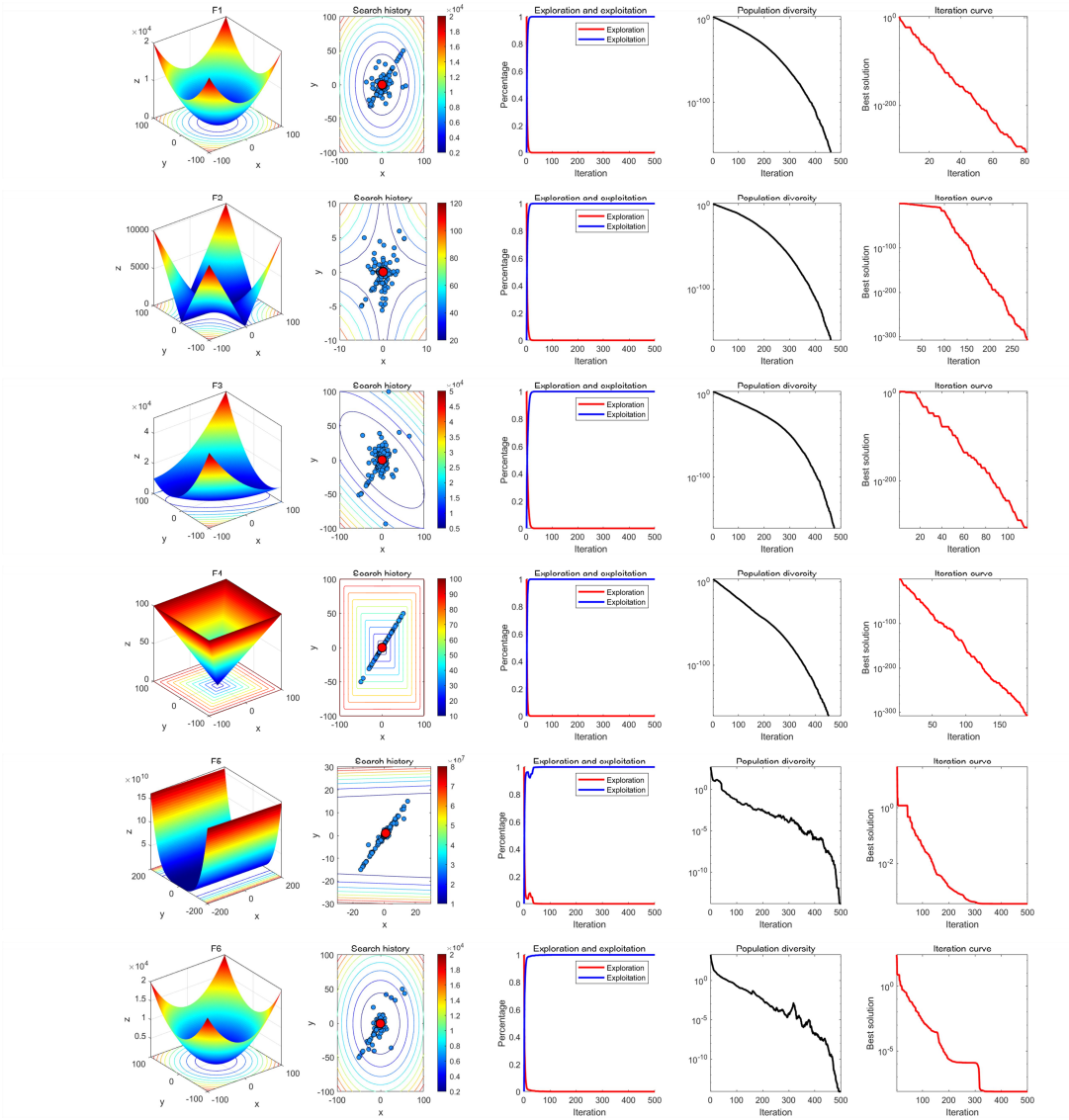


Figure 10. Performance of MRBMO in Qualitative analysis experiment (F1-F6)

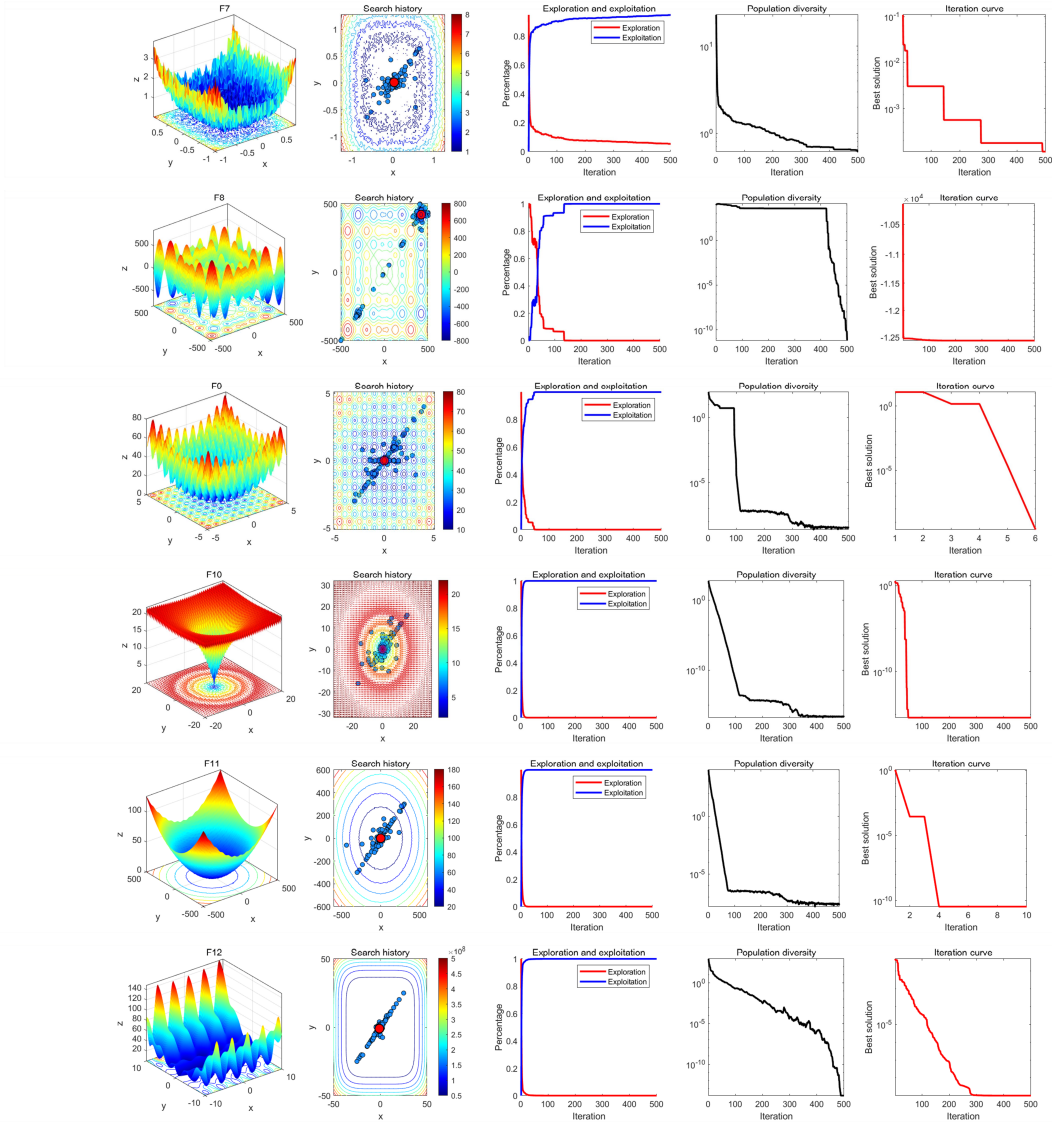


Figure 11. Performance of MRBMO in Qualitative analysis experiment (F7-F12)

