Journal of Computers Vol. 36 No. 3, June 2025, pp. 69-88
https://doi.org/10.63367/19911599202506360300 6

Modified Secretary Bird Optimization Algorithm with Multi-Strategy for Solving Optimization Problems

Ling-Long Tan*, Yu-Ting Yu, and Qi-Jian Wang

School of Electronics Engineering, Anhui Xinhua University, Hefei, China
tanlinglong@axhu.edu.cn, yuyuting@axhu.edu.cn, wangqijian@axhu.edu.cn

Received 4 September 2024; Revised 16 September 2024; Accepted 16 September 2024

Abstract. The Secretary Bird Optimization Algorithm (SBOA) is an innovative meta-heuristic optimization technique, recently developed and inspired by the natural survival behaviors of secretary birds. In practical optimization scenarios, the SBOA, similar to other meta-heuristic algorithms, often suffers from slow convergence rates, susceptibility to local optima, and inconsistent exploitation capabilities. To improve the performance of the SBOA, a modified Secretary Bird Optimization Algorithm (MSBOA) with Multi-strategy Integration is proposed, which combines the SBOA algorithm, golden section search strategy, and differential evolution technique. The improvement includes two parts: During the exploration phase, the MSBOA employs a golden sine strategy to deepen and broaden local searches. To increase the convergence rate, differential evolution is applied to each population position update during the exploitation phase. To evaluate the effectiveness of the MSBOA in real-world scenarios, the approach under consideration was applied to the IEEE CEC-2021 benchmark functions and six real-world optimization problems. In addition, MSBOA was evaluated as a feature selection method using six standard datasets from the University of California, Irvine (UCI). The simulation results showed that MSBOA outperformed the SBOA and other metaheuristic methods in most benchmark functions, real engineering problems, and feature selection tasks. Thus, the results indicate that the proposed algorithm achieves highly competitive performance and outperforms other metaheuristic methods in solving optimization problems.

Keywords: secretary bird optimization algorithm, golden section sine algorithm, differential evolution algorithm, engineering optimization, feature selection

1 Introduction

The paramount objective of optimization research is to identify the optimal solution strategy from many possibilities for a specific problem, thereby maximizing the overall benefit. An optimization algorithm is defined as a search process or rule based on particular principles and mechanisms, operating according to a defined set of rules or criteria to meet the user's needs. Optimization algorithms are broadly classified into two fundamental types: deterministic and stochastic. Stochastic algorithms are further divided into heuristic and metaheuristic algorithms [1]. Deterministic optimization algorithms aim to find optimal or feasible solutions for optimization problems. In some cases, these algorithms can converge quickly to a solution, particularly when the problem structure permits using analytic solutions or efficient methods. However, for complex, nonlinear, or multimodal optimization problems, deterministic algorithms may become trapped in local optima. Deterministic algorithms may only find locally optimal solutions, rather than the globally optimal solution, for non-convex problems. It is important to recognize that no single solution approach universally applies to problem formulations with varied decision objectives, variables, and constraints. The majority of algorithms are designed to deal with specific types of objective functions or constraints. However, optimization problems in various domains-such as management science, computer science, and engineering - require the integration of different decision variables, objective functions, and constraints. Consequently, metaheuristic optimization algorithms have been developed to deal with this complexity.

Metaheuristic algorithms can be divided into five distinct groups based on where they received their initial inspiration: swarm-based, evolutionary-based, physics-based, game-based, and human-based. As a prominent category of metaheuristic algorithms, swarm-based methods encompass several notable algorithms. These include the Gray Wolf Optimization (GWO) algorithm [2], which is inspired by the social structure and hunting strategies of

* Corresponding Author

gray wolves; the Harris Hawks Optimizer (HHO) [3], which mimics the cooperative hunting behaviors and surprise attack strategies of Harris hawks; the GOOSE Optimization Algorithm [4], which emulates the behavior of geese; the Golden Jackal Optimization (GJO) algorithm [5], which is based on the social and cooperative hunting behavior of golden jackals; the Whale Optimization Algorithm (WOA) [6], which is modeled on the bubble net hunting technique of humpback whales; and the Marine Predators Algorithm (MPA) [7], which is inspired by the search behaviors and movement patterns, such as Brownian and Levy movements, of marine predators.

The SBOA suggested by Fu et al. is a metaheuristic algorithm based on swarm intelligence. By mimicking the evasive and predatory behaviors of secretary birds in their natural environment, it solves optimization challenges encountered in the real world. By balancing exploration and exploitation, SBOA successfully reduces the chance of local optima and has shown its effectiveness in several benchmark experiments. Traditional SBOA, while successful, frequently encounters difficulties in practical implementations, including sluggish convergence rates and a propensity to become stuck in local optima. It is imperative to incorporate multi-strategy methods to overcome these difficulties. Based on the previously described research, this work presents the MSBOA method, which improves local search capacity at each iteration and speeds up convergence by combining golden section search and differential evolution techniques. The golden section algorithm is introduced during the search phase to strengthen the local search ability, and the differential evolution algorithm is introduced during the exploitation phase, which uses a certain proportion of individual differential information as a perturbation to provide the algorithm with adaptive step sizes and search directions. The robustness of the MSBOA is demonstrated by numerical simulation experiments on 10 benchmark functions test functions. Furthermore, MSBOA has the smallest minimum cross-entropy when solving real-world engineering optimization problems, further verifying the superiority and applicability of MSBOA in dealing with actual engineering optimization problems.

The rest of this paper is organized as follows: In Section 2, the theoretical basis of the SBOA is presented. In Section 3, the MSBOA optimization algorithm is described in detail. Section 4 describes the testing of the proposed algorithm on the CEC-2021 benchmark functions, including statistical analyses and supporting figures. Sections 5 and 6 discuss the application of the suggested algorithm to six real-world engineering problems and feature selection, respectively, and present performance analyses of both the proposed algorithm and comparative methods. Concluding remarks are presented in Section 7.

2 Related Works

2.1 Secretary Bird Optimization Algorithm

The predatory and evasive behavior of secretary birds in their natural environment is the main inspiration for the mathematical model of the SBOA. These behaviors are bifurcated into two distinct phases: the exploration phase, which simulates the act of predation, and the exploitation phase, which mimics the act of evasion [8].

(1) Phase of Initial Configuration

The commencement of the algorithm involves the random assignment of these initial coordinates, governed by a predefined mathematical formula.

$$X_{i,j} = lb_j + r_{rand} \times (u_j - l_j), i = 1, 2, ..., N, j = 1, 2, ..., dim. \quad (1)$$

X_i is the position of the i_{th} secretary bird. l_j and u_j are the lower and upper bounds. r_{rand} is a random number between the values 0 and 1.

(2) Exploration Phase

This stage mimics the behavior of a secretary bird feeding on a snake and is divided into the following three sub-phases.

Stage 1 Locating prey

The refinement of the secretary bird's coordinates during the Prey Search phase is mathematically represented by Equations (2) and (3).

While: $t < \frac{1}{3}T$,

$$x_{i,j}^{P1} = x_{i,j} + (x_{r1} - x_{r2}) \times R_1, \tag{2}$$

$$X_i = \begin{cases} X_i^{P1}, if\ \mathrm{F}_i^{P1} < F_i \\ X_i \quad , \text{else} \end{cases}. \tag{3}$$

The equation, t and T stand for the current number of iterations and the maximum number of iterations. X_i^{p1} denotes the new state of the i_{th} secretary bird in the first phase, while x_{r1} and x_{r2} are random candidate solutions during the first phase of iteration. R_1 is an array randomly generated within the interval [0,1] with a dimension of $1 \times dim$, where dim is the dimensionality of the solution space. $x_{i,j}^{P1}$ signifies the value of its j_{th} dimension, and F_i^{P1} indicates the fitness value of the objective function for that individual.

Stage 2 Consuming prey

Brownian motion (RB) was employed in this step to simulate the secretary bird's irregular movement. Equation (4) is the mathematical representation of Brownian motion. Consequently, Equations (5) and (6) mathematically encapsulate the adjustment of the secretary bird's position during the prey consumption phase.

$$I_{RB} = randn(1, dim). \tag{4}$$

While: $\frac{1}{3}T < t < \frac{2}{3}T$,

$$x_{i,j}^{P1} = x_b + \exp((\frac{t}{T}) \wedge 4) \times (I_{RB} - 0.5) \times (x_b - x_{i,j}), \tag{5}$$

$$X_i = \begin{cases} X_i^{P1}, if\ \mathrm{F}_i^{P1} < F_i \\ X_i \quad , \text{else} \end{cases}. \tag{6}$$

In this context, $randn(1, dim)$ means the creation of an array with the dimensions, $1 \times dim$ which is randomly derived from a standard normal distribution characterized by a mean of 0, and a standard deviation of 1. Concurrently, x_b is utilized to denote the optimum value identified in the current iteration.

