IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 February 2023, accepted 19 March 2023, date of publication 28 March 2023, date of current version 4 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3262600

RESEARCH ARTICLE

An Improved Black Widow Optimization Algorithm for Engineering Constrained Optimization Problems

DONGXING XU AND JIANCHUAN YIN
Naval Architecture and Shipping College, Guangdong Ocean University, Zhanjiang 524088, China
Guangdong Provincial Engineering Research Center for Ship Intelligence and Safety, Zhanjiang 524088, China

Corresponding author: Jianchuan Yin (yinjianchuan@gdou.edu.cn)

This work was supported in part by the National Natural Science Foundation of P. R. China under Grant 52271361 and Grant 52231014.

ABSTRACT In solving engineering constrained optimization problems, the conventional black widow optimization algorithm (BWOA) has some shortcomings such as insufficient robustness and slow convergence speed. Therefore, an improved black widow optimization algorithm (IBWOA) is proposed by combining methods of double chaotic map, Cauchy center of gravity inverse difference mutation and golden sine guidance strategy. Firstly, the quality of the initial population of the BWOA is improved based on the double chaotic map; Secondly, in order to make full use of the difference information between the current and the optimal position thus improve optimization accuracy, the golden sine algorithm (Gold-SA) is introduced to update the position of the black widow individuals; Finally, the Cauchy barycenter reverse differential mutation operator is employed to increase the diversity of the population, avoid local optimization thus improve the global search ability of the algorithm. In addition, the global convergence characteristics of the IBWOA are analyzed based on the Markov process and the convergence probability reaches 1 for the globally optimal solution. The performance of the proposed IBWOA was evaluated based on eight continuous / discrete hybrid engineering optimization problems and typical benchmark functions. The results show that the improved BWOA can improve the search accuracy, convergence speed and robustness effectively comparing with some other conventional optimization algorithms.

INDEX TERMS Black widow optimization algorithm, double chaotic map, golden sine algorithm (Gold-SA), Cauchy barycentric reverse difference mutation operator, Markov chain, engineering optimization.

I. INTRODUCTION

In the field of real-world engineering technology, many practical problems can be casted as constrained optimization problems. Optimization is designed for solving real-world engineering optimization problems, such as optimal allocation of resources, job shop scheduling, systems with the largest profits, and the design of complex engineering systems that meet a set of constraints, etc. [1], [2], [3]. Algorithms for solving complex engineering optimization problems can be divided into two categories: (1) deterministic algorithms, such as steepest descent method, Newton method and variable metric method, etc.; (2) non-deterministic algorithms, such as particle swarm optimization, genetic algorithm, ant colony algorithm, etc. In real-world engineering applications, almost all engineering problems contain nonlinearity and complex constraints. The objective function is usually discontinuous and non-differentiable. The deterministic algorithms cannot ensure to find the optimal solution. For the non-deterministic algorithms, it does not require the objective function and constraint to be differentiable, and can obtain the optimal solution with a high probability [4]. It avoids the inherent defects of traditional numerical deterministic optimization methods for solving complex optimization problems.

Meta-heuristic algorithms are a kind of non-deterministic algorithms. Meta-heuristic algorithms are simple and flexible. They have no derivation mechanism but possess the ability to avoid local optimization. They can directly operate on structural objects without the constraints of derivation

The associate editor coordinating the review of this manuscript and approving it for publication was Dost Muhammad Khan.

and functional continuity. They can obtain the optimal solution with a high probability. Therefore, many algorithms have been employed for solving engineering constrained optimization problems, such as golden jackal optimization algorithm [5], alpine skiing optimization [6], niche chimp optimization algorithm [7], novel equilibrium optimizer of Lévy flight and iterative cosine operator [8], improved chaotic Harris hawks optimizer [9], boosting sparrow search algorithm [10], etc.

Since the 1970s, a great variety of natural inspired optimization approaches, like swarm intelligent algorithms, have been widely used in the field of natural scientific and engineering constrained optimization [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. Meta-heuristic algorithms mainly fall into two categories: (1) optimization algorithms that imitate the evolutionary characteristics and living habits of biological systems, such as genetic algorithm (GA) [21], differential evolution algorithm (DE) [22], particle swarm optimization algorithm (PSO) [23], grey wolf optimization algorithm (GWO) [24], moth-flame optimization algorithm (MFO) [25], seagull optimization algorithm (SOA) [26], beetle swarm optimization algorithm (BSO) [27], whale optimization algorithm (WOA) [28], tunicate swarm algorithm (TSA) [29], grasshopper optimization algorithm (GOA) [30], ant colony optimization algorithm (ACO) [31], salp swarm algorithm (SSA) [32], dragonfly algorithm (DA) [33], coyote optimization algorithm (COA) [34], bat-inspired algorithm(BA) [35], artificial bee colony optimization algorithm (ABC) [36], shuffled frog leaping algorithm (SFL) [37], and lion swarm optimization (LSO) [38], etc.; (2) optimization algorithms inspired by different physical laws and plant growth laws, such as artificial algae algorithm (AAA) [39], flower pollination algorithm(FPA) [40], simulated annealing algorithm(SA) [41], gravitational search algorithm(GSA) [42], central force optimization algorithm (CFO) [43], electromagnetism-like mechanism algorithm(EM) [44], artificial physics optimization algorithm(APO) [45], transient search optimization algorithm (TSO) [46] and atom search optimization algorithm (ASO) [47].

In addition, the meta-heuristic algorithms based on swarm intelligence strategy is more exploratory, when compared with the algorithms based on a single solution (e.g., Simulated Annealing Algorithm [41], Beetle Antennae Search Algorithm [48], etc.). Multiple search individuals can share information about the search space to avoid falling into a local optimization solution. In addition, the search process of meta-heuristic algorithms can be divided into two stages: exploration stage and exploitation stage [49]. The exploration stage refers to the exploring of possible areas in the search space as widely as possible. The exploitation stage refers to the local search about the possible areas identified in the exploration stage.

In order to achieve the balance between exploration and exploitation and improve the search ability of the algorithms, the improved optimization algorithms based on different hybrid strategies were proposed [50], [51], [52], [53], [54], [55]. The meta-heuristics algorithms with hybrid strategies can achieve the balance between exploration and exploitation [56]. From another point of view, individuals realize exploration and exploitation through three steps in each iteration: social cooperation, self-adaptation and competition [57]. The model parameters are generally determined by experience. Convergence depends on the model parameters. Therefore, the convergence study is crucial in meta-heuristic algorithms. The commonly used methods for convergence analysis of meta-heuristic algorithms include Markov chain, martingale theory [58] and stochastic functional theory [59]. Markov theory is a stochastic process of universal significance, which has been applied to algorithms of particle swarm optimization [60], artificial bee colony [61], chicken swarm [62], bat algorithm [63] and multivariant optimization [64].