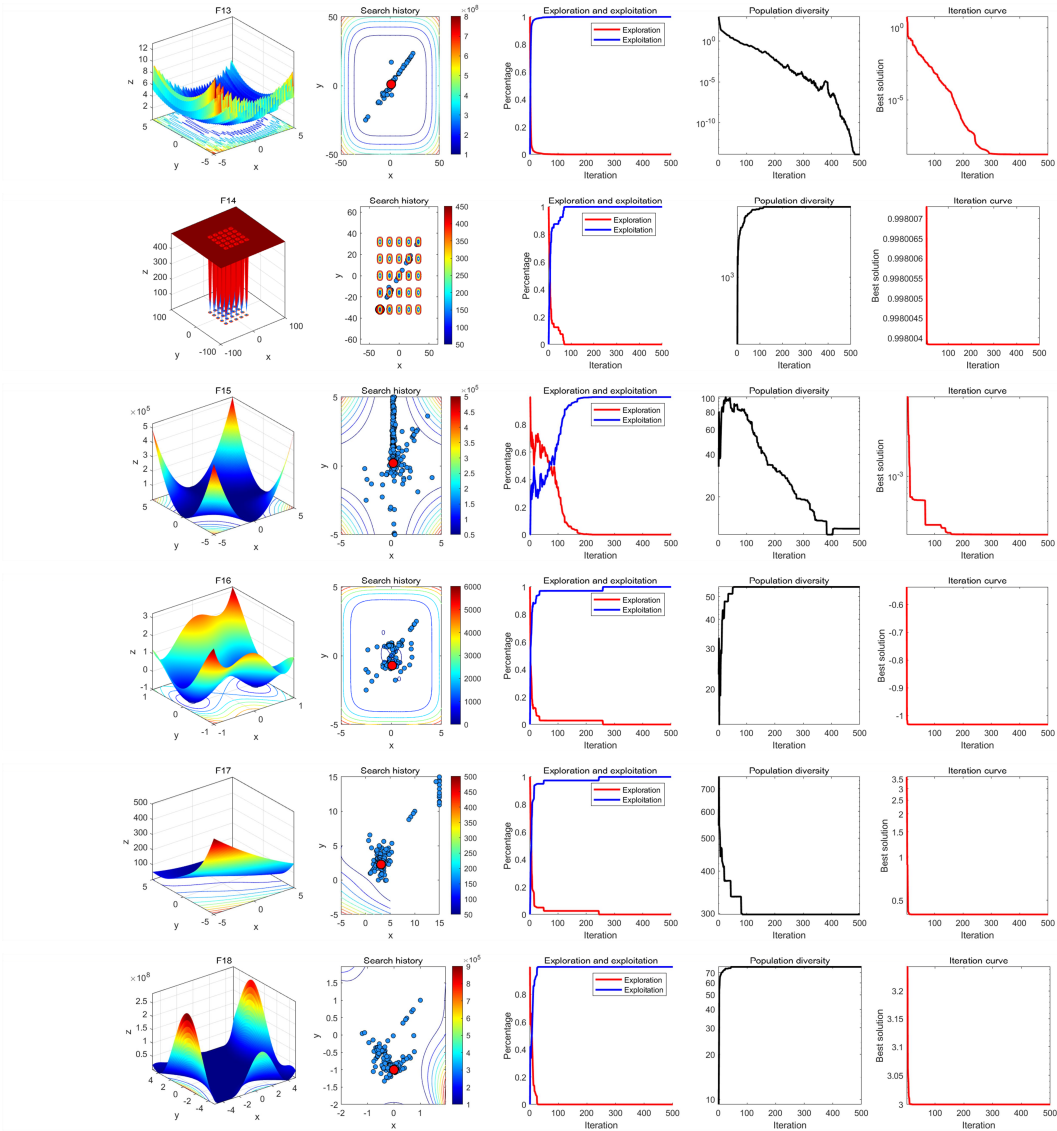
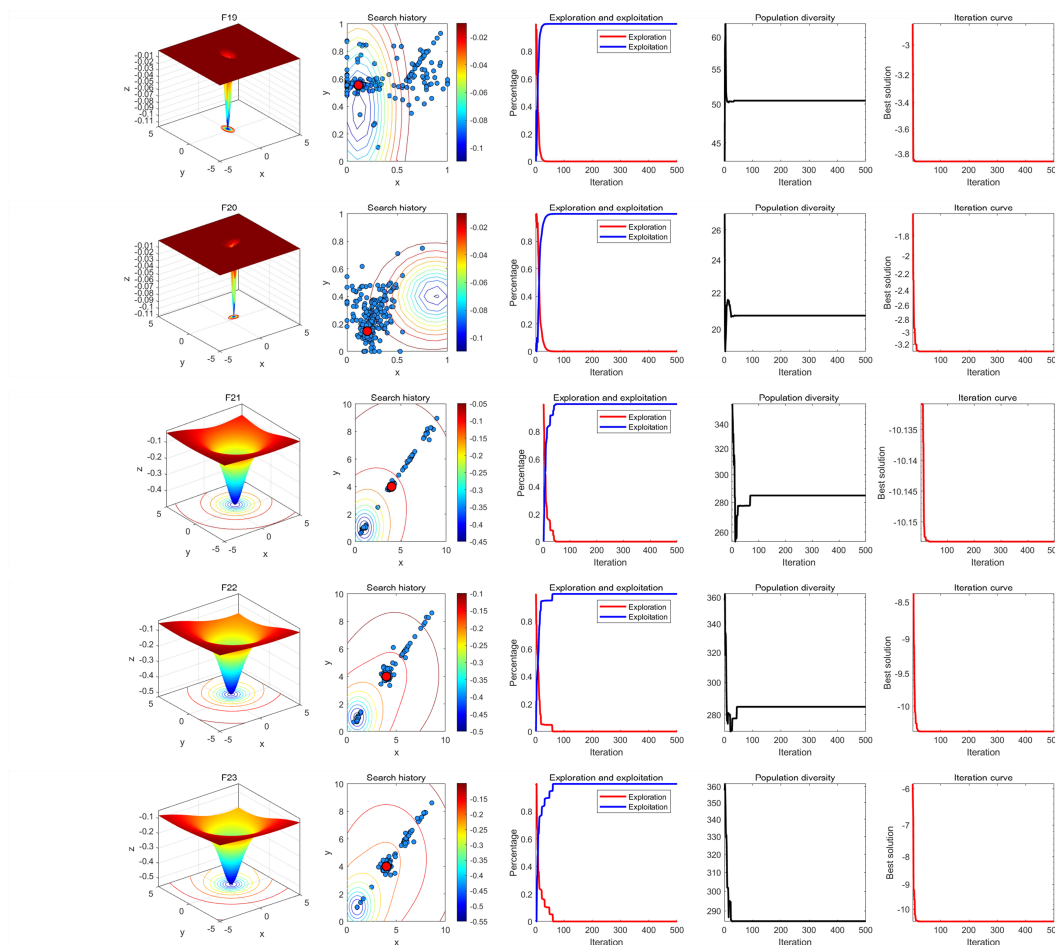


Figure 12. Performance of MRBMO in Qualitative analysis experiment (F13-F18)



**Figure 13.** Performance of MRBMO in Qualitative analysis experiment (F19-F23)

### 8.3. Superiority Comparative Test

To further verify the superiority of MRBMO, we selected Attraction-Repulsion Optimization Algorithm (AROA) [35], Harris Hawks Optimization (HHO) [7], Grey Wolf Optimizer (GWO) [10], Whale Optimization Algorithm (WOA) [21] and RBMO for superiority comparative experiments. The tests were conducted on 23 benchmark functions. The parameter settings for each algorithm are shown in Table 2. We set population size as  $N=30$ , the number of iterations as  $T=500$  and each algorithm is run independently 30 times. And the average fitness (Ave), standard deviation (Std),  $p$ -values of Wilcoxon rank-sum test, and Friedman values of 30 runs were recorded for performance analysis. And we will evaluate the overall effectiveness (OE) of MRBMO. The experimental results are presented in Figure 14, Table 3, Table 3 and Table 4.

From the experiment results, it can be seen that MRBMO converges to the optimal value on most functions, with a standard deviation of zero or close to zero, demonstrating strong stability, robustness and optimization capabilities. For problems like F5-F8 and F20-F23, which are prone to local optima, the Good Nodes Set initialization allows population of MRBMO to be evenly distributed in the solution space, significantly improving the population quality. As a result, MRBMO can escape local optima and achieve better solutions for these types of problems. The incorporation of Enhanced Search-for-food Strategy and Siege-style Attacking-prey Astrategy contributes to higher accuracy with solving complex problem like F5-F6 and F12-F13. Siege-style Attacking-prey Strategy helps MRBMO achieve higher convergence speed and accuracy, enabling MRBMO to find the optimal solutions for F1-F4 and F9-F11 within a limited number of iterations.

In non-parametric tests, the statistical results, as is shown in Table 4, indicate that most  $p$ -values from Wilcoxon rank-sum tests are less than 0.05, suggesting significant differences between the optimization results of MRBMO and the five comparison algorithms. On F9-F11, there is no significant

difference between MRBMO and HHO, because they all found the optimal solutions within a limited number of iterations. There is no significant difference between MRBMO and RBMO on F16, F17, and F19, because MRBMO not only found the optimal solutions for these functions every time but also had a standard deviation equal to or slightly smaller than that of RBMO. This indicates that MRBMO, while maintaining RBMO’s ability to solve complex functions, also achieved faster convergence speed and higher accuracy. And there is a significant difference between MRBMO and the rest algorithms. This experiment further corroborates the reliability of Superiority Test. The average Friedman values for these six algorithms are 5.2985, 3.4449, 3.3703, 3.8087, 3.7580, and 1.3196 respectively. According to these results, MRBMO ranks first in terms of the average Friedman value among the six algorithms, indicating its superior performance. This consistent performance across multiple functions highlights the effectiveness and robustness of MRBMO.

Table 2. Parameter setting for different algorithms

Algorithm	Parameters	Value
AROA	Attraction factor $c$	0.95
	Local search scaling factor 1	0.15
	Local search scaling factor 2	0.6
	Attraction probability 1	0.2
	Local search probability	0.8
	Expansion factor	0.4
	Local search threshold 1	0.9
	Local search threshold 2	0.85
	Local search threshold 3	0.9
HHO	Threshold	0.5
GWO	Convergence factor $a$	2 decreasing to 0
WOA	Spiral factor $b$	1
	Convergence factor $a$	2 decreasing to 0
RBMO	Balance coefficient $\varepsilon$	0.5
MRBMO	Balance coefficient $\varepsilon$	0.5
	Nonlinear factor $k$	1 decreasing to 0