Stage 3 Attacking Prey

To increase the convergence speed of SBOA during the optimization process, improve the efficiency of the algorithm and avoid premature convergence, a non-linear disturbance factor is introduced, denoted as $(1 - \frac{t}{T}) \wedge (2 \times \frac{t}{T})$. Consequently, the repositioning of the secretary bird in the Prey Attack phase is mathematically formalized via Equations (7) and (8).

While: $t > \frac{2}{3}T$,

$$x_{i,j}^{P1} = x_b + (1 - \frac{t}{T}) \wedge (2 \times \frac{t}{T}) \times x_{i,j} \times RL, \tag{7}$$

$$X_i = \begin{cases} X_i^{P1}, if\ \mathrm{F}_i^{P1} < F_i \\ X_i \quad , \text{else} \end{cases}. \tag{8}$$

To bolster the precision of the optimization process, we employ a strategy known as weighted Levy flight, represented by the symbol '*RL*'.

$$RL = 0.5 \times Levy(dim). \tag{9}$$

In this context, the function *Levy*(*dim*) signifies the distribution function associated with Levy flight. The computation of this function is delineated as follows.

$$Levy(d) = s \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\eta}}}. \tag{10}$$

In this scenario, s is designated as a constant value of 0.01, and η is set as a constant value of 1.5. The variables *Levy*(*dim*) and v represent random numbers that fall within the range [0,1]. The computation for σ is presented below.

$$\sigma = \left(\frac{\Gamma(1+\eta) \times \sin(\frac{x\eta}{2})}{\Gamma(\frac{1+\eta}{2} \times \eta \times 2^{\frac{\eta-1}{2}})}\right)^{\frac{1}{\eta}}. \tag{11}$$

In this equation, Γ refers to the gamma function, and the symbol *Levy*(*dim*) is assigned a value of 1.5.

(3) Exploitation Phase

This stage models the tactics employed by secretary birds to avoid predators, incorporating two distinct strategies: flight or rapid running, and camouflage. In developing the SBOA, we assume that there is an equal probability of encountering either of these two scenarios.

i. C1: Camouflage by blending with the environment,

ii. C2: Flight or rapid escape.

In the initial tactic, when a predator's presence is detected, secretary birds first attempt to blend into their environment. If a secure and suitable hiding spot is not readily available, they then resort to fleeing by taking flight or sprinting away. A dynamic disturbance factor is introduced within the existing framework, and it is denoted as $(1-\frac{t}{T})^2$. The evasion techniques employed by secretary birds are mathematically formalized in Equation (12), while the updated scenario is described in Equation (13).

$$x_{i,j}^{P2} = \begin{cases} C_1 : x_b + (2 \times I_{RB} - 1) \times (1 - \frac{t}{T})^2 \times x_{i,j}, \ if \ \mathrm{r} < r_i \\ C_1 : x_{i,j} + R_2 \times (x_r - K \times x_{i,j}), \quad else \end{cases}, \tag{12}$$

$$X_i = \begin{cases} x_{i,j}^{P2}, \ if \ \mathrm{F}_i^{P2} < F \\ X_i, \quad \text{else} \end{cases}. \tag{13}$$

In this scenario, r is assigned the value of 0.5, and R_2 denotes the process of randomly generating an array with dimensions of 1 × *dim*, drawn from a normal distribution. The term x_r signifies a randomly selected candidate solution for the ongoing iteration. Meanwhile, K indicates the selection at random of either an integer of 1 or an integer of 2, which is determined by the calculation described in Equation (14).

$$K = round(1 + r_r(1,1)). \tag{14}$$

Here, the notation $r_r(1,1)$ refers to the process by which a random number is generated that lies within the interval [0,1].

2.2 Golden Section Sine-Wave Guided Mechanism

The Golden Section Sine Algorithm employs the golden section search methodology to effectively condense the solution space. By using this method, the algorithm can investigate every point on the unit circle that corresponds to every point on the sine function. By incorporating the golden ratio into the position update, the algorithm

enhances its capability to thoroughly search areas that are likely to yield better solutions during each iteration, thus accelerating its convergence rate [9]. The following formula can be used to update the sine location of the golden section.

$$P_x(x) = x_i^t \times |\sin(\lambda_1)| + \lambda_2 \times \sin(\lambda_1) \times |x_1 \times P_{bx}(\mathrm{x}) - x_2 \times P_x(\mathrm{x}-1)|, \tag{15}$$

$$x_1 = \gamma_1\varphi + \gamma_2(1-\varphi),\ x_2 = \gamma_1(1-\varphi) + \gamma_2\varphi. \tag{16}$$

In the formula, P_{bx}(x) denotes the global optimal position during the algorithm's iteration, while x_1 and x_2 represents the golden sine coefficients. $\varphi = (\sqrt{5}-1)/2$ is the golden ratio, λ_1 is chosen as a random number from the interval [0, π], and λ_2 is selected from the interval [0, 2π], The initial values for γ_1 and γ_2 are set to π and $-\pi$, respectively. Within the exploration phase of the MSBOA, the golden section sine strategy is integrated to facilitate the positional updates of the secretary birds. The positional update is governed by the formula presented in Equation 17.

$$P_x(x) = x_i^t \times |\sin(\lambda_1)| - \lambda_2 \times \sin(\lambda_1) \times |x_1 \times P_{bx}(\mathrm{x}) - x_2 \times P_x(\mathrm{x}-1)|. \tag{17}$$

2.3 Standard Differential Evolution Algorithm

The Differential Evolution (DE) algorithm is a metaheuristic approach that uniquely combines two main operations: a differential-based mutation strategy and a probability-driven crossover mechanism [10, 11]. The DE algorithm utilizes a fraction of the differential information between individuals as a perturbation vector, allowing it to adaptively adjust both its step size and search direction. In the early stages of evolution, the high variance between individuals generates significant perturbations, allowing a wider search domain and improving the exploratory capabilities of the algorithm. As the algorithm progresses and converges, the variability among individuals decreases, allowing for more refined local exploitation. Due to its ability to learn from the population, the DE algorithm often outperforms other evolutionary algorithms. The procedure involves initializing a population randomly, computing fitness values, and then generating subsequent populations through differential mutation, crossover, and selection operations.

(1) Differential Mutation Operation

Each parent individual $X_{i,N}$, $X_{i,1,N}$, $X_{i,2,N}$, …, $X_{i,M,N}$ in the population undergoes a differential mutation process to generate a mutant individual $Y_{i,N}$, $Y_{i,1,N}$, $Y_{i,2,N}$, …, $Y_{i,M,N}$. This operation introduces genetic diversity by leveraging the differences between individuals within the population.

(2) Crossover Operation

Utilizing the mutant individual $Y_{i,N}$ produced from the differential mutation operation, a trial individual $Z_{i,N}$, $Z_{i,1,N}$, $Z_{i,2,N}$, …, $Z_{i,M,N}$ is created through a binomial crossover with another parent individual $X_{i,N}$. The specifics of the crossover process are delineated in Equation (18).

$$u_{i,j,N} = \begin{cases} Y_{i,j,N}, & rand(0,1) \le CRorj = j_r \\ X_{i,j,N}, & otherwise \end{cases}, \tag{18}$$

In the equation, $j = 1, 2, \ldots, M$, and j_r is a random integer taken from the interval [1, M]. The parameter CR is the crossover probability.

(3) Selection Operation

The process involves the identification of superior individuals based on their fitness values, from both the parental and trial populations, to be carried forward into the subsequent generation. This procedure is delineated in Equation (19).

$$X_{i,N+1} = \begin{cases} Z_{i,N}, f(Z_{i,N}) \le f(X_{i,N}) \\ X_{i,N}, f(Z_{i,N}) > f(X_{i,N}) \end{cases}. \tag{19}$$

In the equation, $X_{i,N+1}$ represents the individual of the subsequent generation.

In the MSBOA, the global best position undergoes a secondary update using the DE algorithm. If a solution with a better fitness value is found, the new solution replaces the original one.

3 The Proposed MSBOA Algorithm

3.1 MSBOA Algorithm

The SBOA frequently encounters difficulties including slow convergence rates and becoming trapped in local optima. The golden sine method was included during the SBOA algorithm's exploration stage to solve these problems. By modifying the conventional method of updating the position, this approach enhances the efficiency of the algorithm's local search. During the exploitation phase, the Differential Evolution (DE) algorithm's crossover operations are integrated with an elimination mechanism, a selection-based random search strategy, and an algorithm to enhance the program's global search capabilities and accelerate convergence. The MSBOA flowchart is depicted in Fig. 1.

3.2 Computation Complexity

This subsection looks at the suggested algorithm's computational complexity. We apply Big O notation, as described by Tallini et al., to investigate the SBOA's temporal complexity. The numbers N, dim, and T represent the secretary bird's population size, and the amount of dimensions, and the maximum number of iterations, respectively. Following the arithmetic rule of Big O notation, the optimization process of the SBOA method determines that the time complexity for randomly initializing the population is O(N). The optimal position is determined and all solution positions are updated throughout the update process, which has a computational complexity of $O(T\times N\times dim) + O(T\times N)$. As a result, $O(N\times(T\times dim+1))$ represents the SBOA's overall computational complexity.