Inspired by the different action strategies of spiders during courtship, A. F. Peña-Delgado, et al. [65] proposed black widow optimization algorithm (BWOA), a novel meta-heuristic optimization algorithm with compact structure. It is used to solve selective harmonic elimination (SHE) equations. The optimization results show that BWOA is reliable and competitive comparing with other meta-heuristic algorithms [65]. However, its application in other fields is rarely reported. There's no optimization algorithm which is suitable for all optimization problems [66]. In other words, a specific meta-heuristic algorithm may achieve ideal results on one set of problems, while may perform unsatisfactory on another set of problems. The development of new meta-heuristic optimization algorithms is still a research hotspot. Although many meta-heuristic algorithms have been proposed, the improvement of their performance is still a challenging task. Like other meta- heuristic algorithms, BWOA also has the defects of insufficient robustness and low convergence speed in solving engineering constrained optimization problems [67].

Following this trend, the conventional BWOA is improved and the key contributions of the resulted improved BWOA (IBWOA) can be defined as follows: 1) the quality of the initial population of the IBWOA is improved by the double chaotic map; 2) the golden sine guidance strategy is employed to improve the position updating mode of IBWOA; 3) the Cauchy barycenter reverse differential mutation operator is used to improve the diversity of the population and improve the global search ability of the algorithm; 4) the global convergence characteristics of the IBWOA are analyzed based on Markov process. The convergence probability of 1 is achieved for the globally optimal solution.

The experiments of eight continuous / discrete mixed engineering optimization problems (e.g., pressure vessel design, welding beam design, tension/compression spring design, etc.) have been conducted and the performance of IBWOA has been evaluated and analyzed.

The rest of this paper is organized as follows. Section II gives brief definition of constrained optimization problems. The basic concepts and principles of BWOA are described in Section III. The improved BWOA with hybrid strategies is proposed in Section IV. Section V discusses the global convergence characteristics of IBWOA based on the Markov process. Section VI presents the experimental results of typical benchmark functions and engineering optimization problems. Finally, Section VII concludes the study and provides the future direction.

II. CONSTRAINED OPTIMIZATION PROBLEMS AND RELATED DESCRIPTIONS

Constrained optimization problems (COPs) can generally be described as follows [68], [69], [70]:

$$\min_{X\in\Omega} f(X) X = (x_1, x_2, \cdots x_m)$$
$$s.t. \begin{cases} g_i(X) \leq 0 & i = 1, 2, \cdots, n \\ h_j(X) = 0 & j = 1, 2, \cdots, q \\ x_k^{lower} \leq x_k \leq x_k^{up} k = 1, 2, \cdots m \end{cases} \tag{1}$$

where $f(X)$ is the objective function; $X \subset \Omega \subseteq S$, X is the solution vector, and S is an m-dimensional rectangle space in R^m defined by the parametric constraints; Ω is the feasible regions which is expressed as:

$$\Omega=\left\{X|g_i(X) \leq 0, i = 1, 2, \cdots, n; h_j(X)= 0, j= 1, 2, \cdots, q\right\}$$

where $g_i(X)$ and $h_j(X)$ are inequality constraints and equality constraints respectively; x_k^{lower} and x_k^{up} are the upper and lower bounds of x_k respectively.

The main purpose of constrained optimization is to search the optimal feasible solution under specific constraints. Metaheuristic algorithm is a global optimization method, which can solve unconstrained optimization problems. Therefore, it is necessary to use constraint processing technology to transform constrained optimization into unconstrained problems. At present, commonly used constraint processing technologies include: (1) penalty functions, (2) multi-objective methods, (3) special representations and operators, (4) repair algorithms, (5) separation of objectives and constraints, (6) hybrid methods [68]. In this paper, penalty function method is used to transform constrained optimization into unconstrained optimization problem. The expression equation is as follows:

$$F(X) = f(X) + P(X)$$
$$s.t. P(X) = \sum_{i=1}^{n} \alpha_i \times \max(0, g_i(X)) + \sum_{j=1}^{q} \rho_j \times \max(0, \left|h_j(X)\right| - \zeta) \tag{2}$$

where $F(X)$ is the fitness function; $P(X)$ is the penalty term for violating the constraint conditions; α_i and ρ_j are the constraint penalty factors respectively; and ζ is the positive tolerance value.

III. THE STANDARD BLACK WIDOW OPTIMIZATION ALGORITHM

The black widow spider is a poisonous spider found from western Canada to southern Mexico. Because only female ones of the type are poisonous, it was named "black widow". Spiders of this type feed on insects such as cockroaches, beetles and butterflies. They weave webs among trees and inhabit forests and swamps. Male spiders judge the mating status of females by sex pheromones. Because females perform cannibalism, males are not interested in hungry or malnourished females. In the conventional BWOA, mathematical models of individual position updating are established depending on different courting strategies and sex pheromone rates of spiders. The standard BWOA algorithm is as follows.

A. MOVEMENT BEHAVIOR

The courtship behavior of black widow spiders on their webs is modeled as linear and spiral movement behavior, as described in Eq. (3):

$$\vec{x}_i(t+1) = \begin{cases} \vec{x}_*(t) - m\vec{x}_{r_1}(t) & if\ rand \leq 0.3 \\ \vec{x}_*(t) - \cos(2\pi\beta)\vec{x}_i(t) & in\ other\ case \end{cases} \tag{3}$$

where $\vec{x}_i(t+1)$ is the new position of the i-th spider, indicating the movement of the i-th spider; $\vec{x}_*(t)$ represents the best search individual in the last iteration; variable m is a floating-point number randomly generated in the interval [0.4, 0.9]; r_1 is a random integer generated in the range [1, PopSize]; $\vec{x}_{r_1}(t)$ represents the r_1 search individual selected, and $i \neq r_{1;}\beta$ is defined as a random float number in the interval [−1.0, 1.0]; $\vec{x}_i(t)$ represents the current search individual; P opSize represents the size of the search agent population; Furthermore, parameters m and β can enable the algorithm to achieve better exploration and exploitation in the iterative process [65].

B. PHEROMONE

Pheromone plays a very important role in the courtship process of black widow spiders. In order to avoid the risk of mating with female spiders who may be hungry, male spiders prefer to avoid cannibalism rather than look for more fertile female spiders. male spiders do not like female spiders with a low pheromone content. In this study, the pheromone content value of black widow spiders is defined in Eq. (4) [65]:

$$pheromone(i) = \frac{fitness_{\max} - fitness(i)}{fitness_{\max} - fitness_{\min}} \tag{4}$$

where $fitness_{\max}$ and $fitness_{\min}$ are the best and worst fitness values ($fitness$ refers to fitness function values) in the current iteration respectively; $fitness(i)$ is the current fitness value of the i-th spider. The pheromone vector is the normalized fitness value in the interval [0,1].