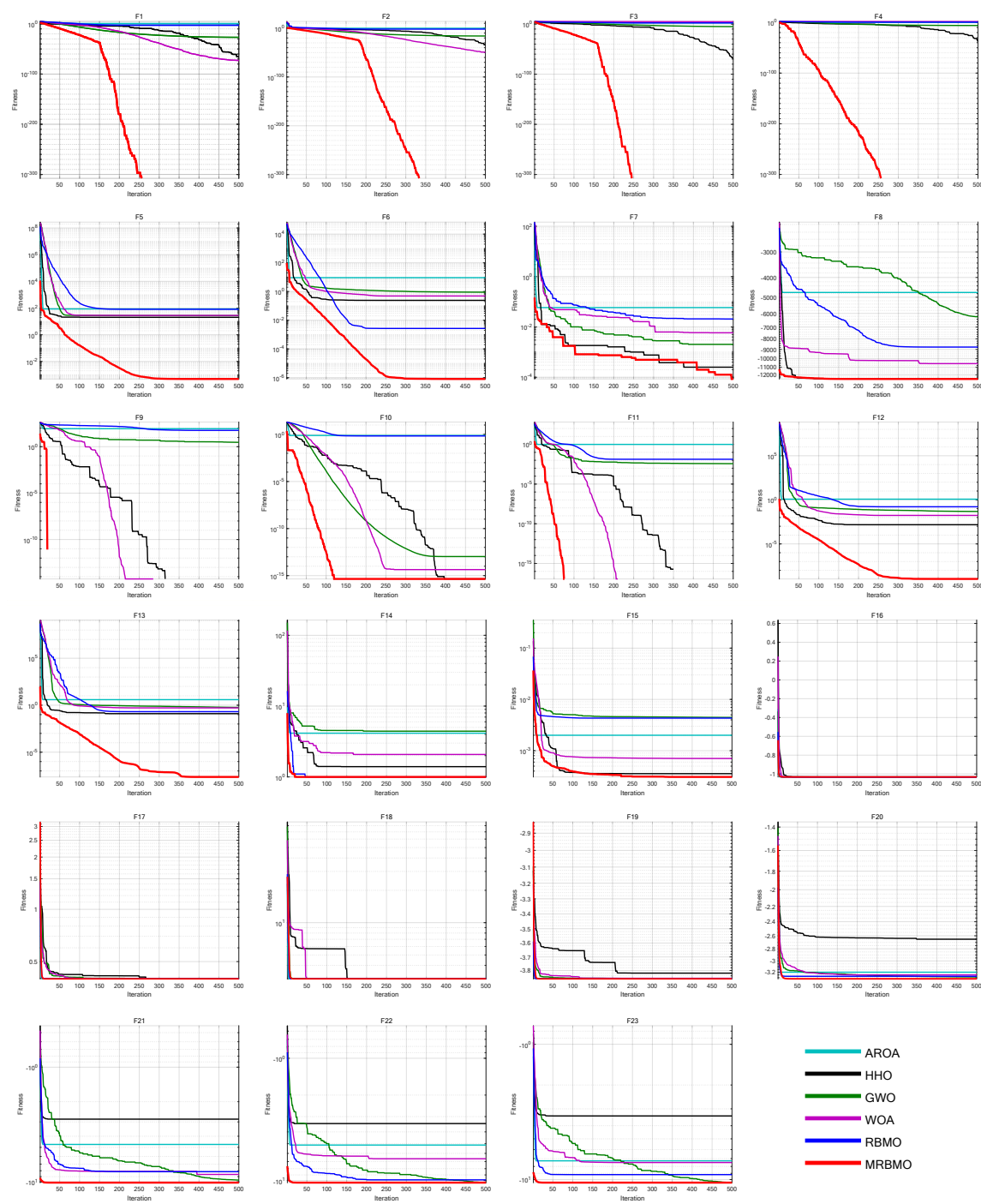


Figure 14. Iteration curves for each algorithm in Superiority comparative test

Table 3. Comparative results of each algorithm in Superiority test

Function	Index	AROA	HHO	GWO	WOA	RBMO	MRBMO
F1	Ave	3.3576E+00	1.1262E-74	1.4069E-27	3.1494E-72	2.5905E-03	0.0000E+00
	Std	2.1713E+00	4.1597E-74	2.4825E-27	1.7144E-71	4.0092E-03	0.0000E+00
F2	Ave	6.8236E-01	2.5359E-37	1.3021E-16	1.2999E-50	2.1738E-02	0.0000E+00
	Std	2.1966E-01	1.0551E-36	1.1792E-16	6.6296E-50	3.8140E-02	0.0000E+00
F3	Ave	2.3645E+02	1.7888E-70	3.8304E-05	4.5910E+04	2.0398E+02	0.0000E+00
	Std	2.4624E+02	9.2604E-70	1.3114E-04	1.3447E+04	1.5643E+02	0.0000E+00
F4	Ave	1.7294E+00	6.2931E-38	7.9187E-07	4.8866E+01	2.5584E+00	0.0000E+00
	Std	9.7121E-01	2.6195E-37	8.3113E-07	2.9838E+01	8.9898E-01	0.0000E+00
F5	Ave	7.6060E+01	1.6391E+01	2.6973E+01	2.7940E+01	7.4743E+01	2.6145E-04
	Std	3.7781E+01	1.4343E+01	6.6554E-01	4.9444E-01	6.8196E+01	2.6787E-04
F6	Ave	1.1738E+01	1.5028E-01	7.1766E-01	4.1328E-01	2.1586E-03	3.0078E-07
	Std	7.1777E+00	2.1271E-01	4.0762E-01	1.8796E-01	3.4735E-03	4.2633E-07
F7	Ave	2.8401E-02	1.1149E-04	1.6559E-03	3.8330E-03	1.8395E-02	9.1647E-05
	Std	1.8619E-02	1.4140E-04	9.3640E-04	3.7850E-03	7.9486E-03	7.5072E-05
F8	Ave	-4.5790E+03	-12569.3437	-5.9321E+03	-1.0617E+04	-8.5147E+03	-12569.4593
	Std	7.0213E+02	7.0213E+02	7.1362E+02	1.9688E+03	7.0735E+02	6.3822E-02
F9	Ave	5.6853E+01	0.0000E+00	3.9727E+00	3.7896E-15	4.7421E+01	0.0000E+00
	Std	6.9642E+01	0.0000E+00	4.2612E+00	1.4422E-14	1.4926E+01	0.0000E+00
F10	Ave	7.3713E-01	4.4409E-16	1.0075E-13	3.6415E-15	8.9531E-01	4.4409E-16
	Std	3.2507E-01	0.0000E+00	1.2260E-14	2.1580E-15	7.2365E-01	0.0000E+00
F11	Ave	1.0157E+00	0.0000E+00	4.6852E-03	2.3941E-02	1.8804E-02	0.0000E+00
	Std	6.9128E-02	0.0000E+00	7.4903E-03	7.5689E-02	2.1627E-02	0.0000E+00
F12	Ave	1.2551E+00	4.1197E-04	4.8043E-02	3.3640E-02	3.7537E-01	2.3563E-09
	Std	2.8262E-01	9.3011E-04	1.9237E-02	5.1397E-02	6.6152E-01	6.4604E-09

Table 3. Comparative results of each algorithm in Superiority test (Continued)

Function	Index	AROA	HHO	GWO	WOA	RBMO	MRBMO
F13	Ave	3.8771E+00	6.1086E-02	6.2003E-01	4.8201E-01	5.2915E-02	3.6628E-04
	Std	6.5630E-01	7.3467E-02	2.3731E-01	2.7243E-01	1.2411E-01	2.0060E-03
F14	Ave	4.7883E+00	1.6255E+00	5.4308E+00	3.1247E+00	1.0311E+00	9.9800E-01
	Std	4.1083E+00	1.1462E+00	4.6220E+00	3.3469E+00	1.8148E-01	8.2465E-17
F15	Ave	5.1259E-03	4.0325E-04	3.1023E-03	1.0325E-03	3.7112E-03	3.0755E-04
	Std	7.9967E-03	1.1820E-04	6.8916E-03	2.1999E-03	7.5779E-03	2.0577E-07
F16	Ave	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00
	Std	5.5337E-05	9.5165E-07	3.4423E-08	1.3538E-09	6.1849E-16	6.1158E-16
F17	Ave	3.9836E-01	3.9803E-01	3.9804E-01	3.9789E-01	3.9789E-01	3.9789E-01
	Std	1.6755E-03	6.1247E-04	8.2333E-04	9.5891E-06	0.0000E+00	0.0000E+00
F18	Ave	3.0025E+00	5.7004E+00	3.0000E+00	3.9001E+00	3.0000E+00	3.0000E+00
	Std	1.0330E-02	8.2387E+00	5.9458E-05	4.9295E+00	1.3848E-15	8.6883E-16
F19	Ave	-3.8577E+00	-3.7476E+00	-3.8619E+00	-3.8571E+00	-3.8628E+00	-3.8628E+00
	Std	9.2577E-03	2.6794E-01	1.8003E-03	9.7964E-03	2.7101E-15	2.7101E-15
F20	Ave	-3.2076E+00	-2.6245E+00	-3.2662E+00	-3.1990E+00	-3.2784E+00	-3.3180E+00
	Std	8.7053E-02	4.6584E-01	7.9583E-02	1.0112E-01	5.8273E-02	2.1707E-02
F21	Ave	-5.7049E+00	-2.9608E+00	-9.2237E+00	-9.1946E+00	-9.4044E+00	-1.0153E+01
	Std	3.3741E+00	1.6463E+00	2.1476E+00	2.4071E+00	2.2848E+00	6.7923E-15
F22	Ave	-6.4097E+00	-3.3593E+00	-1.0401E+01	-6.8982E+00	-8.4732E+00	-1.0403E+01
	Std	3.1860E+00	1.6170E+00	1.4716E-03	3.1916E+00	3.0381E+00	1.1427E-15
F23	Ave	-5.7600E+00	-3.1059E+00	-1.0264E+01	-7.3546E+00	-9.6555E+00	-1.0536E+01
	Std	3.4452E+00	1.2479E+00	1.4812E+00	3.3466E+00	2.3046E+00	1.8366E-15

**Table 4.** Ranking of non-parametric tests of different algorithms

Algorithm	Average Friedman Value	Rank	+/-/-
AROA	5.2985	6	23/0/0
HHO	3.4449	3	20/3/0
GWO	3.3703	2	23/0/0
WOA	3.8087	5	23/0/0
RBMO	3.7580	4	20/2/1
MRBMO	1.3196	1	-

Table 5 summarizes all performance results of MRBMO and other algorithms by a useful metric named overall effectiveness (OE). In Table 5, *w* indicates win, *t* indicates tie and *l* indicates loss. The OE of each algorithm is computed by Eq. 29 [33].

$$OE = \frac{N - L}{L} \cdot 100 \quad (29)$$

where *N* is the total number of tests; *L* is the total number of losing tests for each algorithm.

Results proves that MRBMO is competitive with other algorithms on the benchmark functions with different dimensions.

**Table 5.** Effectiveness of MRBMO and other algorithms

	AROA (w/t/l)	HHO (w/t/l)	GWO (w/t/l)	WOA (w/t/l)	RBMO (w/t/l)	MRBMO (w/t/l)
Total	0/0/23	0/2/21	0/0/23	0/0/23	1/2/20	18/5/1
OE	0%	8.69%	0%	0%	13.04%	95.65%

## 9. Silulation Experiment

To validate the ability of MRBMO to solve real-world problems, we used four engineering design optimization problems to test the performance of MRBMO, in order to verify the effectiveness and applicability of MRBMO in engineering design optimization. We also used an antenna S-parameter suite to test the performance of MRBMO, in order to quickly validate the effectiveness and applicability of MRBMO in antenna S-parameter optimization.

### 9.1. Engineering Design Optimzation

#### 9.1.1. Pressure Vessel Design

A pressure vessel is a common mechanical structure used in fields such as chemical engineering, aerospace, and medical applications. The pressure vessel design problem is a classic structural optimization problem, where the goal is to minimize the manufacturing costs of the pressure vessel, including pairing, forming, and welding processes. The design of the pressure vessel is shown in Figure 15, with caps sealing both ends of the vessel. The cap at one end is hemispherical.  $x_1$  and  $x_2$  represent the wall thickness of the cylindrical section and the head, respectively, while  $x_3$  is the inner diameter of the cylindrical section, and  $x_4$  is the length of the cylindrical section, excluding the head. Thus,  $x_1$ ,  $x_2$ ,  $x_3$ , and  $x_4$  are the four optimization variables of the pressure vessel design problem. The objective function and four optimization constraints are represented as follows:

Variable:

$$x = [x_1, x_2, x_3, x_4]$$

Minimize:

$$f(x) = 0.6224x_1 \cdot x_3 \cdot x_4 + 1.7781x_2 \cdot x_3^2 + 3.1661x_1^2 \cdot x_4 + 19.84x_1^2 \cdot x_3 \quad (30)$$

Subject to:

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0; \quad (31)$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0; \quad (32)$$

$$g_3(x) = -\pi x_3^2 \cdot x_4 - \frac{4}{3}\pi x_3^2 + 1296000 \leq 0; \quad (33)$$

$$g_4(x) = x_4 - 240 \leq 0; \quad (34)$$

Variable range:

$$0 \leq x_1 \leq 99; \quad 0 \leq x_2 \leq 99; \quad 10 \leq x_3 \leq 200; \quad 10 \leq x_4 \leq 99;$$

In this study, we conducted comparative tests between the MRBMO and AROA, HHO, GWO, WOA and RBMO. The parameter settings for each algorithm are provided in Table 2. The number of iterations is set to  $T=500$ , and the population size is  $N=30$ . Each algorithm is run independently for 30 trials on the pressure vessel design problem, and the average values and standard deviations are recorded for performance analysis. The experimental results are shown in Figure 16 and Table 6. As shown in Table 6, MRBMO significantly outperforms the other algorithms in terms of both optimization accuracy and stability for the pressure vessel design problem. This demonstrates that MRBMO has superior solving capabilities when handling this type of problem.