In this study, the MSBOA improves upon the SBOA. Introducing the golden sine algorithm without adding nested loops to the basic SBOA incurs a time complexity of t_1 for the vertical and horizontal crossover algorithms, and $O(T\times N)$ for the DE algorithm. Thus, the time complexity of the MSBOA can be expressed as $O(N\times(2\times T\times dim+1)+t_1)$.

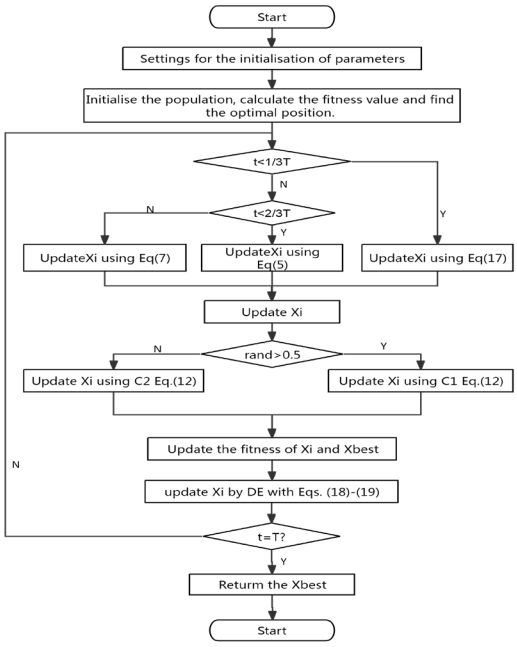


Fig. 1. Flowchart of MSBOA

4 Experimental Results and Analysis

This section presents a comparative analysis of the MSBOA against five contemporary optimization algorithms using the CEC-2021 benchmark suite [12]. Table 1 lists the parameter configurations for the algorithms being compared. A maximum of 200 iterations and a population size multiplier of 30 were uniformly applied across all algorithms. Thirty separate runs of each algorithm were conducted to guarantee the accuracy of the findings. The ensuing section details the empirical outcomes of these trials. The section also highlights the optimal outcomes for each respective test function and dimensionality, distinguished in boldface for emphasis.

Table 1. Comparison of algorithm parameter settings

Algorithms	Parameter name	Parameter values
WOA	a; a2; b	[0,2]; [−1,−2]; 1
GWO	a	[0,2]
GJO	E_1	[1.5, 0]
GOOSE	alpha	[2,0]

A comprehensive experimental evaluation was conducted using the CEC-2021 benchmark suite to assess the performance metrics of the MSBOA. These experiments, supported by the research of Lou et al. (2024), are detailed in Table 2 [13], which presents the specifications of the benchmark functions.

Table 2. CEC-2021 test functions

Function	Features	Rang	Dimension	fmin
F1	UF	[-100,100]	10	0
F2	MF		10	0
F3	MF		10	0
F4	MF		10	0
F5	HF		10	0
F6	HF		10	0
F7	HF		10	0
F8	CF		10	0
F9	CF		10	0
F10	CF		10	0

Table 3. Comparison algorithm experimental results

Function		WOA	GJO	GWO	GOOSE	SBOA	MSBOA
F1	Std.	1.03E-23	6.74E-35	1.88E-15	1.08E+09	1.40E-61	**0.00E+00**
	Mean	4.68E-24	1.62E-35	1.18E-15	8.54E+08	2.55E-62	**1.57E-164**
F2	Std.	4.29E+02	1.82E-12	4.55E-13	6.07E+00	1.12E+03	**0.00E+00**
	Mean	1.97E+02	5.47E-02	4.11E+01	1.05E+03	1.72E+01	**0.00E+00**
F3	Std.	4.07E+00	1.08E+01	2.01E+01	2.21E+02	7.98E+00	**0.00E+00**
	Mean	1.48E+00	2.34E+00	4.01E+01	2.05E+02	5.51E+00	**0.00E+00**
F4	Std.	3.31E-01	5.27E-01	1.09E+00	2.62E+02	1.65E-01	**0.00E+00**
	Mean	6.05E-02	1.44E-01	1.28E+00	1.02E+02	5.91E-02	**0.00E+00**
F5	Std.	5.89E-01	2.49E-01	4.19E+00	1.56E+05	5.71E-01	**2.69E-33**
	Mean	1.08E-01	4.55E-02	3.16E+00	1.15E+05	1.95E-01	**1.60E-33**
F6	Std.	8.49E+01	5.78E-02	2.61E+00	2.37E+02	3.61E+00	**9.30E-03**
	Mean	3.03E+01	3.50E-02	2.76E+00	2.20E+02	1.81E+00	**5.10E-03**
F7	Std.	8.79E-01	4.37E-01	1.70E+00	1.61E+05	2.52E-01	**1.50E-03**
	Mean	3.62E-01	8.88E-02	1.18E+00	9.35E+04	1.17E-01	**2.85E-04**
F8	Std.	4.83E+02	**0.00E+00**	9.98E+00	8.02E+02	**0.00E+00**	**0.00E+00**
	Mean	1.83E+02	**0.00E+00**	2.46E+00	8.76E+02	**0.00E+00**	**0.00E+00**
F9	Std.	1.01E+00	3.37E-15	2.54E-10	9.48E+01	3.06E-43	**3.37E-142**
	Mean	1.84E-01	1.04E-14	2.18E-10	1.34E+02	5.59E-44	**6.43E-143**
F10	Std.	3.35E-02	3.18E+01	1.51E+01	8.81E+01	2.52E+01	**3.89E-04**
	Mean	6.41E-02	3.44E+01	5.48E+01	1.30E+02	3.02E+01	**6.71E-04**

Table 3 shows the mean values and standard deviations produced by the WOA, GWO, GJO, GOOSE, SBOA, and MSBOA algorithms on test functions F1 to F10 (results in bold indicate optimal results). An analysis of the data reveals that for functions F1 through F7, F9, and F10, the MSBOA algorithm yields superior computational

results compared to the other five algorithms. On the F8 test function, the optimization results of GJO, SBOA, and MSBOA are relatively close in performance.

As shown in Fig. 2, the MSBOA exhibits exceptional convergence capabilities. In the unimodal function F1, MSBOA outperforms both other algorithms and SBOA in convergence accuracy. For basic functions, F2 to F4 and F8, MSBOA demonstrates commendable search capabilities, with performance comparable to SBOA. In functions F5 to F7 and F9 to F10, MSBOA consistently surpasses other algorithms in terms of search improvement results. Fig. 2 notably illustrates a pronounced escalation in the count of function inflection points for functions F6 and F7. This observation suggests that the algorithm in question not only demonstrates swift convergence but also adeptly evades local optima in subsequent iterations. Such behavior is indicative of an enhanced capacity for exploration, a critical attribute in the optimization process.