Female spiders with low pheromone levels represent hungry cannibals. If these female spiders exist, they will not be selected, but will be replaced by another one. Therefore, if a pheromone value is equal to or less than 0.3, the spiders'

TABLE 1. The pseudocode of the standard BWOA.

Algorithm 1. The pseudocode of the BWOA for engineering constrained optimization problems

Input population size(*P*opSiz), random initial population position, calculate the fitness value and pheromone value of an individual black widow, and select the global optimal value position $\vec{x}_*(t)$ and the maximum number of iterations (*K*)
for *t=1:K*
Initialize random parameters m, β .
  for *j=1:P*opSize
    if *random<0.3* ***then***
      $\vec{x}_j(t+1) = \vec{x}_*(t) - m\vec{x}_{r_1}(t)$
    else
      $\vec{x}_j(t+1) = \vec{x}_*(t) - \cos(2\pi\beta)\vec{x}_j(t)$
    end if
    if *Pheromone* (*j*)*<0.3*
      Update the black widow individual using Eq. (5).
    end if
  Calculate the fitness of the black widow individual $\vec{x}_j(t+1)$.
    if $f(\vec{x}_j(t+1)) < f(\vec{x}_*(t))$
      $\vec{x}_*(t) = \vec{x}_j(t+1)$
    end if
  end for
  Calculate the pheromone of black widow individuals using Eq. (4).
end for
Output $\vec{x}_*(t)$

strategy can be expressed in Eq. (5):

$$\vec{x}_i(t) = \vec{x}_*(t) + \frac{1}{2}[\vec{x}_{r_1}(t) - (-1)^{\sigma} * \vec{x}_{r_2}(t)] \quad (5)$$

where $\vec{x}_i(t)$ is the updated position of the i-th spider with a low pheromone content; $\vec{x}_*(t)$ represents the optimal position of the population in the last iteration; r_1 and r_2 are random integers in the interval [1, *P*opSize], and $r_1 \neq r_2$; $\vec{x}_{r_1}(t)$ and $\vec{x}_{r2}(t)$ represent the selected individuals r_1 and r_2 respectively; σ is {0,1} binary random number. To sum up, BWOA can obtain competitive results with low optimization parameters [65]. The implementation process is shown in Table 1.

IV. THE IMPROVED BLACK WIDOW OPTIMIZATION ALGORITHM

In view of the shortcomings of the above standard BWOA, such as insufficient robustness and slow convergence speed, an improved BWOA is proposed. The latter incorporates double chaotic map, golden sine search strategy and Cauchy barycenter reverse differential mutation operator, which can effectively avoid the premature convergence of the original BWOA and consequently improve the optimization level of the algorithm.

A. POPULATION INITIALIZATION THROUGH DOUBLE CHAOTIC MAP

The initial diversity of the population can effectively expand the search range of the algorithm and improve the optimization level and convergence speed of the algorithm [71], [72]. In the standard BWOA, the random initial population position would cause the uneven position distribution of black widows, which will affect the development ability of the algorithm. Chaotic motion map can trace all stages through a process without repetition in a certain range, so it has the strengths of randomness, regularity and ergodicity, and can effectively make up for the defects of a random initialization method [73], [74]. At present, many scholars apply conventional chaotic map models like logistic map and sine map to the population initialization of the optimization algorithm, thus avoiding the shortcomings of random population initialization and speeding up the convergence of the algorithm [75], [76]. It can be seen from Fig. 1 that the value frequency is not homogenous over the range [0,1] for logistic and sine maps. The nonuniform traversal of logistic map or sine map will reduce the optimization efficiency of the algorithm. It is unfavorable to identify the optimal position, especially when the global optimal position is not at either end of the search range. Therefore, a new double chaotic map strategy is adopted [77], which is modeled by Eq. (6). It can be seen from Fig. 1 that the initial particles generated through double chaotic map have better diversity and are more evenly distributed in the search space, which can improve the optimization efficiency of the algorithm.

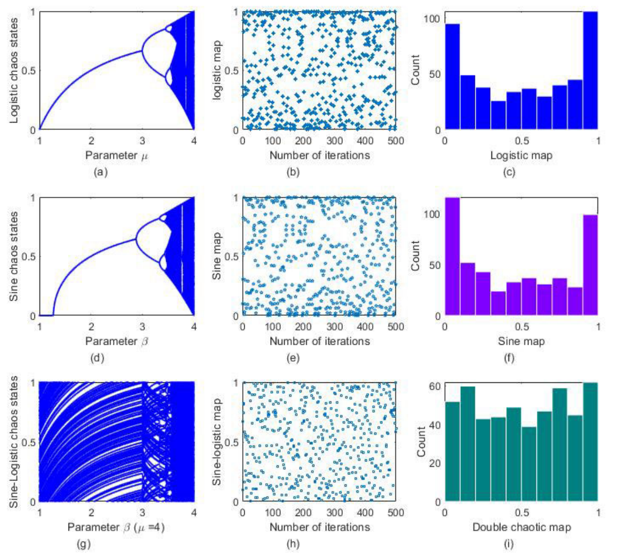

FIGURE 1. Bifurcation diagram and sequence distribution of chaotic map.

$$\begin{cases} x_{k+1} = u \cdot x_k(1 - x_k) \\ y_{k+1} = \dfrac{\omega}{4}\sin(\pi \cdot y_k) \\ z_{k+1} = \mathrm{mod}(x_{k+1} + y_{k+1}, 1) \end{cases} \quad (6)$$

where x_k, y_k and z_k comprise the k-th chaotic number; $\omega \in (0, 4]$ and $u = 4$, and mod () is the remainder function. Because 0, 0.25, 0.5, 0.75, and 1 are breakpoints in the definition domain, these values are not processed during mapping.

Therefore, $x, y, z \in (0, 1)$, $x_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$. The initial position of a black widow individual is indicated by $\{z_k+1\}$ through linear transformation, as shown in Eq. (7).

$$\vec{x}_i = lb_i + (ub_i - lb_i) \times z_{k+1} \tag{7}$$

where ub_i and lb_i are the upper and lower boundaries of the search space respectively.

B. THE GOLDEN SINE GUIDANCE STRATEGY

In order to make full use of the difference information between the spider and the optimal position to gradually approach the optimal solution, the golden sine guidance strategy is introduced to avoid the premature convergence of the original algorithm thus improve the optimization efficiency of the algorithm. The movement of spiders on their web is divided into linear movement and spiral movement. After obtaining information from a possible mate, a black spider moves to the best position $\vec{x}_*(t)$ through spiral movement. The optimal position is the guiding coordinates in the process of spiral motion.