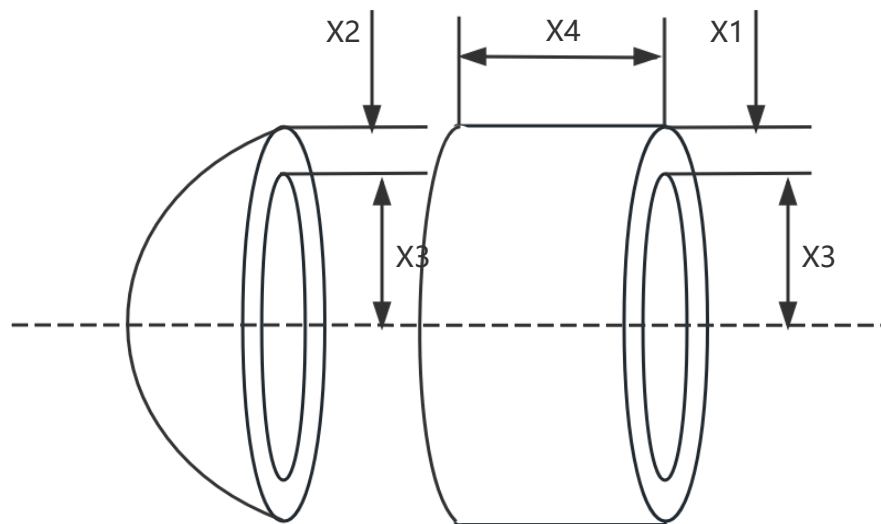


Figure 15. The structure of a Pressure Vessel

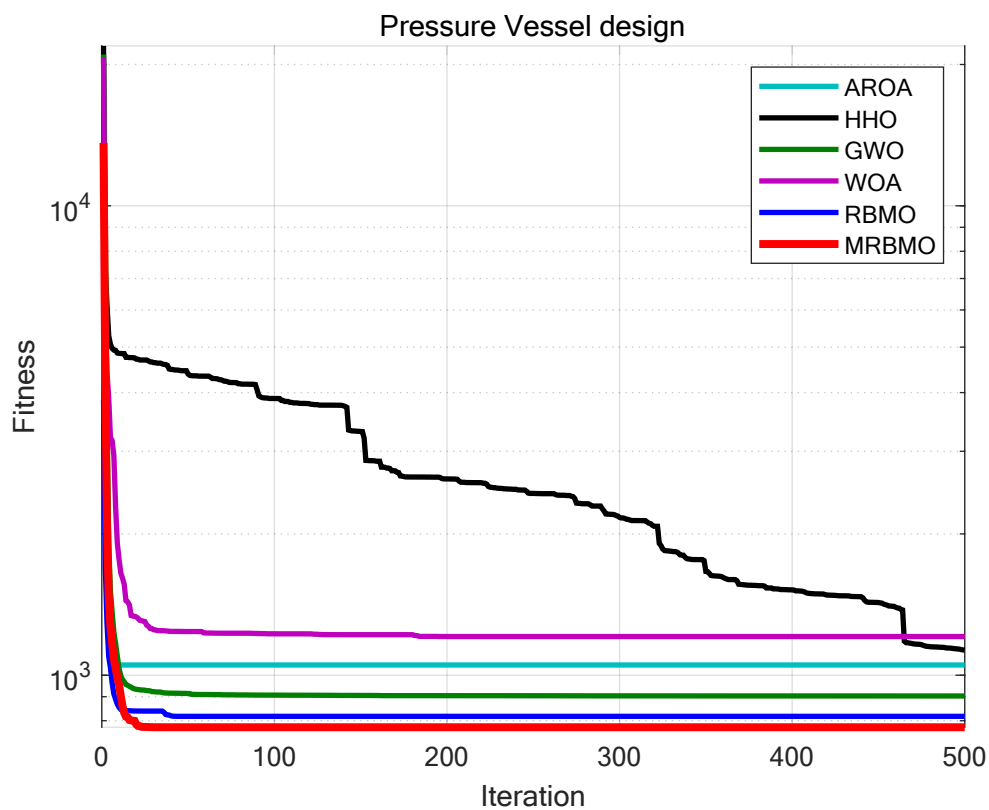


Figure 16. Iteration curves in Pressure Vessel design

### 9.1.2. Piston Lever Design

A piston lever is a typical mechanical structure, as shown in Figure 17, and its design problem is classified as a classical engineering optimization problem. It involves the adjustment of multiple geometric and mechanical parameters with the aim of minimizing material usage or structural weight while satisfying constraints such as strength and stability. This optimization seeks to achieve a balance between economic efficiency and structural performance and is widely applied in mechanical engineering, vehicle design, and other industrial scenarios, especially in lightweight and efficient design of moving components.

In the Piston Lever design problem, the objective is to minimize the total material consumption of the Piston Lever while ensuring that the structural strength and performance meet design requirements. The geometric structure of the Piston Lever is defined by multiple design parameters that describe the relationships between its key dimensions. In this problem, the Piston Lever consists of multiple structural parts with the following important characteristics: one end is fixed, while the other end bears an applied force; its mechanical properties are influenced by geometric features such as radius and length; and the design of each geometric part is controlled by decision variables. From the geometric relationships, the meanings of variables  $x_1$  to  $x_4$  are as follows:  $x_1$  and  $x_2$  are Primary length and width parameters of the geometric structure, which govern the overall lever arm.  $x_3$  is cross-sectional radius at the point of force application, affecting force distribution.  $x_4$  is Geometric dimension related to the support point.

The objective function for the Piston Lever design problem can be described as,

Variable:

$$x = [x_1, x_2, x_3, x_4]$$

Minimize:

$$f(x) = 0.25\pi x_3^2(L_2 - L_1) \quad (35)$$

Subject to:

$$g_1(x) = QL \cos \theta - RF \leq 0; \quad (36)$$

$$g_2(x) = Q(L - x_4) - M_{\max} \leq 0; \quad (37)$$

$$g_3(x) = 1.2(L_2 - L_1) - L_1 \leq 0; \quad (38)$$

$$g_4(x) = \frac{x_3}{2} - x_2 \leq 0; \quad (39)$$

Variable range:

$$0.05 \leq x_1 \leq 500; \quad 0.05 \leq x_2 \leq 500; \quad 0.05 \leq x_4 \leq 500; \quad 0.05 \leq x_3 \leq 120;$$

Where:

$$Q = 10000; \quad P = 1500; \quad L = 240; \quad M_{\max} = 1.8 \times 10^6;$$

$$L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2}; \quad L_2 = \sqrt{(x_4 \sin \theta + x_1)^2 + (x_2 - x_4 \cos \theta)^2};$$

$$R = \frac{|-x_4(x_4 \sin \theta + x_1) + x_1(x_2 - x_4 \cos \theta)|}{\sqrt{(x_4 - x_2)^2 + x_1^2}};$$

$$F = 0.25\pi P x_3^2;$$

This study compares MRBMO with AROA, HHO, GWO, WOA and RBMO, with the parameter settings for each algorithm shown in Table 2. The number of iterations is uniformly set to  $T=500$ , and the population size is set to  $N=30$ . Each algorithm is run independently 30 times on the Piston Lever design problem, with the average fitness and standard deviation recorded for performance analysis. The experimental results are shown in Figure 18 and Table 6. As shown in Table 6, in the Piston Lever design problem, MRBMO demonstrates significantly superior optimization accuracy and stability compared to other algorithms. This indicates that MRBMO has a substantial advantage in handling such problems.