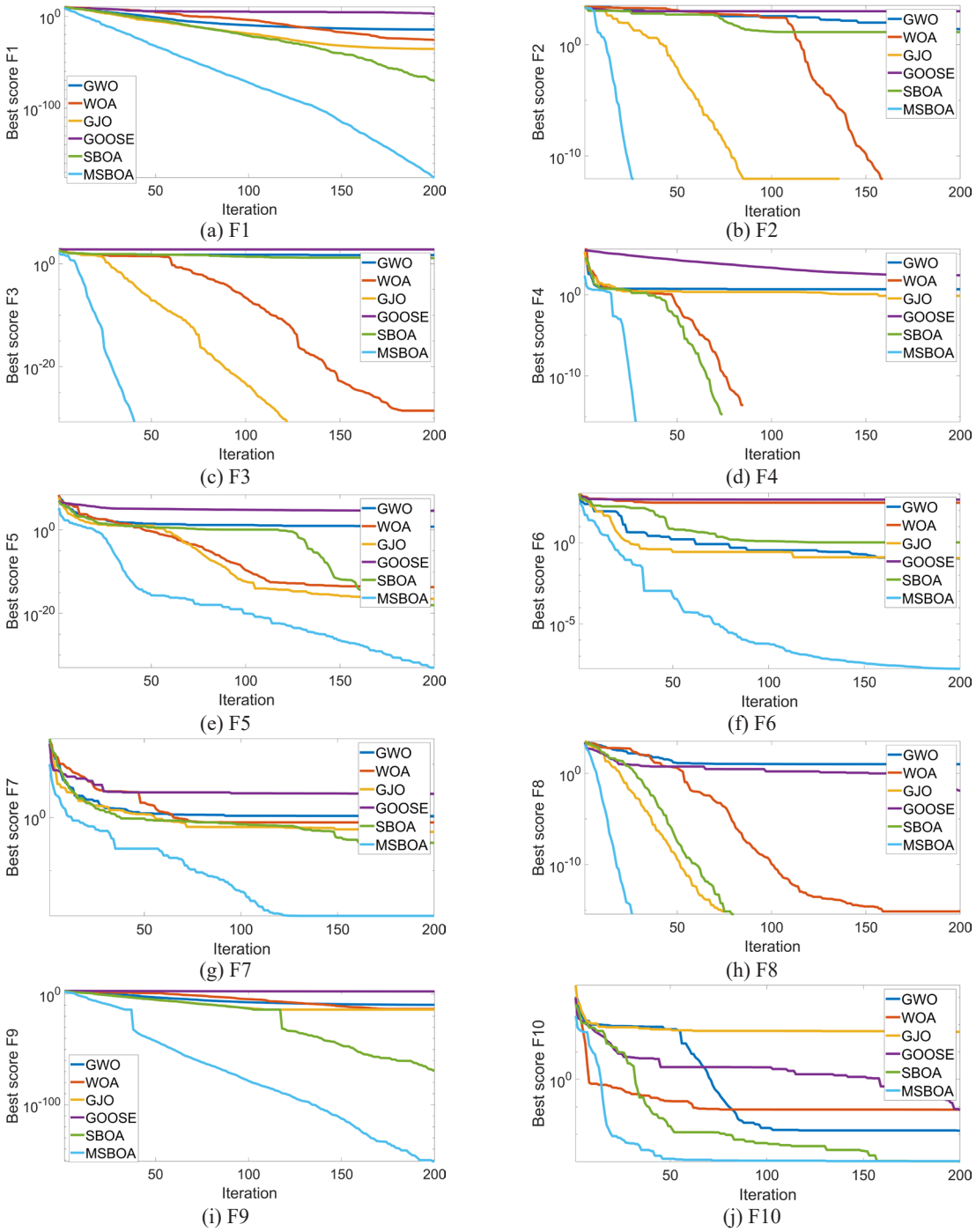


Fig. 2. Convergence effect of test function

5 Engineering Optimization Problems

The MSBOA is deployed to tackle a suite of six engineering design problems that encompass a spectrum of complexities within the engineering discipline. These problems vary in the number of design variables and constraints, ranging from four to fourteen design variables and from simple to complex constraints. For example, the relatively simple pressure vessel design problem involves only four design variables, while the optimization design problem of an industrial refrigeration system is a highly complex constrained optimization problem with numerous constraints and design variables. As a result, the computational complexity of these problems is considerable, which is a serious obstacle to the stability and performance of the algorithm. Each experiment was repeated ten times and compared with other algorithms to ensure the credibility and impartiality of the results.

5.1 Pressure Vessel

The objective of the pressure vessel optimization design problem is to reduce economic costs while complying with a set of constraints. These charges cover the vessel's material, forming, and welding costs [14]. The limitations concern the cylinder's length, the cylinder's radii and head, and the thickness of the cylinder and wall. The following are the restrictions and objective functions for this optimization problem.

Minimize:

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3. \tag{20}$$

Subject to:

$$\begin{aligned} &p_1(x) = 0.0193x_3 - x_1 \le 0, \;\; p_2(x) = 0.00954x_3 - x_2 \le 0, \\ &p_3(x) = 1296000 - \pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 \le 0, \;\; p_4(x) = x_4 - 240 \le 0. \end{aligned} \tag{21}$$

In the equation, x_1 and x_2 represents the thickness of the cylinder head and the thickness of the cylinder wall respectively. x_3 denotes the radii of the cylinder and cylinder head, while x_4 denotes the cylinder length. The variables $x_1 \ge 0.0625$, $x_2 \le 6.1875$, x_1 and x_2 are all discrete and have a 0.0625 interval, while x_3 and x_4 are continuous.

Fig. 3(a) shows the convergence curves of MSBOA compared to several other algorithms for the pressure vessel design problem. The comparative results between MSBOA and these algorithms are detailed in Table 4. Table 5 shows the statistical indicators related to this optimization problem. From the simulation results, it can be seen that the optimal solution for the pressure vessel design problem was achieved with the design variables (0,7782, 0,3846, 40,3196, 200,0000), which gives an objective function value of 5.89×10^3. These findings highlight the significantly enhanced optimization capability of the improved MSBOA relative to the original algorithm.

Table 4. Pressure vessel design optimization data

Algorithm	Optimum variables				
	x_1	x_2	x_3	x_4	Optimum cost
MSBOA	0.7782	0.3846	40.3196	200.0000	5.89E+03
SBOA	0.7792	0.3863	40.3700	199.3127	7.32E+03
WOA	1.0433	0.6525	52.8719	77.0769	6.33E+04
GWO	0.8038	0.4129	41.5960	184.5133	7.30E+03
GJO	0.8209	0.4392	42.1347	177.6859	7.85E+03
GOOSE	1.0760	0.5359	55.7496	178.1193	4.64E+05

Table 5. Pressure vessel design problem statistical results

Algorithm	Best	Mean	Worst	Std	Median
MSBOA	5.89E+03	5.89E+03	5.89E+03	2.76E-06	5.89E+03
SBOA	7.32E+03	6.58E+03	5.89E+03	5.72E+02	6.60E+03
WOA	6.33E+04	2.37E+04	7.30E+03	2.18E+04	1.38E+04
GWO	7.30E+03	6.49E+03	6.02E+03	4.93E+02	6.22E+03
GJO	7.85E+03	7.07E+03	6.15E+03	6.40E+02	7.25E+03
GOOSE	4.64E+05	2.32E+05	1.15E+04	1.44E+05	2.23E+05

5.2 Gear Train

This research deals with a constrained engineering design problem with the aim of determining the optimal configuration of a compound gear train. The goal is to minimize the overall gear ratio, a crucial parameter that affects the system's efficiency and performance [15]. The design variables are four integer values, each representing the number of teeth in the gears of the train. These are denoted as (z_d, z_b, z_a, z_f). It is noteworthy that the problem is characterized by a single inequality constraint, thereby precluding the presence of any equality constraints. The following are the restrictions and objective functions for this optimization problem.

Consider:

$$X = (x_1, x_2, x_3, x_4) = (z_d, z_b, z_a, z_f),\ x \in \{12,13,...,60\}. \tag{22}$$

Minimize:

$$f(X) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right)^2. \tag{23}$$

Subject to: $12 \le x_i \le 60$, $i = 1, 2, 3, 4$.

The MSBOA algorithm was implemented along with several comparison algorithms to validate the performance of the proposed improved algorithm for the gear train design problem. Table 6 shows the experimental results. The MSBOA algorithm achieved the optimal design solution for the gear train design problem with the design parameters set to (43,406, 16,0911, 19,0857, 48,5472), resulting in an objective function value of 9.94×10^{-11}. Table 7 shows the statistical indicators for the optimization of this problem. Fig. 3(b) also shows the convergence behavior of the algorithms in question. The significant advantages of the MSBOA algorithm in this design context are highlighted by the analysis of the experimental data.

Table 6. Gear train design optimization data

Algorithm	Optimum variables				Optimum cost
	z_d	z_b	z_a	z_f	
MSBOA	43.4060	16.0911	19.0857	48.5472	9.94E-11
SBOA	44.0869	21.4056	13.4048	42.5846	2.36E-09
WOA	45.6606	25.5141	12.0000	47.1944	2.73E-08
GWO	50.7359	26.0247	14.6477	53.4249	2.73E-08
GJO	53.7231	12.0000	37.4380	57.4747	2.73E-08
GOOSE	40.3327	15.0948	14.5043	38.6243	1.10E-02

Table 7. Gear train design problem statistical results

Algorithm	Best	Mean	Worst	Std	Median
MSBOA	9.94E-11	1.85E-11	2.70E-12	3.00E-11	2.70E-12
SBOA	2.36E-09	1.22E-09	1.55E-10	6.84E-10	1.09E-09
WOA	2.73E-08	7.10E-09	9.92E-10	1.07E-08	2.36E-09
GWO	2.73E-08	3.63E-09	2.31E-11	8.35E-09	1.04E-09
GJO	2.73E-08	5.14E-09	8.89E-10	8.12E-09	1.86E-09
GOOSE	1.10E-02	1.30E-03	2.36E-09	3.50E-03	4.94E-07

5.3 Welded Beam Structure

The objective of the welded beam structure is to minimize manufacturing costs. It includes seven constraints related to stress, deflection, welding, and geometry [16]. The variables include weld thickness x_1, height x_2, length x_3, and beam thickness x_4. The following are the restrictions and objective functions for this optimization problem.