Although it can accelerate the convergence speed of the optimization algorithm at later stages, this strategy tends to make the spider individuals gather rapidly in the search space. It will result in the reduction of population diversity. The probability of falling into local optimization will increase significantly. Therefore, this paper introduces the golden sine algorithm (Golden-SA) to improve spiders' movement strategy. Compared with other meta-heuristic algorithms, Golden-SA uses the golden section coefficient in the process of location updating to explore a search space [78]. It improves the optimization accuracy.

The golden sine search strategy is expressed as follows [78]:

$$\begin{cases} x_i^{t+1} = x_i^t \times |\sin(R_1)| + R_2 \times \sin(R_1) \times \left|\lambda_1 \times P_*^t - \lambda_2 \times x_i^t\right| \\ \lambda_1 = a + (1-\tau) \times b \\ \lambda_2 = (1-\tau) \times a + \tau \times b \\ \tau = \sqrt{5} - 1/2 \end{cases} \tag{8}$$

where P_*^t is the historical best position and x_i^t is the position of the i-th individual at the t-th iteration; R_1 and R_2 are random numbers; Moreover, $R_1 \in [0, 2\pi]$ indicates the distance that the i-th individual moves, while $R_2 \in [0, \pi]$ indicates the direction of the i-th individual's position updating; τ is the golden section number. In Gold-SA, initial default values for a and b are considered to be $-\pi$ and π, respectively. λ_1 and λ_2 are two coefficients obtained after introducing the golden section coefficient. The values of λ_1 and λ_2 are updated as the objective function value changes. These coefficients narrow the search space and allow the current value to approach the target value. After incorporating the golden sine search algorithm, the position updating equation of the black widow spider during its movement on the web is as follows:

$$\vec{x}_i(t+1) \begin{cases} \vec{x}_*(t) - m\vec{x}_{r_1}(t)\ if\ rand \leq 0.3 \\ \vec{x}_i(t) \times |\sin(R_1)| + R_2 \times \sin(R_1) \times \cdots \\ |\lambda_1 \times \vec{x}_*(t) - \lambda_2 \times \vec{x}_i(t)|\ in\ other\ \ cases \end{cases} \tag{9}$$

It can be seen from Eq. (9) that when i-th spider moves to the prospective mate's position on the web, it will exchange information with the optimal individual every time the position is updated. Each spider individual can fully appreciate the difference between itself and the optimal individual.

In addition, the distance and direction of the spider individual's movement can be controlled by adjusting parameters R_1, R_2, λ_1 and λ_2. The search space can be gradually reduced. The convergence speed and optimization efficiency of the algorithm can be improved significantly [78].

C. CAUCHY BARYCENTER REVERSE DIFFERENTIAL MUTATION OPERATOR

The population diversity in the BWOA will decrease sharply at each later stage of iteration. In order to improve the population diversity, expand the search space and prevent the algorithm falling into local optimization, this paper introduces the Cauchy barycenter reverse differential mutation strategy to generate mutated spiders. The reverse mutation of center of gravity [79] is expressed like this: Let $(x_{1j}, x_{2j}, \ldots, x_{Nj})$ be the value of N spiders in the j-th dimension, the number of the population be N and the number of the dimensions be D. Then the center of gravity of the spider population in the j-th dimension is described by Eq. (10). The center of gravity of the population is $Z_g = (Z_1, Z_2, \ldots, Z_j, \ldots, Z_D)$.

$$Z_j = \frac{x_{1j} + x_{2j} + \cdots + x_{Nj}}{N} \tag{10}$$

Let the position of the i-th spider $x_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{iD})$, then the reverse solution of the center of gravity of the j-th dimension corresponding to the i-th spider is as follows:

$$x_{oj} = 2 * cauchy(0, 1) * Z_j - x_{ij} \tag{11}$$

In Eq. (11), in order to prevent the reverse solution from expressing any other individuals and thus affecting the convergence speed of the algorithm, the dynamic coefficient conforming to the standard Cauchy distribution{Cauchy (0,1)} is introduced. The introduction of the Cauchy mutation can effectively avoid position repetition, expand the search space and make the algorithm obtain a stronger global exploration ability. The differential mutation operator is the core of the differential evolution algorithm, which iterates through processes of mutation, crossover and selection [22], [80]. The weighted sum of the difference between any two individual vectors randomly selected in the search space and the individual vector randomly selected by the third individual indicates population mutation. The mutated individual is expressed as follows:

$$x_{new\_i} = x_{r1} + F \cdot (x_{r2} - x_{r3}) \tag{12}$$

where F is the scaling factor; the individual index is the integer of $new\_i \neq r_1 \neq r_2 \neq r_3 \in [1, PopSize]$; x_{r1} and $x_{r2} - x_{r3}$ are the basis vector and the difference vector respectively.

The Cauchy reverse mutation of the center of gravity can maintain the diversity of the population. At the early stage of the iteration, the differences between the individuals of the population are large, and the mutated spider can expand the exploration space. At the later stages of the iteration, the generation of mutated spiders can maintain the diversity of the population. In addition, the differential evolution algorithm takes the difference between two randomly selected individuals as the difference vector, but ignores the direction of the difference vector. Consequently, although the differential mutation operator can improve the search ability of the algorithm to a certain extent, the algorithm tends to fall into local optimality at the later stages of the iteration.

Therefore, combining the advantages of the above two mutation strategies, a new mutation operator is employed which is referred to as Cauchy barycenter reverse differential mutation operator, with the mutated spider near the center of gravity of the contemporary population. The mutation operator is shown in Eq. (13):

$$X_{new} = Z_g + F^*(X_{m_2} - X_{worst}) + F^*(X_{best} - X_{m_1}) \quad (13)$$

where Z_g is the center of gravity of the population; F is the scaling factor, randomly generated in the interval [2, 0]. The two randomly selected individuals and their reverse individuals are sorted according to their fitness value from good to bad as X_{best}, X_{m_1}, X_{m_2} and X_{worst}. Equation (13) shows that its expression is related to the fitness value of the four selected individuals for each mutated spider, starting from the center of gravity of the population and moving in the direction of the optimal individual X_{best}. The strategy of barycenter reverse learning can bring about a reverse solution far from local extreme points, increase the diversity of the population, and thus improve the searching ability of the algorithm and prevent its falling into local optimization. The set of the difference vector is toward X_{best}. The new mutation operator (Cauchy barycenter reverse differential mutation strategy) is therefore not only of randomness, but also gives guidance. When the new hybrid strategies are employed to the algorithm for engineering constrained optimization problems, the IBWOA has a fast convergence speed, and can balance global exploration and local exploitation satisfactorily. The pseudocode of the implementation process of the IBWOA is shown in Table 2.