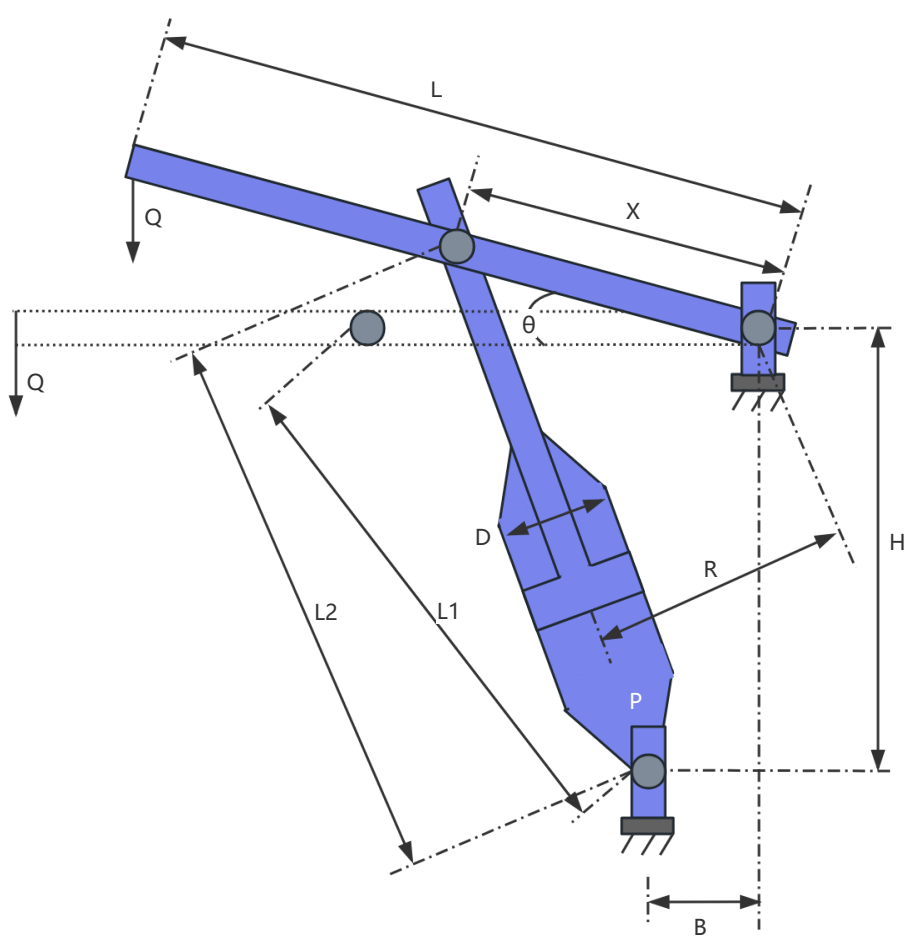


Figure 17. Structure of a piston lever

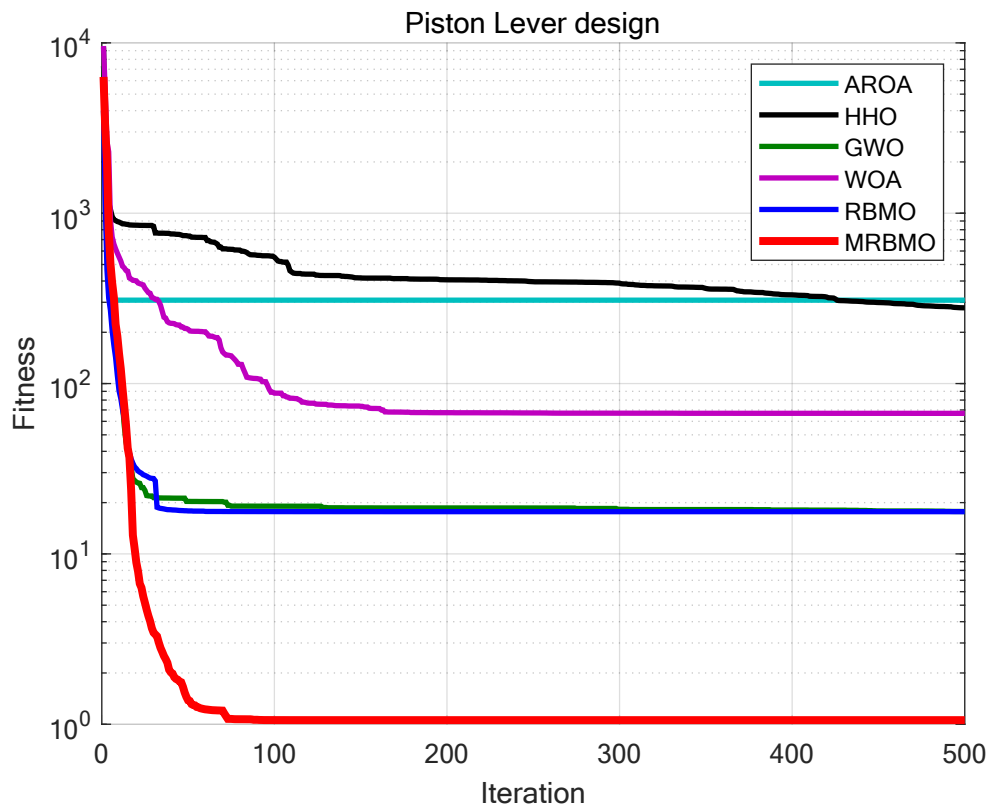


Figure 18. Iteration curves of the algorithms in Piston Lever design problem

### 9.1.3. Robot Gripper Design

The robot gripper design problem is a classic engineering optimization problem, widely applied in industrial automation, medical robotics, and logistics. The goal is to maximize gripping performance or minimize material usage under the constraints of gripping force range, structural requirements, and geometric stability, thereby optimizing the structural efficiency and cost-effectiveness of the robot gripper. Figure 19 is the structure of a robot gripper.

The robot gripper involves several critical parameters related to geometry, mechanics, and motion:  $x_1, x_2, x_3, x_4$  are geometric parameters of the gripper;  $x_5$  is the force applied to the gripper;  $x_6$  is the length of the gripper;  $x_7$  is the angular offset of the gripper.

The objective function of the Robot Gripper design problem can be described as:

Variable:

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] \quad (40)$$

Minimize:

$$f(x) = - \left[ \max_{z \in [0, Z_{\max}]} F_1(x, z, 2) + \min_{z \in [0, Z_{\max}]} F_1(x, z, 2) \right] \quad (41)$$

Subject to:

$$g_1(x) = -Y_{\min} + F_1(x, Z_{\max}, 1) \leq 0; \quad (42)$$

$$g_2 = -F_1(x, Z_{\max}, 1) \leq 0; \quad (43)$$

$$g_3(x) = Y_{\max} - F_1(x, 0, 1) \leq 0; \quad (44)$$

$$g_4(x) = F_1(x, 0, 1) - Y_G \leq 0; \quad (45)$$

$$g_5(x) = x_6^2 + x_4^2 - (x_1 + x_2)^2 \leq 0; \quad (46)$$

$$g_6(x) = x_2^2 - (x_1 - x_4)^2 - (x_6 - Z_{\max})^2 \leq 0; \quad (47)$$

$$g_7(x) = Z_{\max} - x_6 \leq 0 \quad (48)$$

Variable range:

$$10 \leq x_1 \leq 150; \quad 10 \leq x_2 \leq 150; \quad 100 \leq x_3 \leq 200; \quad 0 \leq x_4 \leq 50;$$

$$10 \leq x_5 \leq 150; \quad 100 \leq x_6 \leq 300; \quad 1 \leq x_7 \leq 3.14$$

Where:

$$P = 100; \quad Z_{\max} = 100;$$

$$Y_{\min} = 50; \quad Y_{\max} = 100; \quad Y_G = 150;$$

$F_1(x, z, flag)$  : Calculate the grabbing force or the applied force

When Flag=1, calculate the grabbing force:

$$F_1(x, z, 1) = 2(x_5 + x_4 + x_3 \cdot \sin(\beta + x_7));$$

When Flag=2, calculate the applied force:

$$F_1(x, z, 2) = \frac{Px_2 \cdot \sin(\alpha + \beta)}{2x_3};$$

$$\alpha = \arccos\left(\frac{x_1^2 + g^2 - x_2^2}{2x_1g}\right) + \phi_o;$$

$$\beta = \arccos\left(\frac{x_2^2 + g^2 - x_1^2}{2x_2g}\right) - \phi_o;$$

$$g = \sqrt{x_4^2 + (z - x_6)^2};$$

$$\phi_o = \arctan\left(\frac{x_4}{x_6 - z}\right)$$

This study compares MRBMO with AROA, HHO, GWO, WOA, and RBMO. The parameter settings for each algorithm are shown in Table 2. The iteration number  $T=500$  and population size  $N=30$  were uniformly set for all experiments. Each algorithm was executed independently 30 times on the robot gripper design problem, and the average fitness value and standard deviation were recorded for performance analysis. The experimental results are presented in Figure 20 and Table 6.

From Table 6, it can be observed that in the Robot Gripper design problem, MRBMO significantly outperforms other algorithms in terms of optimization accuracy and stability. This demonstrates that MRBMO has a considerable advantage in addressing such problems..

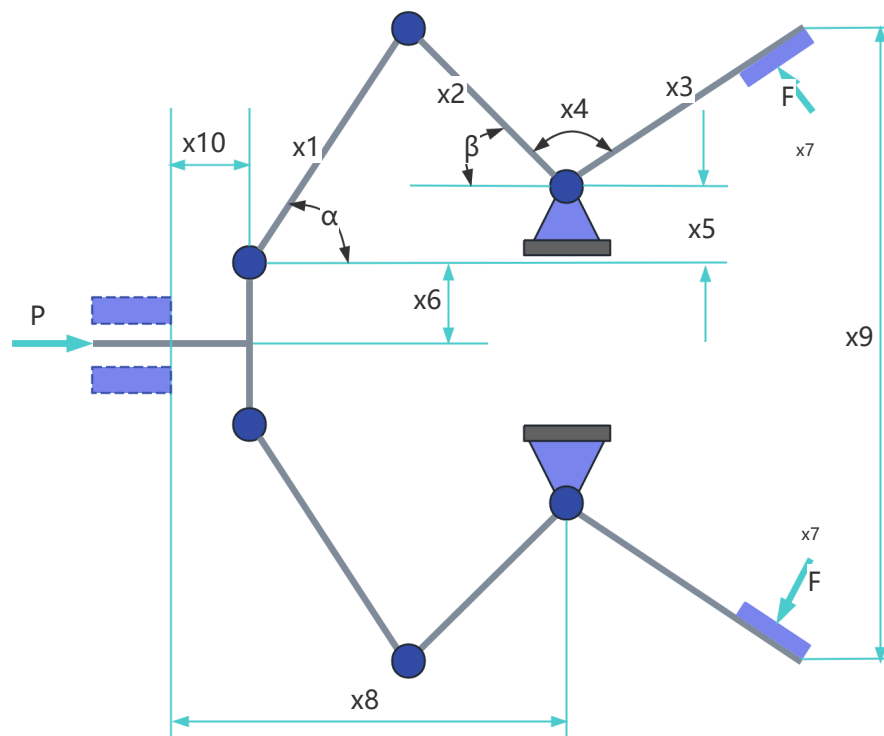


Figure 19. Robot Gripper design problem

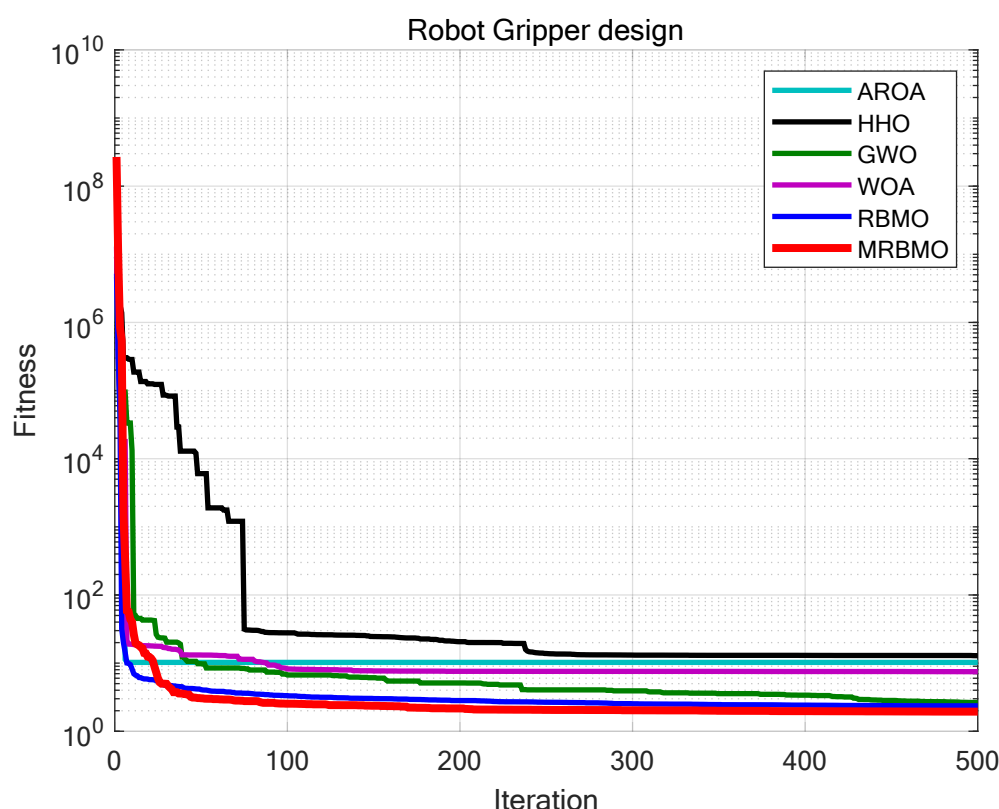


Figure 20. Iteration curves on solving Robot Gripper design problem

### 9.2. Industrial Refrigeration System Design

In the chemical plant design, an industrial refrigeration system is one of the key auxiliary facilities, widely used in chemical production processes, especially in operations such as chemical reactions, storage, transportation, and refining, where temperature control and heat exchange are critical. Chemical plants often require significant cooling and temperature control to maintain reaction stability, ensure product quality, reduce energy consumption and emissions, and ensure the proper functioning of equipment. Therefore, industrial refrigeration systems play a crucial role in the design of chemical plants. The industrial refrigeration system design problem focuses on minimizing energy consumption and cost while ensuring efficient cooling performance, as shown in Figure 21. The objective is to configure the system components, such as compressors, condensers, and evaporators, to achieve the lowest operating cost and optimal heat exchange efficiency. The problem includes fourteen variables: compressor power  $x_1$  and  $x_2$ , refrigerant flow rate and mass flow  $x_3$  through  $x_6$ , characteristics of the condenser and evaporator  $x_7$  and  $x_8$ , compression ratios  $x_9$  and  $x_{10}$ , temperature parameters  $x_{11}$  and  $x_{12}$ , and flow rate parameters  $x_{13}$  and  $x_{14}$ . Specifically, compressor power  $x_1$  and  $x_2$  control the cooling capacity; refrigerant flow rate and mass flow  $x_3$  through  $x_6$  indicate the refrigerant flow through condensers, evaporators, and receivers;  $x_7$  and  $x_8$  represent the sizing parameters of the condenser and evaporator;  $x_9$  and  $x_{10}$  define the compression degree and compressor efficiency;  $x_{11}$  and  $x_{12}$  manage the temperature differential for heat exchange; and  $x_{13}$  and  $x_{14}$  govern the flow rate of cooling water or refrigerant, affecting overall system performance. Industrial refrigeration system design problem is modeled below.