Minimize:

$$f(X) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2). \tag{24}$$

Subject to:

$$\begin{aligned}
&p_1(X) = \tau(X) - \tau_{\max} \le 0,\ p_2(X) = \sigma(X) - \sigma_{\max} \le 0,\ p_3(X) = \delta(X) - \delta_{\max} \le 0,\\
&p_4(X) = x_1 - x_4 \le 0,\ p_5(X) = P - P_c(X) \le 0,\ p_6(X) = 0.125 - x_1 \le 0,\\
&p_7(X) = 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0,\\
&\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{X_2}{2R} + (\tau'')^2},\ \tau' = \frac{P}{\sqrt{2} x_1 x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right),\\
&R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},\ J = 2\left\{\sqrt{2} x_1 x_2 \left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\\
&\sigma(\overline{X}) = \frac{6PL}{x_4 x_3^2},\ \delta(\overline{X}) = \frac{6PL^3}{E x_3^2 x_4},\ P_c(\overline{X}) = \frac{4.013E\sqrt{x_3^2 x_4^6 / 36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\
&P = 6000\ lb,\ \text{L}=14\ in,\ \delta_{\max} = 0.25\ in,\ E = 30 \times 10^6\ psi,\\
&G = 12 \times 10^6\ psi,\ \tau_{\max} = 13{,}600 psi,\ \sigma_{\max} = 30{,}000 psi.
\end{aligned} \tag{25}$$

Variable range: $0.1 \le x_1, x_4 \le 2$, $0.1 \le x_2, x_3 \le 10$.

Table 8 and Table 9 show comparisons of the best and statistical results obtained by different optimizers for the welded beam design problem. Convergence plots for these algorithms are illustrated in Fig. 3(c). Among these algorithms, MSBOA achieves the minimum cost that satisfies the constraints, outperforming the other five algorithms in terms of cost-effectiveness. In addition, the MSBOA exhibits the lowest standard deviation among the evaluated algorithms, which underscores its robustness in addressing the problem at hand. The experimental outcomes corroborate the superior performance of MSBOA in comparison to the other algorithms under consideration.

Table 8. Welded beam structure design optimization data

Algorithm	Optimum variables				Optimum cost
	x_1	x_2	x_3	x_4	
MSBOA	0.7782	0.3846	40.3196	200.0000	5.89E+03
SBOA	0.7792	0.3863	40.3700	199.3127	7.32E+03
WOA	1.0433	0.6525	52.8719	77.0769	6.33E+04
GWO	0.8038	0.4129	41.5960	184.5133	7.30E+03
GJO	0.8209	0.4392	42.1347	177.6859	7.85E+03
GOOSE	1.0760	0.5359	55.7496	178.1193	4.64E+05

Table 9. Welded beam structure design problem statistical results

Algorithm	Best	Mean	Worst	Std	Median
MSBOA	5.89E+03	5.89E+03	5.89E+03	2.76E-06	5.89E+03
SBOA	7.32E+03	6.58E+03	5.89E+03	5.72E+02	6.60E+03
WOA	6.33E+04	2.37E+04	7.30E+03	2.18E+04	1.38E+04
GWO	7.30E+03	6.49E+03	6.02E+03	4.93E+02	6.22E+03
GJO	7.85E+03	7.07E+03	6.15E+03	6.40E+02	7.25E+03
GOOSE	4.64E+05	2.32E+05	1.15E+04	1.44E+05	2.23E+05

5.4 Step-cone Pulley

The primary objective of this problem is to minimize the weight of a four-stage conical pulley system using five variables. Four variables denote the diameter of each pulley stage, while the fifth represents the width of the pulleys [17]. There are eleven non-linear constraints in the problem. The following are the restrictions and objective functions for this optimization problem.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \omega \end{bmatrix}. \tag{26}$$

$16 \le \omega \le 100$, $40 \le d_i \le 100m$, while, $i = 1, 2, 3, 4$.

Minimize:

$$f(X) = \rho\omega\left[d_1^2\left(1+\left(\frac{N_1}{N}\right)^2\right)+d_2^2\left(1+\left(\frac{N_2}{N}\right)^2\right)+d_3^2\left(1+\left(\frac{N_3}{N}\right)^2\right)+d_4^2\left(1+\left(\frac{N_4}{N}\right)^2\right)\right]. \tag{27}$$

Subject to:

$$\begin{aligned} &h_1\left(\vec{x}\right) = G_1 - G_2 = 0,\ h_2\left(\vec{x}\right) = G_1 - G_3 = 0,\ h_3\left(\vec{x}\right) = G_1 - G_4 = 0, \\ &g_{i=1,2,3,4}\left(\vec{x}\right) = -V_i \le 2,\ g_{i=5,6,7,8}\left(\vec{x}\right) = \left(0.75 \times 745.6998\right) - D_i \le 0. \end{aligned} \tag{28}$$

Where,

$$G_i = \frac{\pi d_i}{2}\left(1+\frac{N_i}{N}\right)+\frac{\left(\frac{N_i}{N}-1\right)^2}{4a}+2a,\ i=(1,2,3,4),\ V_i = \exp\left(\mu\left\{\pi-2\sin^{-1}\left\{\left(\frac{N_i}{N}-1\right)\frac{d_i}{2a}\right\}\right\}\right),\ i=(1,2,3,4),$$

$$D_i = st\omega(1-R_i)\frac{\pi d_i N_i}{60},\ i=(1,2,3,4),\ t=8mm,\ s=1.75MPa,\ \mu=0.35,\ \rho=7200kg/m^3,\ a=3mm.$$

The proposed MSBOA and comparison algorithms are applied to optimize the design parameters of the step-cone pulley and the results are summarised in Table 10. The results of the simulation indicate that MSBOA has achieved the optimal solution for the step cone pulley design problem with design variables of (29.8977, 37.2343, 50.7967, 84.4957, 90.0000). This gives an objective function value of 12.80. The statistical indicators for this optimization task are shown in Table 11. They underline the outstanding performance of MSBOA compared to competing algorithms. Fig. 3(d) shows the convergence curve of MSBOA. It shows the path taken by MSBOA to reach the optimal solution for this engineering challenge.

Table 10. Step-cone pulley design optimization data

Algorithm	Optimum variables					Optimum cost
	d_1	d_2	d_3	d_4	ω	
MSBOA	29.8977	37.2343	50.7967	84.4957	90.0000	1.28E+01
SBOA	29.8813	37.2895	50.7976	84.496	90.0000	1.28E+01
WOA	31.2554	41.8955	51.9985	86.4983	87.9182	2.04E+01
GWO	30.3785	38.2880	51.1197	84.5493	90.0000	1.30E+01
GJO	30.4838	35.9626	52.2804	84.6074	90.0000	1.33E+01
GOOSE	30.4672	34.5932	56.5287	84.5146	89.9808	1.90E+07

Table 11. Step-cone pulley design problem - statistical results

Algorithm	Best	Mean	Worst	Std	Median
MSBOA	1.28E+01	1.28E+01	1.28E+01	3.65E-15	1.28E+01
SBOA	1.28E+01	1.28E+01	1.28E+01	9.20E-03	1.28E+01
WOA	2.04E+01	1.66E+01	1.33E+01	3.18E+00	1.54E+01
GWO	1.30E+01	1.29E+01	1.28E+01	5.93E-02	1.29E+01
GJO	1.33E+01	1.30E+01	1.29E+01	1.56E-01	1.30E+01
GOOSE	1.90E+07	3.81E+06	1.33E+01	7.45E+06	2.99E+01

5.5 Robot Gripper Problem

This research uses MSBOA to optimize the performance of a robotic gripper [18]. The primary goal is to minimize the variance in the forces exerted by the gripper, in particular the discrepancy in the minimum and maximum forces that can be generated by the gripper. Seven decision variables, each subject to a set of six non-linear constraints, are optimized in this complex engineering challenge [19]. The following are the restrictions and objective functions for this optimization problem.

Minimize:

$$f\left(\vec{x}\right) = -\min_z F_k\left(x,z\right)+\max_z F_k\left(x,z\right). \quad (29)$$

Subject to:

$$p_1\left(\vec{x}\right)=-Y_{\min}+y\left(\vec{x},Z_{\max}\right)\le 0,\ p_2\left(\vec{x}\right)=-y\left(x,Z_{\max}\right)\le 0,\ p_3\left(\vec{x}\right)=Y_{\max}-y\left(\vec{x},0\right)\le 0,$$
$$p_4\left(\vec{x}\right)=y\left(\vec{x},0\right)-Y_G\le 0,\ p_5\left(\vec{x}\right)=I^2+e^2-\left(a+b\right)^2\le 0, \quad (30)$$
$$p_6\left(\vec{x}\right)=b^2-\left(a-e\right)^2-\left(l-Z_{\max}\right)^2\le 0,\ p_7\left(\vec{x}\right)=Z_{\max}-l\le 0\ .$$

Where,

$$\alpha=\cos^{-1}\left(\frac{a^2+p^2-b^2}{2ap}\right)+\phi,\ p=\sqrt{e^2+\left(z-l\right)^2},\ \beta=\cos^{-1}\left(\frac{b^2+p^2-a^2}{2bp}\right)-\phi,\ \phi=\tan^{-1}\left(\frac{e}{l-z}\right),$$
$$y\left(x,z\right)=2\left(f+e+c\sin(\beta+\delta)\right),\ F_k=\frac{Db\sin(\alpha+\beta)}{2c\cos\left(\alpha\right)},\ Y_{\min}=50,\ Y_{\max}=100,\ Y_G=150,\ Z_{\max}=100,\ D=100.$$

With bounds: $0\le e\le 50$, $100\le c\le 200$, $10\le f, a, b\le 150$, $1\le\delta\le 3.14$, $100\le l\le 300$.