D. THE TIME COMPLEXITY OF THE IMPROVED BLACK WIDOW OPTIMIZATION ALGORITHM

Without losing generality, let f be any optimization problem and assume that $O(f)$ is the time complexity involved in calculating relevant functional values. Therefore, the time complexity of the BWOA is notated as $O(t_{\text{Max}} \times n_{\text{Sp}} \times f)$. For the IBWOA, there is no new loops nested. According to the multiplication principle and the addition principle of time complexity, the time complexity of the IBWOA is $O(t_{\text{Max}} \times n_{\text{Sp}} \times n_{\text{DE}} \times f)$, where t_{Max} is the maximum number of iterations and n_{Sp} is the number of spiders (population size), n_{DE} is the number of the optimization problem evaluations when the Cauchy barycenter reverse differential mutation operator is executed. The complexity slightly increases comparing with that of the original algorithm. However, the search accuracy, convergence speed and robustness of the algorithm are greatly improved.

TABLE 2. The pseudocode of the improved BWOA.

Algorithm 2 The pseudocode of the improved BWOA for engineering constrained optimization problems

Input population size($PopSiz$), initialize the population position through double chaotic map, calculate the fitness value and pheromone value of a black widow individual, and select the global optimal position $\vec{x}_*(t)$ and the maximum number of iterations (K)
for *t=1:K*
Initialize the random parameter m and related parameters of the golden sine search strategy.
for *j=1:*PopSize
if *random*<0.3 ***then***
$\vec{x}_j(t+1) = \vec{x}_*(t) - m\vec{x}_{r_1}(t)$
else
$\vec{x}_j(t+1) = \vec{x}_j(t) \times |\sin(R_1)| + R_2 \times \sin(R_1) \times \cdots |\lambda_1 \times \vec{x}_*(t) - \lambda_2 \times \vec{x}_j(t)|$
end if
if *Pheromone* (*j*)<0.3
Update the black widow individual using Eq. (5).
end if
Update the black widow individual using the Cauchy barycenter reverse differential mutation operator in Eq. (13).
Calculate the fitness value of a black widow individual $\vec{x}_j(t+1)$.
if $f(\vec{x}_j(t+1)) < f(\vec{x}_*(t))$
$\vec{x}_*(t) = \vec{x}_j(t+1)$
end if
end for
Calculate the pheromone of black widow individuals using Eq. (4).
end for
Output $\vec{x}_*(t)$

E. THE SPACE COMPLEXITY

The space complexity of the IBWOA is the maximum space considered in the initialization process to be used at any time. Therefore, the total space complexity of the IBWOA is $O(n_{\text{Sp}} \times dim)$, where dim is the dimension of the optimization problem.

V. CONVERGENCE ANALYSIS OF THE IMPROVED BLACK WIDOW OPTIMIZATION ALGORITHM

In this paper, Markov process is used to analyze the convergence of IBWOA [64]. According to the pseudocode of the IBWOA, the current state of an individual is related not only to its state at the previous time, but also to the state of the whole population. Therefore, the transfer of a single individual does not conform to the Markov process. However, the transformation process of the state of the whole population is related only to the current state of the population.

Therefore, the transfer process of the population is a Markov process [81]. Because the number of the population and the number of the states of the whole population in the IBWOA are limited in a discrete space, the population sequence is a finite Markov chain. Moreover, the IBWOA adopts the elite retention strategy, so its corresponding Markov process is an absorbing Markov process. In the IBWOA, the space defined by the positions of all black widow individuals is notated as D, and the position of black widow individuals is represented by X, $X \in D$. The state space of Markov chain is G, and the number of the dimensions in the state space is L, $L = |G| = |D|^N$. The set of black widow states at the t-th iteration is expressed as

$$Q(t) = (X_1(t), X_2(t), \ldots, X_N(t)) \tag{14}$$

where $X_i(t) \in D$, N is the number of black widow individuals in the group with $N < \infty$.

Definition 1: The global optimal solution set of the optimization problem is

$$G^* = \{X^* | \forall X \neq X^*, f(X) > f(X^*)\} \tag{15}$$

and let $G(Q(t)) = |Q(t) \cap G^*|$ represent the numberof optimal solutions contained in the black widow population.

Definition If $\lim_{t\to\infty} P(G(Q(t)) > 0 | Q(0) = Q_0) = 1$ exists for any initial state Q_0, the algorithm converges to the global optimal solution with a probability of 1.

Theorem 1: The number of optimal solutions in the black widow population is monotonic and non-decreasing. That is, for any $\forall t \geq 0$, there is

$$\lim_{t\to\infty} P(G(Q(t+1)) < m | G(Q(t) = m)) = 0, m \geq 0 \tag{16}$$

Proof: Because the BWOA adopts the optimization preserving technique, the optimal black widow position is reserved. Therefore, the optimal solution of the old black widow population cannot be eliminated in the new population. That is, when the number of optimal solutions in the population is $m \geq 0$ at the t-th iteration, the probability that the number of optimal solutions in the population is less than m is 0 at the $(t+1)$-th iteration.

Theorem 2: The IBWOA may find the global optimal solution at any time. That is,

$$P(G(Q(t+1)) > 0 | G(Q(t) = 0)) > 0, \forall t \geq 0 \tag{17}$$

Proof: According to the search strategy of the IBWOA, the black widow population is generated by the moving behavior of black widows, the behavior of avoiding cannibalism and Cauchy barycenter reverse differential mutation strategy in the search space. Therefore, the probability that the black widow individual is any possible solution at any time is not equal to 0, the probability that the black widow individual is the global optimal solution at any time is not equal to 0. Therefore, when the number of optimal solutions in the old population is equal to 0, the probability that the number of optimal solutions in the new population is not equal to 0 is greater than 0.

Theorem 3: After the Cauchy barycenter reverse differential mutation strategy, the IBWOA that retains the optimal solution in each generation converges to the global optimal solution with a probability of 1. That is,

$$\lim_{t\to\infty} P(G(Q(t)) > 0) = 1 \tag{18}$$

Proof: Let the probability that the number of optimal solutions in the population is i at t-th iteration be

$$P_i(t) = P(G(Q(t)) = i)$$

According to Bayesian conditional probability formula:

$$\begin{aligned} P_0(t+1) = P(G(Q(t+1)) = 0 | G(Q(t)) = 0) \\ \times P(G(Q(t)) = 0) + P(G(Q(t)) \neq 0) \\ \times P(G(Q(t+1)) = 0 | G(Q(t)) \neq 0) \end{aligned} \tag{19}$$

According to **Theorem 1:**

$$\begin{aligned} P(G(Q(t+1)) = 0 | G(Q(t)) \neq 0) = 0 (20) \\ \Rightarrow P_0(t+1) = P(G(Q(t+1)) \\ = 0 | G(Q(t)) = 0) \times P_0(t) \end{aligned} \tag{20}$$