Variable:

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}]$$

Minimize:

$$y = f(x) \quad (49)$$

Subject to:

$$g_1 = \frac{1.524}{x_7} - 1 \leq 0; \quad (50)$$

$$g_2 = \frac{1.524}{x_8} - 1 \leq 0; \quad (51)$$

$$g_3 = 0.07789 \cdot x_1 - \frac{2 \cdot x_9}{x_7} - 1 \leq 0; \quad (52)$$

$$g_4 = \frac{7.05305 \cdot x_1^2 \cdot x_{10}}{x_9 \cdot x_8 \cdot x_2 \cdot x_{14}} - 1 \leq 0; \quad (53)$$

$$g_5 = \frac{0.0833 \cdot x_{14}}{x_{13}} - 1 \leq 0; \quad (54)$$

$$g_6 = \frac{47.136 \cdot x_2^{0.333} \cdot x_{12}}{x_{10}} - 1.333 \cdot x_8 \cdot x_{13}^{2.1195} + \frac{62.08 \cdot x_{13}^{2.1195} \cdot x_8^{0.2}}{x_{12} \cdot x_{10}} - 1 \leq 0; \quad (55)$$

$$g_7 = 0.04771 \cdot x_{10} \cdot x_8^{1.8812} \cdot x_{12}^{0.3424} - 1 \leq 0; \quad (56)$$

$$g_8 = 0.0488 \cdot x_9 \cdot x_7^{1.893} \cdot x_{11}^{0.316} - 1 \leq 0; \quad (57)$$

$$g_9 = \frac{0.0099 \cdot x_1}{x_3} - 1 \leq 0; \quad (58)$$

$$g_{10} = \frac{0.0193 \cdot x_2}{x_4} - 1 \leq 0; \quad (59)$$

$$g_{11} = \frac{0.0298 \cdot x_1}{x_5} - 1 \leq 0; \quad (60)$$

$$g_{12} = \frac{0.056 \cdot x_2}{x_6} - 1 \leq 0; \quad (61)$$

$$g_{13} = \frac{2}{x_9} - 1 \leq 0; \quad (62)$$

$$g_{14} = \frac{2}{x_{10}} - 1 \leq 0; \quad (63)$$

$$g_{15} = \frac{x_{12}}{x_{11}} - 1 \leq 0; \quad (64)$$

Where:

$$f(x) = 63098.88 \cdot x_2 \cdot x_4 \cdot x_{12} + 5441.5 \cdot x_2^2 \cdot x_{12} + 115055.5 \cdot x_2^{1.664} \cdot x_6$$

$$+ 6172.27 \cdot x_2^2 \cdot x_6 + 63098.88 \cdot x_1 \cdot x_3 \cdot x_{11} + 5441.5 \cdot x_1^2 \cdot x_{11}$$

$$+ 115055.5 \cdot x_1^{1.664} \cdot x_5 + 6172.27 \cdot x_1^2 \cdot x_5 + 140.53 \cdot x_1 \cdot x_{11}$$

$$\begin{aligned}
& +281.29 \cdot x_3 \cdot x_{11} + 70.26 \cdot x_1^2 + 281.29 \cdot x_1 \cdot x_3 + 281.29 \cdot x_3^2 \\
& +14437 \cdot x_8^{1.8812} \cdot x_{12}^{0.3424} \cdot x_{10} \cdot x_1^2 \cdot \frac{x_7}{x_{14} \cdot x_9} \\
& +20470.2 \cdot x_7^{2.893} \cdot x_{11}^{0.316} \cdot x_{12}
\end{aligned}$$

Variable range:

$$0.001 < x_1 < 5; \quad 0.001 < x_2 < 5; \quad 0.001 < x_3 < 5; \quad 0.001 < x_4 < 5;$$

$$0.001 < x_5 < 5; \quad 0.001 < x_6 < 5; \quad 0.001 < x_7 < 5; \quad 0.001 < x_8 < 5;$$

$$0.001 < x_9 < 5; \quad 0.001 < x_{10} < 5; \quad 0.001 < x_{11} < 5; \quad 0.001 < x_{12} < 5;$$

$$0.001 < x_{13} < 5; \quad 0.001 < x_{14} < 5;$$

A comparative test was also conducted between MRBMO and AROA, HHO, GWO, WOA, and RBMO, with parameters as specified in Table 2. Each algorithm ran independently for 30 trials, maintaining a maximum iterations of  $T=500$  and a population size of  $N=30$ . The experimental results, presented in Figure 22 and Table 6. The results demonstrate that MRBMO consistently escapes local optima, continuously searching for better solutions even when other algorithms are trapped in sub-optimal states. Compared with other algorithms, MRBMO shows exceptional stability and accuracy in solution-seeking. Therefore, MRBMO proves to be a highly robust and outstanding optimization tool for handling complex design optimization tasks.

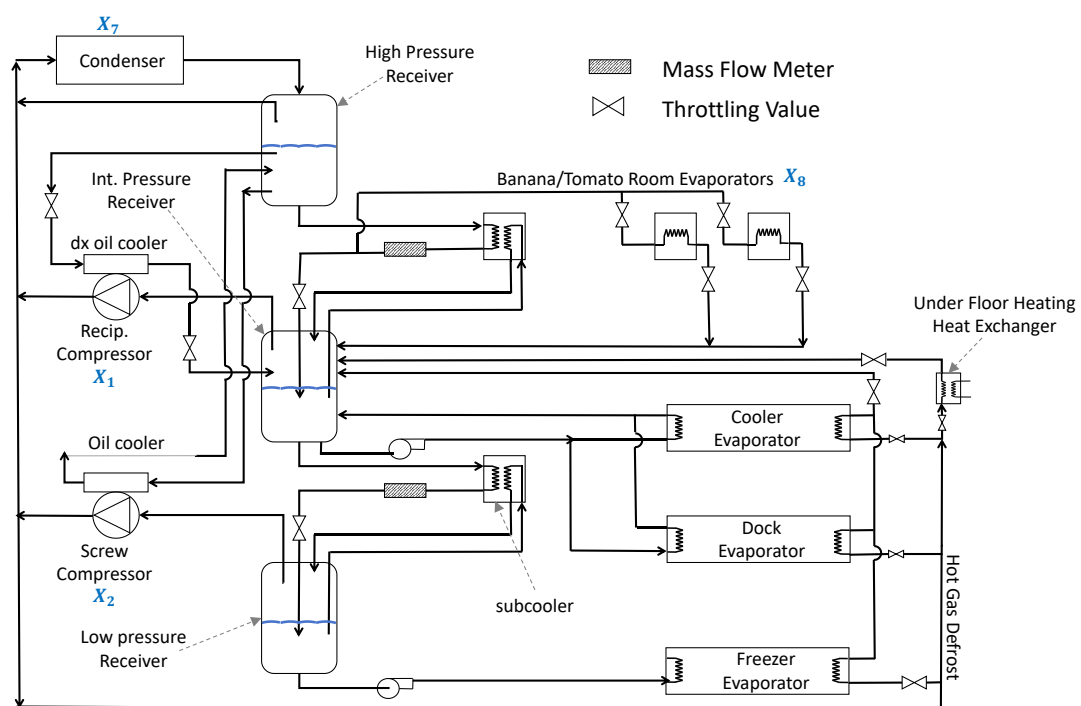


Figure 21. The structure of an industrial refrigeration system

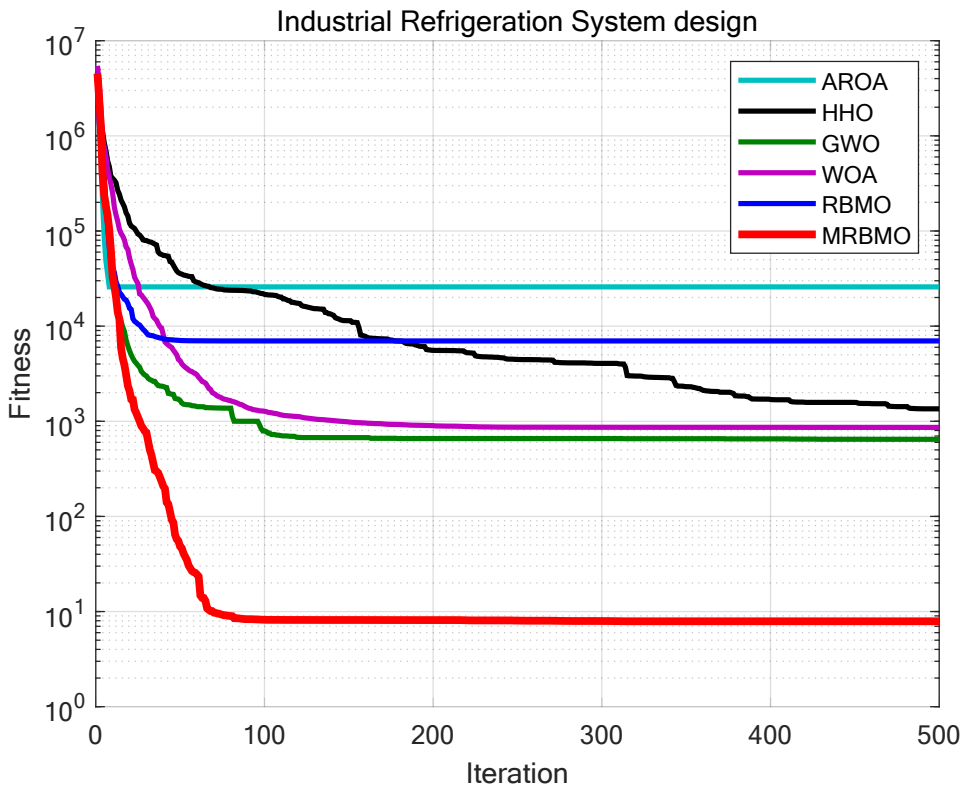


Figure 22. Iteration curves of the algorithms in Industrial Refrigeration System design problem