The proposed MSBOA and comparison algorithms were used to optimize the robot gripper design variables, and the results are summarised in Table 12. They show the superior performance of MSBOA when compared to competing algorithms. Statistical indicators derived from the optimization of the problem are shown in Table 13. Fig. 3(e) illustrates the convergence curve of MSBOA. It shows the path taken by MSBOA to reach the optimal solution for this engineering challenge.

Table 12. Robot gripper optimization data

Algorithm	Optimum variables							Optimum cost
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	
MSBOA	100.0000	38.1966	200.0000	0.0000	125.2204	100.0000	2.1050	2.99E+00
SBOA	99.9488	38.1447	199.968	0.0000	28.4753	100.0000	1.6008	4.34E+00
WOA	93.4649	31.6613	100.0000	0.0000	41.111	100.0000	1.8119	4.92E+01
GWO	95.9319	33.7675	182.5086	0.0000	133.9747	100.0000	2.2739	7.39E+00
GJO	93.6277	31.703	197.8412	0.0000	84.7737	100.0000	2.0085	6.65E+00
GOOSE	146.0027	111.9677	144.838	28.2971	41.6091	162.1755	2.2290	1.11E+07

Table 13. Robot gripper design problem - statistical results

Algorithm	Best	Mean	Worst	Std	Median
MSBOA	2.99E+00	5.70E-01	7.27E-17	1.20E+00	7.27E-17
SBOA	4.34E+00	4.34E-01	7.31E-17	1.37E+00	8.11E-17
WOA	4.92E+01	1.09E+01	2.41E-16	1.51E+01	4.68E+00
GWO	7.39E+00	5.06E+00	1.16E-16	2.02E+00	5.32E+00
GJO	6.65E+00	2.07E+00	1.22E-16	2.78E+00	1.37E-15
GOOSE	1.11E+07	1.21E+06	5.84E+00	3.49E+06	6.79E+02

5.6 Optimal Design of Industrial Refrigeration System

With the escalating consumption of primary energy sources, energy conservation, and emission reduction have become crucial focal points across various sectors. Industrial refrigeration systems, accounting for a substantial share of corporate energy usage, require optimization and efficient control. The design optimization of industrial refrigeration systems poses a complex engineering challenge, involving fourteen design variables and fifteen constraints [20]. The following are the restrictions and objective functions for this optimization problem.

Minimize:

$$f\left(\vec{x}\right)=63098.88x_2x_4x_{12}+5441.5x_2^2x_{12}+115055.5x_2^{1.664}x_6+6172.27x_2^2x_6+63098.88x_1x_3x_{11}+5441.5x_1^2x_{11}$$
$$+115055.5x_1^{1.664}x_5+6172.27x_1^2x_5+140.53x_1x_{11}+281.29x_3x_{111}+70.26x_1^2+281.29x_1x_3+281.29x_3^2 \quad (31)$$
$$+14437x_8^{1.8812}x_{12}^{0.3424}x_{10}x_{14}^{-1}x_7x_9^{-1}+20470x_7^{2.893}x_{11}^{0.316}x_1^2.$$

Subject to:

$$p_1\left(\vec{x}\right)=1.524x_7^{-1}\le 1,\ p_2\left(\vec{x}\right)=1.524x_8^{-1}\le 1,\ p_3\left(\vec{x}\right)=0.07789x_1-2x_7^{-1}x_9-1\le 0,$$
$$p_4\left(\vec{x}\right)=7.05305x_9^{-1}x_1^2x_{10}x_8^{-1}x_2^{-1}x_{14}^{-1}-1\le 0,\ p_5\left(\vec{x}\right)=0.0833x_{13}^{-1}x_{14}-1\le 0,$$
$$p_6\left(\vec{x}\right)=47.136x_2^{0.333}x_{10}^{-1}x_{12}-1.333x_8x_{13}^{2.1195}+62.08x_{13}^{2.1195}x_{12}^{-1}x_8^{0.2}x_{10}^{-1}-1\le 0,$$
$$p_7\left(\vec{x}\right)=0.04771x_{10}x_8^{1.8812}x_{12}^{0.3424}-1\le 0,\ p_8\left(\vec{x}\right)=0.0488x_9x_7^{1.893}x_{11}^{0.316}-1\le 0, \quad (32)$$
$$p_9\left(\vec{x}\right)=0.0099x_1x_3^{-1}-1\le 0,\ p_{10}\left(\vec{x}\right)=0.0193x_2x_4^{-1}-1\le 0,\ p_{11}\left(\vec{x}\right)=0.0298x_1x_5^{-1}-1\le 0,$$
$$p_{12}\left(\vec{x}\right)=0.056x_2x_6^{-1}-1\le 0,\ p_{13}\left(\vec{x}\right)=2x_9^{-1}-1\le 0,\ p_{14}\left(\vec{x}\right)=2x_{10}^{-1}-1\le 0,$$
$$p_{15}\left(\vec{x}\right)=x_{12}x_{11}^{-1}-1\le 0.$$

With bounds: $0.001\le x_i\le 5$, i = 1, 2, ..., 14.

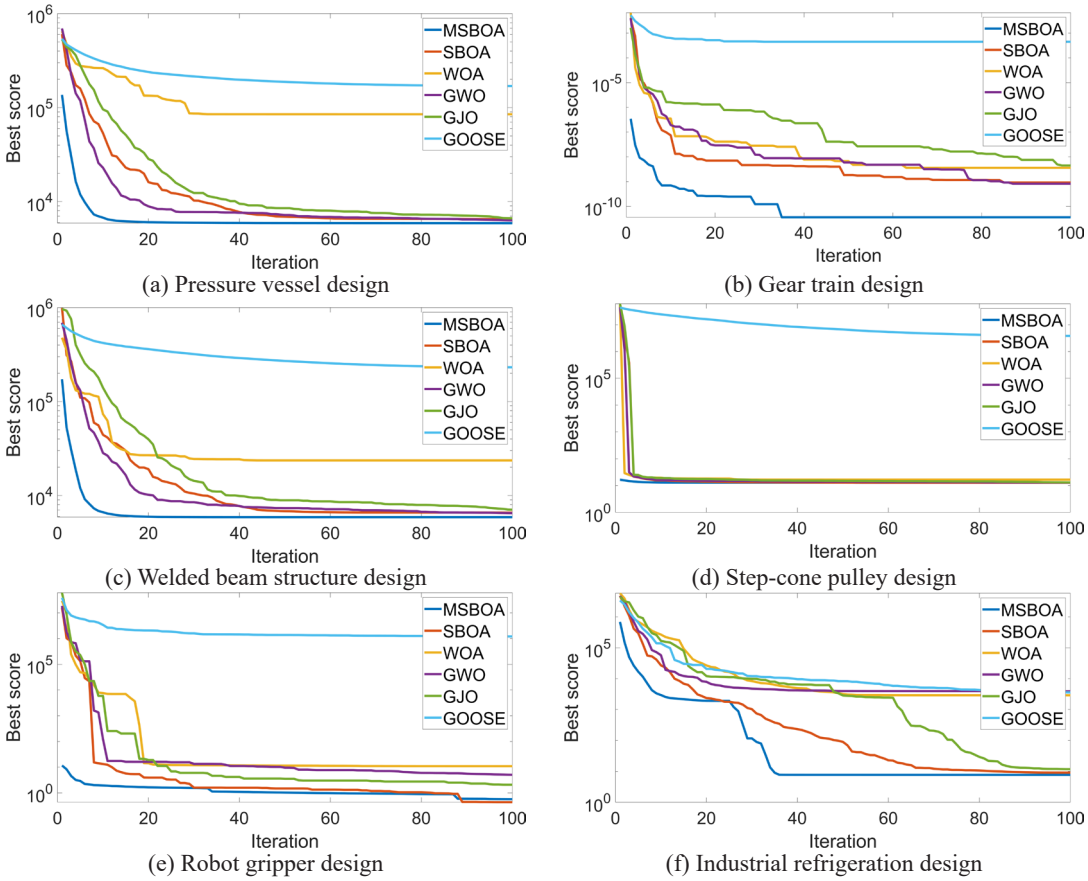
The MSBOA algorithm was utilized to optimize 14 key variables of the industrial refrigeration system and determine their optimal values. Table 14 presents the optimal variable values achieved by each algorithm, while Table 15 reports statistical indicators for this optimization process. Convergence plots for these algorithms are illustrated in Fig. 3(f). Analysis of Table 14 indicates that the MSBOA demonstrates significant efficiency in optimizing the engineering problem involving 14 design variables. Each parameter value showed varying degrees of improvement, ultimately converging to an optimal value of 7.7140. This demonstrates that the MSBOA algorithm is robust in solving multi-constraint engineering problems.