According to **Theorem 2:**

$$P(G(Q(t+1)) > 0 | G(Q(t)) = 0) > 0 \tag{21}$$

Let:

$$\zeta = \min(P(G(Q(t+1)) > 0 | G(Q(t)) = 0)),$$

That is,

$$\begin{aligned} & 0 < \zeta \leq P(G(Q(t+1)) > 0 | G(Q(t)) = 0) \leq 1 \\ & \Rightarrow P(G(Q(t+1)) = 0 | G(Q(t)) = 0) \\ & = 1 - P(G(Q(t+1)) \neq 0 | G(Q(t)) = 0) \\ & = 1 - P(G(Q(t+1)) > 0 | G(Q(t)) = 0) \\ & \leq 1 - \zeta < 1 (t = 0, 1, \cdots) \\ & \Rightarrow P_0(t+1) \leq (1-\zeta) P_0(t) \leq \cdots \leq (1-\zeta)^{t+1} P_0(0) \end{aligned} \tag{22}$$

Therefore, when t tends to infinity, it can be concluded that:

$$\begin{aligned} 0 \leq P_0(t+1) \leq (1-\zeta)^{t+1} P_0(0) = 0 \\ \Rightarrow \lim_{t\to\infty} P_0(t-1) = 0 \end{aligned}$$

It can be seen from the above:

$$\begin{aligned} \lim_{t\to\infty} P(G(Q(t+1)) > 0) = 1 - \lim_{t\to\infty} P(G(Q(t+1)) = 0) \\ = 1 - \lim_{t\to\infty} P_0(t+1) = 1 \\ \Rightarrow \lim_{t\to\infty} P(G(Q(t)) > 0) = 1 \end{aligned} \tag{23}$$

That is, as time tends to infinity, IBWOA algorithm can search for the global optimal solution with a probability of 1, and the theorem is proved.

TABLE 3. Optimization results of typical benchmark functions.

Function Name	Algorithm	Worst	Best	Mean	Std
Step Function	DA	10104.55586	178.0757025	1664.108133	2032.012325
	MFO	10105.54417	0.579179176	1354.42324	3490.118086
	SCA	105.9863553	4.410170293	17.07908513	21.43624641
	BWO	7.388150767	4.387971065	6.480335564	0.628539163
	IBWO	**0.776182567**	**0.004010437**	**0.217404086**	**0.201138463**
Generalized Penalized Function	DA	2430744.007	6.116721535	174325.4196	498122.6065
	MFO	256000023.6	1.791206878	8535496.227	46738588.7
	SCA	621782.5817	0.703720971	37961.75397	129761.3351
	BWO	1.3431488	0.497917507	0.836236611	0.22337905
	IBWO	**0.026157166**	**9.4572E-06**	**0.005230503**	**0.006652286**

VI. EXPERIMENT OF ENGINEERING CONSTRAINED OPTIMIZATION PROBLEMS

A. TYPICAL BENCHMARK FUNCTIONS

In order to verify the effectiveness and advantage of the IBWOA, typical benchmark functions of *Step Function* and *Generalized Penalized Function*, are selected respectively [54]. *Step Function* is single-mode function, which can evaluate the exploitation ability of the algorithm. *Generalized Penalized Function* is multi-mode function, which can evaluate the exploration ability of the algorithm. The equations are as follows.

1)*Step Function*(F_1):

$$F_1(x) = \sum_{i=1}^{n} (x_i + 0.5)^2 \tag{24}$$

where dimension is 30, variable range is [−100, 100], and optimal value is zero.

2) *Generalized Penalized Function*(F_2):

$$F_2(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$$

$$y_i = 1 + \frac{x_i + 4}{4},\ u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0\ a > x_i > -a \\ k(-x_i - a)^m x_i < a \end{cases} \tag{25}$$

where dimension is 30, variable range is [−50, 50], and optimal value is zero.

The simulation results of the IBWOA are compared with those of the dragonfly algorithm (DA) [33], moth-flame optimization algorithm (MFO) [25], sine cosine algorithm (SCA) [82] and BWOA [65]. The population size of each algorithm was set at 30, the maximum number of iterations was 500. Other parameters were set at the default values of the algorithms. In order to avoid the contingency of optimization results, each benchmark function was run on each algorithm for 30 times.

The *worst*, *best*,and *mean* objective function values, and the standard deviation (*Std*) for each test problem are designated to be the evaluation index to verify convergence accuracy and robustness for the IBWOA. It is shown in Table 3 that the values obtained from the IBWOA outperform the DA, MFO, SCA and BWO results regarding the *worst*, *best*, and *mean* values and the standard deviation for the single-mode functions F_1(*Step Function*) and the multi-mode functions F_2(*Generalized Penalized Function*). It can be seen from Fig. 2 that the convergence curve of IBWOA decreases faster than the other algorithms. It shows that IBWOA outperforms the DA, MFO, SCA and BWO regarding convergence speed and optimization ability. Therefore, when optimizing single-mode and multi-mode functions, IBWOA has better performance in terms of convergence accuracy, convergence speed and robustness.

B. ENGINEERING CONSTRAINED OPTIMIZATION PROBLEMS

In this paper, the IBWOA is used to solve the optimization design problems in real-world engineering [5], [83]. Moreover, the results are compared with those given by butterfly optimization algorithm(BOA) [84], transient search optimization (TSO) [46], dragonfly algorithm(DA) [33] and BWOA [65]. It further shows the superiority of the IBWOA in solving real-world engineering problems. Each algorithm runs for 30 times independently and the number of iterations is 300. All algorithm parameters are set to the default values in the original literature.

1) CANTILEVER BEAM DESIGN PROBLEM

This problem is a structural engineering design example related to the weight optimization of a square section cantilever beam. One end of the beam is rigidly supported, and the vertical force acts on the free node of the cantilever, which is shown in Fig. 3. The beam is composed of five hollow square blocks with constant thickness. Its height (or width) is a decision variable, and its thickness remains unchanged as 2/3. This problem can be expressed by the following analytical equation:

Objective function:

$$f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \tag{26}$$

Subject to:

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$$

Boundary condition:

$$0.01 \leq x_i \leq 100, i = 1, \cdots, 5.$$

The best solutions for cantilever beam design problem obtained by IBWOA and other methods are listed in Table 4. It can be seen that the IBWOA provides better solution than those provied by other methods. The statistical results of IBWOA and other methods are listed in Table 5. For