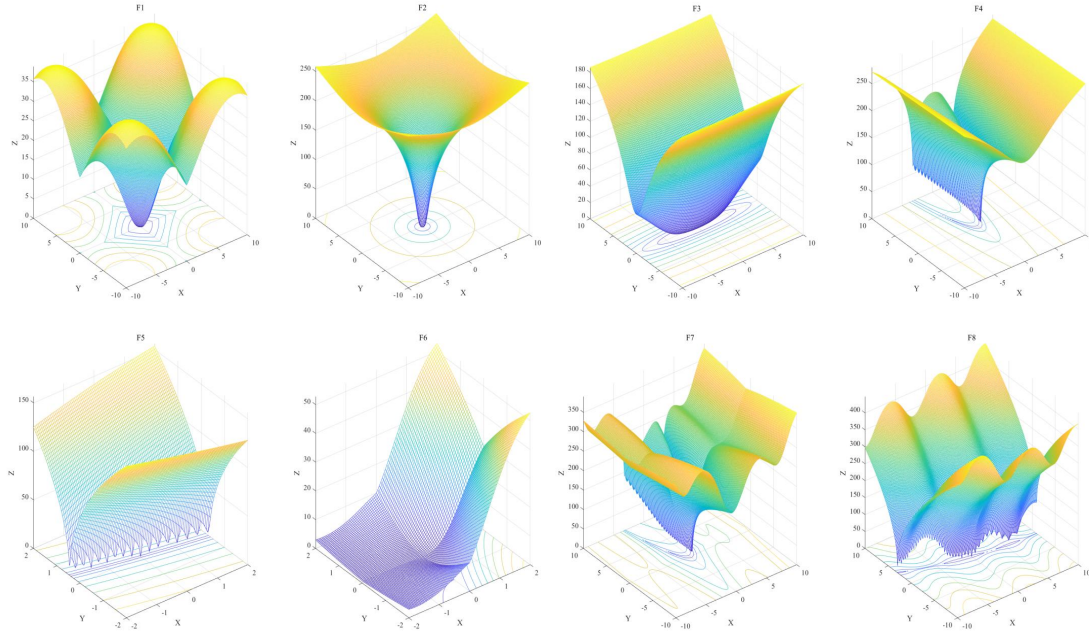
Table 6. Results of different algorithms on various engineering design optimization problems

Problems	Index	AROA	HHO	GWO	WOA	RBMO	MRBMO
Pressure vessel	Ave	1051.497457	1131.127862	903.228152	1209.015966	817.427522	774.810116
	Std	284.348170	279.641639	275.992023	547.039872	195.056827	96.712573
Piston lever	Ave	237.669131	374.504869	51.098610	41.949668	17.698731	1.057175
	Std	165.924633	203.438416	80.569535	83.949697	52.625219	0.000000
Robot gripper	Ave	10.245274	12.850251	2.634413	7.488088	2.360270	1.940320
	Std	10.434973	23.741498	1.835745	18.086681	0.833777	0.549999
Industrial refrigeration system	Ave	25855.679209	1351.066268	646.177525	861.821754	6987.501406	7.900666
	Std	19798.951282	4597.977707	3496.391366	4161.740403	9330.060337	0.814027

9.3. Antenna S-Parameter Optimization

The optimization of antenna S-parameters (scattering parameters) is a critical aspect of the design of wireless communication systems, radar, and other electronic devices, serving as a key factor in ensuring the efficient operation of wireless systems. S-parameters describe the reflection and transmission characteristics of antennas, primarily including  $S_{11}$  (reflection coefficient) and  $S_{21}$  (transmission coefficient). Optimizing these parameters can enhance antenna performance, reduce signal loss, and improve radiation efficiency. metaheuristic algorithms are capable of finding optimal or near-optimal solutions within complex design spaces. When integrated with electromagnetic simulation software, they create an iterative optimization workflow. Designers can effectively optimize the S-parameters of antennas, thereby enhancing their performance and reliability, by selecting appropriate algorithms, configuring suitable parameters, and utilizing relevant objective functions. However, validating the suitability of algorithms for optimizing antenna S-parameters through simulation can be time-consuming and resource-intensive. Therefore, Zhen Zhang et al. developed a benchmark test suite for antenna S-parameter optimization [11] to intuitively and rapidly assess the performance of metaheuristic

tic algorithms in antenna design. This benchmark suite simulates the characteristics of electromagnetic simulations and addresses common antenna issues, ranging from single antennas to multiple antennas, thereby tackling the structural design challenges for both types. They demonstrated that the test suite they proposed has the same effect as the electromagnetic simulation of antenna S-parameters, that is, if an algorithm performs well on the test suite, it is suitable for antenna S-parameter optimization. Details of the benchmark functions of the test suite is listed below and Figure 23 is the landscapes of the functions.



**Figure 23.** Landscapes of eight benchmark functions in the test suite for antenna S-parameter optimization

$$F1 = 20 \log(2(\sum_{i=1}^n |\sin(\frac{x_i}{8})|^2) + \sum_{i=1}^n |\sin(\frac{x_i}{8})|) + 1) \quad (65)$$

where  $F1$  is a uni-modal function characterized by a rose-shaped valley, with a minimum value of 0 and a dimension of 8. It is continuous, differentiable, and non-separable for single antenna optimization.

$$F2 = 20 \log(10(\sum_{i=1}^n x_i^2)^2 + 1) \quad (66)$$

where  $F2$  is a uni-modal function with a steep narrow valley, having a minimum value of 0 and a dimension of 8. It is continuous, differentiable, separable, and scalable, for multi-antenna design optimization.

$$F3 = 20 \log(10(\sum_{i=1}^n 0.01i^5 x_i^2)^2 + 1) \quad (67)$$

where  $F3$  is a uni-modal function featuring a long narrow valley, with a minimum value of 0 and a dimension of 8. It is continuous, differentiable, separable, and scalable, for both single and multiple antenna optimization.

$$F4 = 20 \log((\sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 - (x_i - 1)^2)) + 1) \quad (68)$$

where  $F4$  is a uni-modal function characterized by steep and banana-shaped curved valleys, with a minimum value of 0 and a dimension of 8. It is continuous, differentiable, non-separable, and scalable, for multi-antenna design optimization.

$$F5 = 100\sqrt{|x_2 + 1 - 0.01(x_1 - 10)^2|} + 0.01|x_1| \quad (69)$$

where  $F5$  is a multi-modal function with long narrow valleys, having a minimum value of 0 and a dimension of 2. It is continuous, non-differentiable, non-separable, and non-scalable, for multi-antenna design optimization.

$$F6 = 20\log((0.01(\sum_{i=1}^n |x_i|)^2(\sin(0.8x_1) + 2)^2 + 1) \quad (70)$$

where  $F6$  is a multi-modal function with long narrow valleys that intersect, featuring a minimum value of 0 and a dimension of 8. It is continuous, scalable, non-differentiable, and non-separable, for multi-antenna optimization.

$$F7 = 20\log((\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2))^2 - (x_i - 1)^2)) + 1) \\ + 20\log(0.01(\sum_{i=1}^n |x_i|)^2(\sin(0.8x_1) + 2)^2 + 1) \quad (71)$$

where  $F7$  is a multi-modal compositional function with long narrow and intersecting valleys, with a minimum value of 0 and a dimension of 8. It is continuous, non-differentiable, non-separable, and scalable, for multi-antenna design optimization.

$$F8 = 100\sqrt{|x_2 + 1 - 0.01(x_1 - 10)^2|} + 0.01|x_1| \\ + 20\log(0.01(\sum_{i=1}^n |x_i|)^2(\sin(0.8x_1) + 2)^2 + 1) \quad (72)$$

where  $F8$  is a multi-modal function characterized by long narrow and intersecting valleys, with a minimum value of 0 and a dimension of 8. It is continuous, non-differentiable, non-separable, and scalable, for multi-antenna optimization.

We selected AROA, HHO, GWO, WOA, RBMO, and MRBMO to evaluate their performance on the antenna S-parameter optimization through the benchmark test suite. Both the number of iterations  $T=500$  and the population size  $N=30$  were kept consistent across all tests. Each algorithm was executed independently for 30 runs on eight benchmark functions from the suite, and we recorded average fitness, standard deviation,  $p$ -value from the Wilcoxon rank-sum test, and Friedman value for comprehensive performance analysis. Based on the experimental results, MRBMO excels in the antenna S-parameter optimization benchmark suite, significantly outperforming the other algorithms. MRBMO exhibits rapid convergence speeds and high accuracy across many functions. In particular, MRBMO showcases robust optimization capabilities in the benchmark functions. Additionally, the Wilcoxon rank-sum test and Friedman test confirm that MRBMO's performance in various aspects is significantly superior to that of the other algorithms, highlighting its overall excellence.

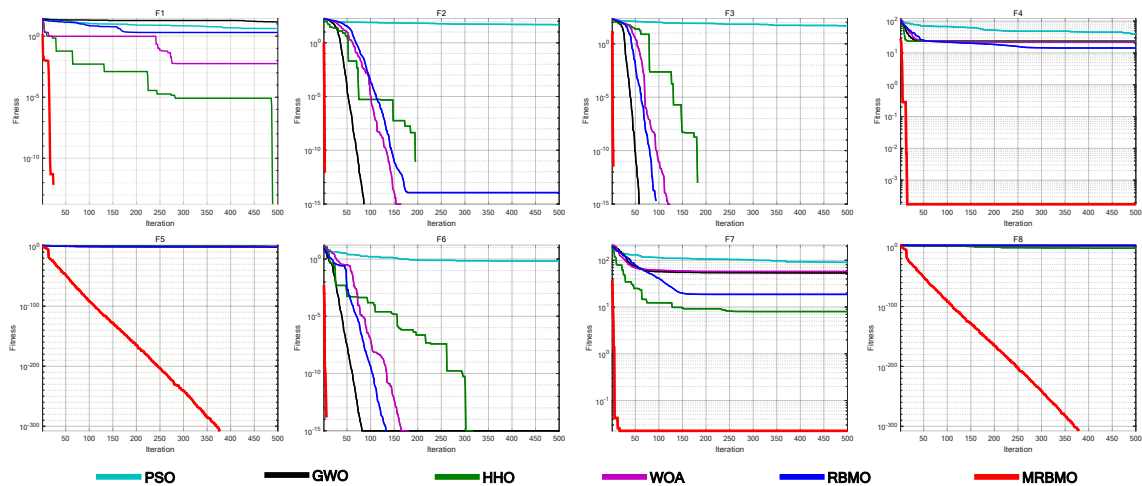


Figure 24. Iteration curves on antenna S-parameter optimization

Table 7. Comparative results of each algorithm in antenna S-parameter optimization