Table 14. Industrial refrigeration system optimization data

Optimum variables	Algorithm					
	MSBOA	SBOA	WOA	GWO	GJO	GOOSE
x_1	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03
x_2	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.90E-03	1.00E-03
x_3	1.00E-03	1.00E-03	1.00E-03	4.20E-03	1.30E-03	1.00E-03
x_4	1.00E-03	7.90E-03	3.00E+00	2.50E-03	4.20E-03	1.86E+00
x_5	1.00E-03	1.10E-03	1.87E+00	2.93E-01	2.01E-01	3.12E+00
x_6	1.00E-03	1.20E-03	3.74E+00	2.17E-01	1.10E-03	1.77E+00
x_7	1.51E+00	1.51E+00	1.47E+00	1.51E+00	1.51E+00	1.52E+00
x_8	1.52E+00	1.52E+00	2.67E+00	3.06E+00	3.02E+00	5.00E+00
x_9	5.00E+00	5.00E+00	5.00E+00	4.86E+00	5.00E+00	2.99E+00
x_{10}	2.00E+00	2.16E+00	1.91E+00	3.28E+00	2.94E+00	2.99E+00
x_{11}	1.00E-03	1.00E-03	4.30E-01	1.00E-03	1.00E-03	2.81E+00
x_{12}	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03
x_{13}	7.30E-03	7.10E-03	1.00E-03	5.50E-03	6.60E-03	6.60E-03
x_{14}	8.76E-02	7.50E-02	1.00E-03	5.54E-02	7.86E-02	3.45E-02
Optimum cost	7.71E+00	2.13E+01	1.04E+04	1.95E+04	3.97E+01	7.70E+03

Table 15. Industrial refrigeration design problem - statistical results

Algorithm	Best	Mean	Worst	Std	Median
MSBOA	7.71E+00	7.71E+00	7.71E+00	1.50E-03	7.71E+00
SBOA	2.13E+01	9.11E+00	7.72E+00	4.29E+00	7.72E+00
WOA	1.04E+04	2.90E+03	6.33E+01	2.99E+03	2.78E+03
GWO	1.95E+04	3.90E+03	8.50E+00	8.19E+03	9.59E+00
GJO	3.97E+01	1.18E+01	8.05E+00	9.87E+00	8.33E+00
GOOSE	7.70E+03	3.59E+03	1.03E+02	2.22E+03	3.53E+03



(a) Pressure vessel design (b) Gear train design

(c) Welded beam structure design (d) Step-cone pulley design

(e) Robot gripper design (f) Industrial refrigeration design

Fig. 3. Convergence diagrams

6 Feature Selection

A critical technique in dimensionality reduction is feature selection, the process of choosing a subset of features from a larger set to optimize system performance. This process enhances predictive accuracy, reduces model complexity, accelerates computations, and improves interpretability. Mathematically, the selection of features is described as a multi-objective optimization problem. The fitness function that evaluates the selected features is formulated as follows [21].

$$fitness = (1-\sigma)\times D + \sigma \times \frac{|l|}{|L|}. \tag{33}$$

Where D is the classification error rate of the classifier, σ is the weight coefficient, $|L|$ and $|l|$ are represent the number of features in the original dataset and the number of features in the subset selected by the algorithm.

To further validate the optimization capabilities of the MSBOA, this section applies it to the problem of feature selection. The algorithm has been tested on six sets of data from the University of California, Irvine (UCI), details of which are given in Table 16. All datasets were classified using a KNN classifier with $k = 5$. Accuracy was assessed using a tenfold cross-validation method.

This study employs the MSBOA for feature selection across six datasets and compares its performance with the WOA, GWO, GJO, GOOSE, and SBOA algorithms. To reduce randomness and error, the size of the population was fixed at 30 and each algorithm ran 100 iterations independently. Table 17 provides a comparison of the average classification accuracies achieved by these algorithms across the six datasets. In particular, compared to the other benchmark algorithms, MSBOA shows a significantly higher average classification accuracy. Table 18 shows the number of features within the optimal subsets selected by the six algorithms. The results indicate that MSBOA consistently identifies optimal feature subsets across all six datasets. MSBOA achieved the best results in 5 out of 6 datasets in Table 19, which shows the average fitness values. Table 20 illustrates the best fitness scores, demonstrating that MSBOA consistently obtained the highest fitness values across these datasets. Fig. 4 displays the convergence curves illustrating the algorithm's adaptability. It visually demonstrates the enhanced convergence rate of MSBOA when addressing the feature selection problem. Compared to WOA, GWO, GJO, GOOSE, and SBOA, MSBOA exhibits significantly better convergence of fitness values across five datasets. This underscores MSBOA's distinct advantage in feature selection.

Table 16. Describe the benchmark datasets used

Datasets	Number of features	Number of instances
Zoo	16	101
Vehicle	18	846
Sonar	60	208
Heart	13	270
Ionosphere	34	351
Breast	30	569

Table 17. Comparison of the average classification accuracy

Data set	Zoo	Vehicle	Sonar	Heart	Ionosphere	Breast
GWO	0.9806	0.2244	0.9317	0.8395	0.1934	0.7738
WOA	0.9774	0.2307	0.9222	0.8383	0.1821	0.7738
GOOSE	0.9290	0.1902	0.9365	0.8111	0.1840	0.7631
GJO	0.9903	0.2484	0.9810	0.8444	0.2302	0.7702
SBOA	0.9839	0.2469	0.9746	0.8469	0.2330	0.7690
MSBOA	1.0000	0.2480	0.9889	0.8506	0.2358	0.7738

Table 18. Comparison of the average number of feature selections

Data set	Zoo	Vehicle	Sonar	Heart	Ionosphere	Breast
GWO	0.4118	0.2632	0.3426	0.4071	0.3000	0.3000
WOA	0.3706	0.2842	0.3180	0.4000	0.1914	0.3000
GOOSE	0.3529	0.3421	0.3787	0.3786	0.3229	0.4000
GJO	0.2941	0.2316	0.2443	0.4643	0.1086	0.3000
SBOA	0.3412	0.2105	0.2672	0.4214	0.0857	0.3000
MSBOA	0.3412	0.2105	0.2443	0.4286	0.0857	0.3000

Table 19. Comparison of the average of the fitness values

Data set	Zoo	Vehicle	Sonar	Heart	Ionosphere	Breast
GWO	0.0471	0.7103	0.0740	0.1802	0.7389	0.2336
WOA	0.0412	0.7014	0.0948	0.1762	0.7473	0.2336
GOOSE	0.0643	0.7402	0.0454	0.1802	0.7247	0.2343
GJO	0.0294	0.6978	0.0180	0.1762	0.6963	0.2336
SBOA	0.0294	0.6978	0.0372	0.1762	0.6963	0.2336
MSBOA	0.0294	0.6978	0.0164	0.1762	0.6963	0.2336

Table 20. Comparison of the best of the fitness values

Data set	Zoo	Vehicle	Sonar	Heart	Ionosphere	Breast
GWO	0.0586	0.7243	0.0957	0.1852	0.7559	0.2336
WOA	0.0574	0.7208	0.1018	0.1856	0.7553	0.2336
GOOSE	0.0992	0.7631	0.0948	0.2079	0.7667	0.2532
GJO	0.0381	0.6996	0.0416	0.1864	0.7037	0.2368
SBOA	0.0486	0.6989	0.0496	0.1799	0.6989	0.2379
MSBOA	0.0341	0.6978	0.0344	0.1773	0.6963	0.2336