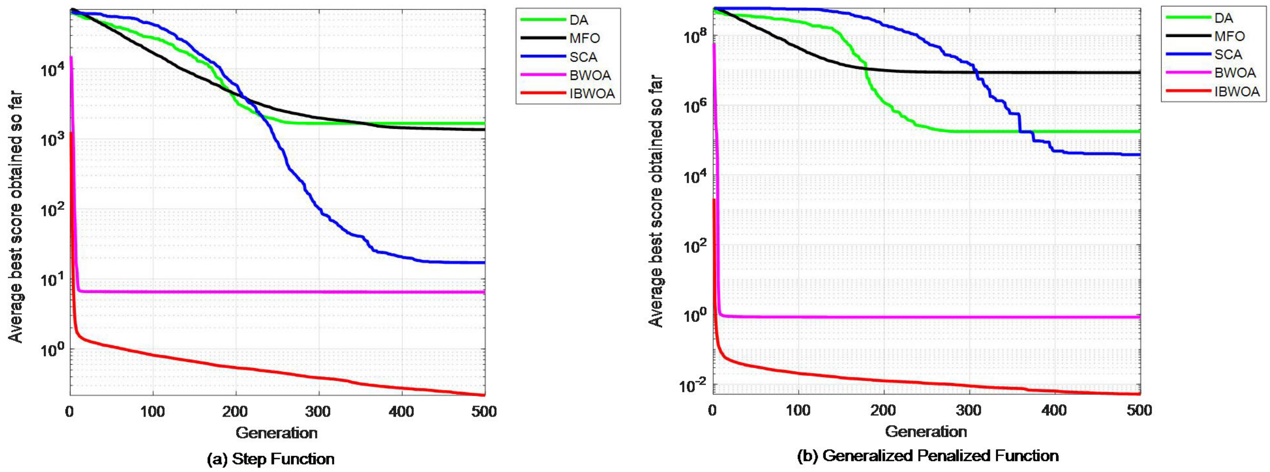

FIGURE 2. **Average fitness value convergence curves of typical benchmark functions.**

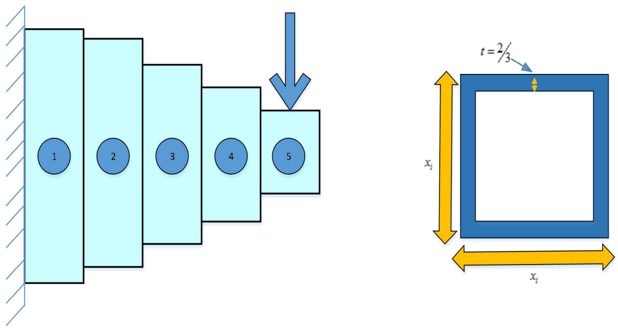

FIGURE 3. **Schematic diagram of cantilever beam.**

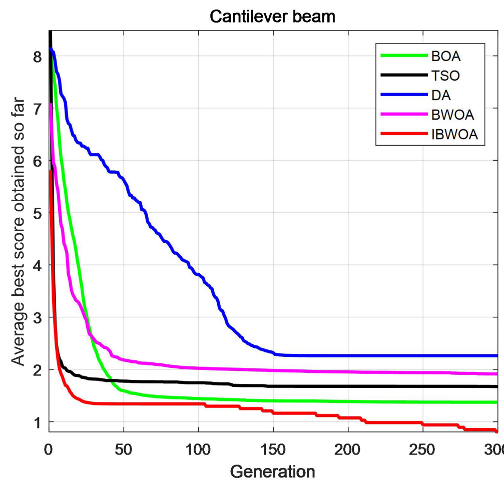

FIGURE 4. **Average fitness curve of different algorithms for solving cantilever beam problem.**

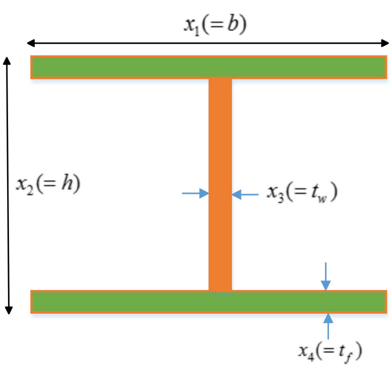

FIGURE 5. **Schematic diagram of I-beam.**

cantilever beam design problem, the average value ranking is IBWOA> BOA> TSO> BWOA> DA; The standard deviation ranking is IBWOA> BOA> TSO> BWOA> DA. It can be seen that IBWOA outperforms other algorithms in respect of the worst value, optimal solution, average value and standard deviation.

The results show that the convergence accuracy and stability of IBWOA outperform BOA, TSO, BWOA, and DA. Although the optimal solution of DA outperforms the other three algorithms, the robustness of DA is the worst. The BOA is relatively stable in solving the cantilever beam design problem. BOA has competitive results compared with TSO, DA and BWOA. The fitness curves of different algorithms for solving cantilever beam problems are given in Fig. 4. It can be seen that the IBWOA outperforms the four compared algorithms regarding convergence speed. Therefore, the searching ability of IBWOA outperforms other algorithms in solving the cantilever beam design problem.

2) I-BEAM DESIGN PROBLEM

Another typical engineering optimization problem is the design of I-beam. The purpose is to minimize the vertical deflection of the beam, while meeting the cross-sectional area and stress constraints under a given load, which is shown in Fig. 5. Flange width $b(= x_1)$, section height $h(= x_2)$, web thickness $t_w(= x_3)$ and web thickness flange $t_f(= x_4)$ are variables of I-beam problem. When the length (L) and elastic modulus (E) of the beam are 5200 cm and 523.104 kN/cm^2 respectively, the maximum vertical deflection of the beam is

TABLE 4. Optimal results of cantilever beam design problem.

Algorithms	Variables					Constraint Condition	
	x_1	x_2	x_3	x_4	x_5	$g_1(x)$	$f(x)$
BOA	6.22373849	4.913533583	4.76547206	3.450466335	2.319875518	−0.009007195	1.352400566
TSO	5.773883475	6.365765694	4.375453381	3.851536113	1.739391428	−0.000301831	1.379416278
DA	5.984872872	5.249215807	4.711539201	3.391073538	2.165581948	0	1.341742482
BWOA	5.894118984	5.170788886	4.799692457	3.47454337	2.185794544	2.87423E−06	1.343184889
IBWOA	6.011447674	5.309421625	4.494122494	3.504642558	2.154042343	−3.63021E−07	**1.3399576**

TABLE 5. Statistical results of cantilever beam design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	1.401086047	1.372017145	1.352400566	0.013158386
TSO	2.114970896	1.671958618	1.379416278	0.197220276
DA	7.298200057	2.261830173	1.341742482	1.986101536
BWOA	3.762470102	1.913088951	1.343184889	0.600229728
IBWOA	**1.340057121**	**1.339963419**	**1.3399576**	**1.81246E−05**

$f(x) = PL^3/48EI$. The objective function and constraints of the problem are as follows:

Objective function:

$$f(X) = \frac{5000}{x_3(x_2-2x_4)^3/12 + (x_1x_4^3/6) + 2bx_4(x_2-x_4/2)^2} \tag{27}$$

Subject to:

$$g_1(X) = 2x_1x_3 + x_3(x_2-2x_4) \leq 300,$$

$$g_2(X) = \frac{18x_2 \times 10^4}{x_3(x_2-2x_4)^3 + 2x_1x_3(4x_4^2+3x_2(x_2-2x_4))} + \frac{15x_1 \times 10^3}{(x_2-2x_4)x_3^2 + 2x_3x_1^3} \leq 56,$$