Functions	Index	AROA	HHO	GWO	WOA	RBMO	MRBMO
F1	Ave	1.7007E+00	9.2124E-08	8.6647E-01	1.4631E-02	5.2133E-01	0.0000E+00
	Std	1.5126E+00	4.6530E-07	1.2638E+00	3.5044E-02	5.5072E-02	0.0000E+00
F2	Ave	4.2205E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Std	1.2381E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F3	Ave	3.6173E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Std	6.4519E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F4	Ave	1.8043E+01	9.0489E+00	1.4741E+01	1.5110E+01	4.1788E+00	1.0350E-12
	Std	2.1884E-01	9.0112E+00	1.1248E+00	3.2140E+00	6.2593E+00	3.6532E-12
F5	Ave	1.1390E-01	3.9068E-02	1.1857E-01	3.2537E-02	2.1036E-02	0.0000E+00
	Std	1.0944E-01	8.1350E-02	9.1713E-02	1.5953E-02	1.5856E-02	0.0000E+00
F6	Ave	1.6116E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Std	3.2889E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F7	Ave	3.8973E+01	1.1164E+01	2.4949E+01	2.5166E+01	1.4856E+01	8.5536E-07
	Std	3.0862E+00	1.6702E+01	3.9713E+00	8.1446E+00	2.4862E-05	4.3045E-06
F8	Ave	3.6929E+01	0.0026409	1.6150E+01	50.6156	2.7344E+01	0.0000E+00
	Std	1.5096E+01	6.6231E-03	1.6858E+01	1.9452E+01	1.6428E+01	0.0000E+00

Table 8. Ranking of non-parametric tests of different algorithms on antenna S-parameter optimization

Algorithm	Average Friedman Value	Rank	+/-/-
AROA	5.6625	6	8/0/0
HHO	2.8375	2	5/3/0
GWO	3.2167	3	5/3/0
WOA	3.6792	4	5/3/0
RBMO	3.8542	5	5/3/0
MRBMO	1.7500	1	-

10. Disciussion

The evaluation of MRBMO, as detailed in this study, demonstrates its potential as a powerful tool for solving complex optimization problems. The algorithm’s hybrid nature, combining probabilistic and bio-inspired techniques, allows it to effectively navigate large and multidimensional search spaces, achieving superior performance on the benchmark functions.

The experimental results indicate that MRBMO excels in balancing exploration and exploitation, a critical aspect for ensuring comprehensive search and convergence to optimal solutions. This paper uses a series of benchmark functions to evaluate MRBMO, including uni-modal, multi-modal, and compositional problems. The experimental results show that MRBMO is superior to the selected algorithm in terms of convergence speed and solving accuracy, especially in high-dimensional search space, it shows faster convergence speed, robustness and adaptability. In the simulation experiments, MRBMO performed best in four engineering design optimization problems and antenna S-parameter optimization. These results validate the effectiveness and advantages of MRBMO in handling complex optimization tasks, providing a new perspective for the research and application of optimization algorithms.

Future research directions include the adaptation of MRBMO to specific real-world applications, such as solving TSP problems, financial modeling and machine learning hyper-parameter optimization.

**Author Contributions:** Conceptualization, Junhao Wei; methodology, Junhao Wei; software, Junhao Wei; validation, Junhao Wei; formal analysis, Yuzheng Yan; investigation, Zikun Li; resources, Ngai Cheong; data curation, Ruishen Zhou, Baili Lu and Shirou Pan; writing—original draft preparation, Junhao Wei; writing—review and editing, Junhao Wei and Ngai Cheong; visualization, Junhao Wei and Yanzhao Gu; supervision, Ngai Cheong; project administration, Baili Lu; funding acquisition, Ngai Cheong.

**Funding:** ‘This research and the APC was funded by Macao Polytechnic University (MPU Grant no: RP/FCA-06/2022) and Macao Science and Technology Development Fund (FDCT Grant no: 0044/2023/ITP2).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No dataset is used in this reasearch.

**Acknowledgments:** The supports provided by Macao Polytechnic University (MPU Grant no: RP/FCA-06/2022) and Macao Science and Technology Development Fund (FDCT Grant no: 0044/2023/ITP2) enabled us to conduct data collection, analysis, and interpretation, as well as cover expenses related to research materials and participant recruitment. MPU and FDCT investment in our work have significantly contributed to the quality and impact of our research findings.

**Conflicts of Interest:** The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

- Ave    Average fitness
- Std    Standard deviation
- OE    overall effectiveness

## References

1. James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.
2. John H. Holland. Genetic algorithms. *Scientific American*, 267(1):66–73, 1992.
3. Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.
4. Gang Xu. An adaptive parameter tuning of particle swarm optimization algorithm. *Applied Mathematics and Computation*, 219(9):4560–4569, 2013.
5. Shengwei Fu, Ke Li, Haisong Huang, Chi Ma, Qingsong Fan, and Yunwei Zhu. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems. *Artificial Intelligence Review*, 57, 2024. DOI:10.1007/s10462-024-10716-3.
6. Chixin Xiao, Zixing Cai, and Yong Wang. A good nodes set evolution strategy for constrained optimization. In *2007 IEEE Congress on Evolutionary Computation*, pages 943–950. IEEE, 2007.
7. Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849–872, 2019.
8. Wen Long, Jianjun Jiao, Ming Xu, Mingzhu Tang, Tiebin Wu, and Shaohong Cai. Lens-imaging learning harris hawks optimizer for global optimization and its application to feature selection. *Expert Systems with Applications*, 202:117255, 2022.
9. Yancang Li, Muxuan Han, and Qinglin Guo. Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization. *KSCE Journal of Civil Engineering*, 24(12):3703–3713, 2020.
10. Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.
11. Zhen Zhang, Hongcai Chen, Fan Jiang, Yang Yu, and Qingsha S. Cheng. A benchmark test suite for antenna S-parameter optimization. *IEEE Transactions on Antennas and Propagation*, 69(10):6635–6650, 2021.
12. Adegboye O. R., Feda A. K., Ishaya M. M., et al. Antenna S-parameter optimization based on golden sine mechanism based honey badger algorithm with tent chaos. *Heliyon*, 9(11), 2023.
13. E. Park, S. R. Lee, and I. Lee. Antenna placement optimization for distributed antenna systems. *IEEE Transactions on Wireless Communications*, 11(7):2468–2477, 2012.
14. F. Jiang, C. Y. Chiu, S. Shen, et al. Pixel antenna optimization using NN-port characteristic mode analysis. *IEEE Transactions on Antennas and Propagation*, 68(5):3336–3347, 2020.
15. K. Karthika, K. Anusha, K. Kavitha, et al. Optimization algorithms for reconfigurable antenna design: A review. *Advances in Microwave Engineering*, pages 85–103, 2024.
16. J. Cai, H. Wan, Y. Sun, et al. Artificial bee colony algorithm-based self-optimization of base station antenna azimuth and down-tilt angle. *Telecommunications Science*, 1:69–75, 2021.
17. J. Zhang, J. Xu, Q. Chen, et al. Machine-learning-assisted antenna optimization with data augmentation. *IEEE Antennas and Wireless Propagation Letters*, 22(8):1932–1936, 2023.
18. L. Cui, Y. Zhang, R. Zhang, et al. A modified efficient KNN method for antenna optimization and design. *IEEE Transactions on Antennas and Propagation*, 68(10):6858–6866, 2020.
19. F. Peng and X. Chen. An antenna optimization framework based on deep reinforcement learning. *IEEE Transactions on Antennas and Propagation*, 2024.
20. Dervis Karaboga and Beyza Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108–132, 2009.
21. Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67, 2016.
22. J. O. Agushaka, A. E. Ezugwu, A. K. Saha, et al. Greater cane rat algorithm (GCRA): A nature-inspired metaheuristic for optimization problems. *Heliyon*, 2024.
23. J. Xue and B. Shen. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Systems Science & Control Engineering*, 8(1):22–34, 2020.
24. P. J. M. Van Laarhoven, E. H. L. Aarts, and P. J. M. Van Laarhoven. Simulated annealing. *Springer Netherlands*, 1987.
25. J. Xue and B. Shen. Dung beetle optimizer: A new metaheuristic algorithm for global optimization. *The Journal of Supercomputing*, 79(7):7305–7336, 2023.
26. L. Dayou, Y. Pu, and Y. Ji. Development of a multiobjective GA for advanced planning and scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 42:974–992, 2009.

27. S. He, E. Prempan, and Q. H. Wu. An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization*, 36(5):585–605, 2004.
28. A. R. Yildiz. Comparison of evolutionary-based optimization algorithms for structural design optimization. *Engineering Applications of Artificial Intelligence*, 26(1):327–333, 2013.
29. İ. Şahin, M. Dörterler, and H. Gökçe. Optimization of hydrostatic thrust bearing using enhanced grey wolf optimizer. *Mechanics*, 25(6):480—486, 2019.
30. Yu F, Guan J, Wu H, et al. Lens imaging opposition-based learning for differential evolution with cauchy perturbation[J]. *Applied Soft Computing*, 2024, 152: 111211.
31. Li Y, Han M, Guo Q. Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization[J]. *KSCE Journal of Civil Engineering*, 2020, 24(12): 3703-3713.
32. J. Wei, Y. Gu, K. L. E. Law and N. Cheong, "Adaptive Position Updating Particle Swarm Optimization for UAV Path Planning," 2024 22nd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Seoul, Korea, Republic of, 2024, pp. 124-131.
33. Nadimi-Shahraki Mohammad H., Taghian Shokooh, Mirjalili Seyedali and Faris Hossam. (2020). MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Applied Soft Computing*. 97. 10.1016/j.asoc.2020.106761.
34. Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art[J]. *IEEE transactions on evolutionary computation*, 2010, 15(1): 4-31.
35. Cymerys K, Oszust M. Attraction-Repulsion Optimization Algorithm for Global Optimization Problems[J]. *Swarm and Evolutionary Computation*, 2024, 84: 101459.
36. Xiong G, Zhang J, Shi D, et al. Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm[J]. *Energy conversion and management*, 2018, 174: 388-405.
37. Tu Q, Chen X, Liu X. Multi-strategy ensemble grey wolf optimizer and its application to feature selection[J]. *Applied Soft Computing*, 2019, 76: 16-30.
38. Shen Y, Zhang C, Gharehchopogh F S, et al. An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems[J]. *Expert Systems with Applications*, 2023, 215: 119269.
39. Li Y, Zhao L, Wang Y, et al. Improved sand cat swarm optimization algorithm for enhancing coverage of wireless sensor networks[J]. *Measurement*, 2024, 233: 114649.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.