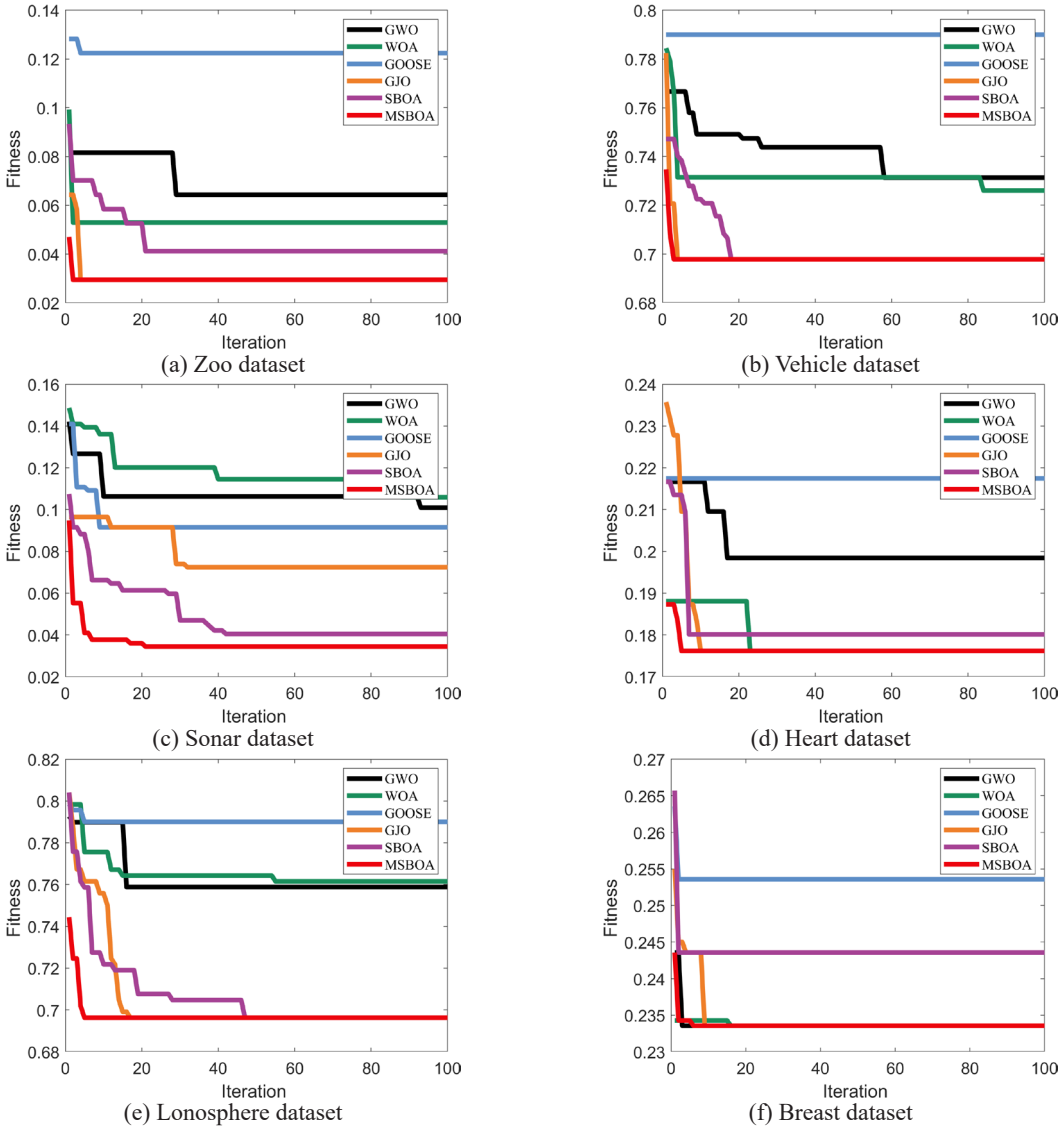


Fig. 4. Iterative evolutionary curve for algorithms

7 Conclusions

This research introduces an innovative optimization approach using MSBOA. The notable performance improvement is credited to integrating golden section search strategies and differential evolution techniques. Various

benchmark tests, real-world engineering challenges, and six different UCI standard feature selection datasets have been used to rigorously evaluate the effectiveness of MSBOA. The MSBOA has excelled in the CEC2021 benchmark suite, surpassing other algorithms such as the SBOA, WOA, GWO, GJO, and GOOSE. It has offered superior or competitive solutions in practical engineering optimization scenarios, including the design of gear trains, pressure vessels, welded beam structures, step-cone pulleys, Robot gripper problems, and industrial refrigeration systems. Experimental results show that MSBOA is stable in solving multi-constraint engineering problems. Feature selection experiments conducted on six UCI standard datasets demonstrated that MSBOA outperforms both the standard SBOA and other comparative algorithms. Statistical analysis confirms that MSBOA improves both the confidence of achieving global optimality and the robustness of the results. While the MSBOA has shown remarkable efficacy, it is crucial to recognize its limitations. The computational complexity of the algorithm may present obstacles when scaling to larger problems. Additionally, parameter fine-tuning is non-trivial and necessitates careful consideration to achieve optimal outcomes.

8 Acknowledgement

This research was funded by the Natural Science Research Project of Anhui Province University, China (Grant No. KJ2021A1165), the Academic Funding Program for Top Talents in Higher Education Disciplines (Grant No. gxbjZD2022086), and the Research Team Project of Anhui Xinhua University (Grants Nos. Kytd201904 and Kytd202204).

References

[1] Y. Djenouri, M. Comuzzi, Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem, Information Sciences 420(2017) 1-15. https://doi.org/10.1016/j.ins.2017.08.043

[2] H.M. Song, M.H. Sulaiman, M.R. Mohamed, An Application of Grey Wolf Optimizer for Solving Combined Economic Emission Dispatch Problems, International Review on Modelling and Simulations 7(5)(2014) 838-844. https://doi.org/10.15866/iremos.v7i5.2799

[3] Z. Peng, P. Pirozmand, Y. Xiong, Improved Harris Hawks Optimizer algorithm to solve the multi-depot open vehicle routing problem, Evolutionary Intelligence 17(4)(2024) 2495-2513. https://doi.org/10.1007/s12065-023-00898-0

[4] R.K. Hamad, T.A. Rashid, GOOSE algorithm: a powerful optimization tool for real-world engineering challenges and beyond, Evolving Systems 15(4)(2024) 1249-1274. https://doi.org/10.1007/s12530-023-09553-6

[5] N. Chopra, M.M. Ansari, Golden jackal optimization: A novel nature-inspired optimizer for engineering applications, Expert Systems with Applications 198(2022) 116924. https://doi.org/10.1016/j.eswa.2022.116924

[6] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software 95(2016) 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[7] M. Kumar, K. Rajwar, K. Deep, Analysis of Marine Predators Algorithm using BIAS toolbox and Generalized Signature Test, Alexandria Engineering Journal 95(2024) 38-49. https://doi.org/10.1016/j.aej.2024.03.060

[8] Y. Fu, D. Liu, J. Chen, L. He, Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems, Artificial Intelligence Review 57(2024) 1-102. https://doi.org/10.1007/s10462-024-10729-y

[9] M. Li, Z. Liu, H. Song, An improved algorithm optimization algorithm based on RungeKutta and golden sine strategy, Expert Systems with Applications 247(2024) 123262. https://doi.org/10.1016/j.eswa.2024.123262

[10] P. Civicioglu, E. Besdok, A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, Artificial Intelligence Review 39(2013) 315-346. https://doi.org/10.1007/s10462-011-9276-0

[11] B. Mohanty, S. Panda, P.K. Hota, Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system, International Journal of Electrical Power & Energy Systems 54(2014) 77-85. https://doi.org/10.1016/j.ijepes.2013.06.029

[12] M. Joshi, K. Kalita, P. Jangir, I. Ahmadianfar, S. Chakraborty, A Conceptual Comparison of Dragonfly Algorithm Variants for CEC-2021 Global Optimization Problems, Arabian Journal for Science and Engineering 48(2)(2023) 1563-1593. https://doi.org/10.1007/s13369-022-06880-9

[13] Y. Wang, C. Sui, C. Liu, J. Sun, Y. Wang, Chicken swarm optimization with an enhanced exploration-exploitation tradeoff and its application, Soft Computing 27(12)(2023) 8013-8028. https://doi.org/10.1007/s00500-023-07990-8

[14] J. Wang, H. Ouyang, S. Li, W. Ding, L. Gao, Equilibrium optimizer-based harmony search algorithm with nonlinear dynamic domains and its application to real-world optimization problems, Artificial Intelligence Review 57(2024) 166. https://doi.org/10.1007/s10462-024-10793-4

[15] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems 97(2019) 849-872. https://doi.org/10.1016/j.future.2019.02.028
[16] B. Jonsson, Z. Barsoum, J.-O. Sperle, Weight optimization and fatigue design of a welded bogie beam structure in a construction equipment, Engineering Failure Analysis 19(2012) 63-76. https://doi.org/10.1016/j.engfailanal.2011.09.006
[17] S. Talatahari, M. Azizi, Optimization of constrained mathematical and engineering design problems using chaos game optimization, Computers & Industrial Engineering 145(2020) 106560. https://doi.org/10.1016/j.cie.2020.106560
[18] Y. Wei, Y. Chen, T. Ren, Q. Chen, C. Yan, Y. Yang, Y. Li, A Novel, Variable Stiffness Robotic Gripper Based on Integrated Soft Actuating and Particle Jamming, Soft Robotics 3(3)(2016) 134-143. https://doi.org/10.1089/soro.2016.0027
[19] B.S. Yildiz, N. Pholdee, S. Bureerat, A.R. Yildiz, S.M. Sait, Robust design of a robot gripper mechanism using new hybrid grasshopper optimization algorithm, Expert Systems 38(3)(2021) e12666. https://doi.org/10.1111/exsy.12666
[20] H. Bayzidi, S. Talatahari, M. Saraee, C.-P. Lamarche, Social Network Search for Solving Engineering Optimization Problems, Computational Intelligence and Neuroscience 2021(1)(2021) 8548639. https://doi.org/10.1155/2021/8548639
[21] S. Arora, M. Sharma, P. Anand, A Novel Chaotic Interior Search Algorithm for Global Optimization and Feature Selection, Applied Artificial Intelligence 34(2020) 292-328. https://doi.org/10.1080/08839514.2020.1712788