Boundary condition:

$$10 \leq x_1 \leq 50,$$
$$10 \leq x_2 \leq 80,$$
$$0.9 \leq x_3 \leq 5,$$
$$0.9 \leq x_4 \leq 5.$$

The best solutions for I-beam design problem obtained by IBWOA and other methods are listed in Table 6. It can be seen that IBWOA, BWOA and DA yield the best results in dealing with this problem excepted BOA and TSO. The statistical results of IBWOA and other methods listed in Table 7 show that IBWOA achieves better results than other methods in terms of worst value, optimal solution, mean value and standard deviation. In other words, the convergence accuracy and robustness of IBWOA outrank BOA, TSO, BWOA, and DA. In the case of similar optimization results, the worst value, mean value and standard deviation obtained by DA are 0.013332052, 0.013100482 and 6.1866E−05 respectively. It outranks the BOA, TSO and BWOA. Therefore, compared with BOA, TSO and BWOA, DA has certain advantages in solving I-beam problem.

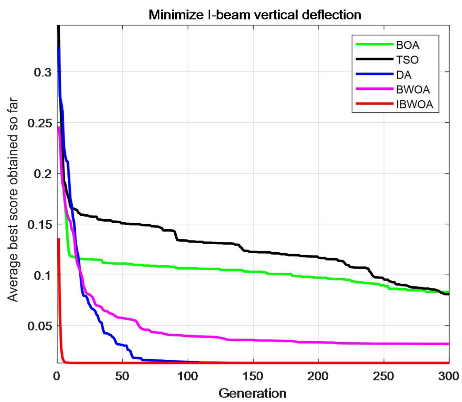

FIGURE 6. Average fitness curve of different algorithms for solving I-beam problem.

In addition, the fitness curves of different algorithms for solving I-beam problems are shown in Fig. 6. It can be noticed that the convergence speed of IBWOA outperforms other algorithms. It further shows that IBWOA is a good alternative for solving I-beam optimization problem.

3) THREE-BAR TRUSS DESIGN PROBLEM

This example considers the Three-bar planar truss structure, which is shown in Fig. 7. Subjecting to stress (σ) constraints on each of the truss members, the volume of the statically loaded three-bar truss should be minimized. The objective is to evaluate the optimal cross-sectional areas, $A_1(= x_1)$ and $A2(= x_2)$. The objective function and constraints of the problem are as follows:

Objective function:

$$f(X) = (2\sqrt{2}\,x_1 + x_2) \times l \tag{28}$$

Subject to:

$$g_1(X) = \frac{\sqrt{2}\,x_1 + x_2}{\sqrt{2}\,x_1^2 + 2x_1x_2}P - \sigma \leq 0,$$

TABLE 6. Optimal results of I-beam design problem.

Algorithms	Variables				Constraint Condition		
	x_1	x_2	x_3	x_4	$g_1(x)$	$g_2(x)$	$f(x)$
BOA	76.67057529	36.41227086	1.10177914	2.935454035	−8.221308704	−0.516374749	0.015465429
TSO	79.87215724	40.14625984	0.954504399	2.854171699	−0.041676099	−1.153104918	0.013424752
DA	80	50	0.9	2.321792261	0	−1.570228476	**0.013074119**
BWOA	80	50	0.9	2.321792254	−6.22017E−07	−1.570228465	**0.013074119**
IBWOA	80	50	0.9	2.321792261	0	−1.570228476	**0.013074119**

TABLE 7. Statistical results of I-beam design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	0.268408415	0.083271048	0.015465429	0.076690198
TSO	0.407247701	0.08081661	0.013424752	0.113224379
DA	0.013332052	0.013100482	0.013074119	6.1866E−05
BWOA	0.38535359	0.031975731	0.013074119	0.071765512
IBWOA	**0.013074119**	**0.013074119**	**0.013074119**	**8.8219E−18**

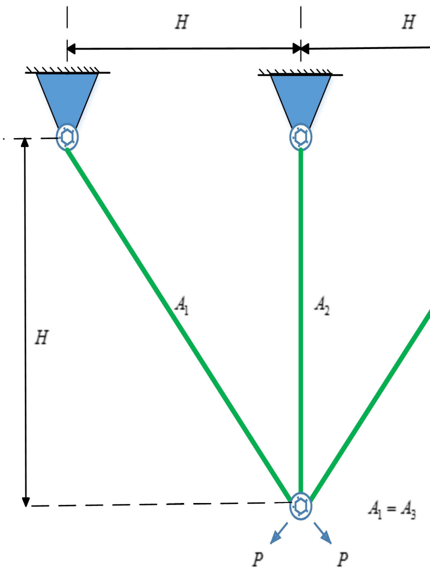

FIGURE 7. Schematic diagram of three-bar truss.

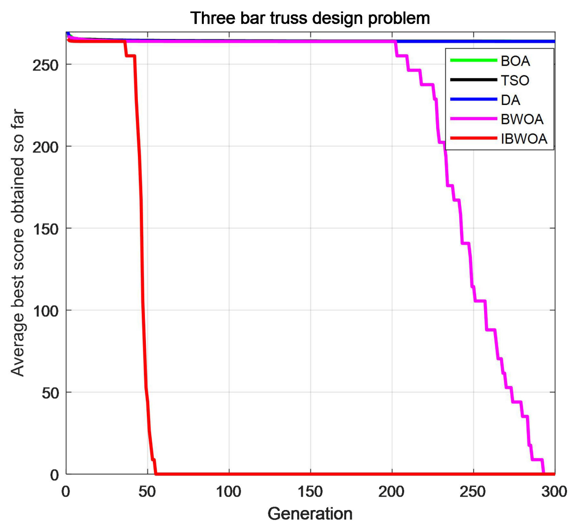

FIGURE 8. Average fitness curve of different algorithms for solving three-bar truss problem.

$$g_2(X) = \frac{x_2}{\sqrt{2}\,x_1^2 + 2x_1x_2}P - \sigma \le 0,$$
$$g_3(X) = \frac{1}{\sqrt{2}\,x_2 + x_1}P - \sigma \le 0,$$
$$l = 100\ cm,$$
$$P = 2kN/cm^3,$$
$$\sigma = 2kN/cm^3,$$

Boundary condition:

$$0 \le x_1, x_2 \le 1.$$

The best solutions for Three-bar Truss design problem obtained by IBWOA and other methods are listed in Table 8. It can be seen that IBWOA and BWOA provide better results than other algorithms. However, from the statistical results of IBWOA and other methods listed in Table 9, IBWOA and BWOA outperform the other methods in respect of the worst value, optimal solution, mean value and standard deviation. According to the fitness curves of different algorithms for solving the three-bar truss problem in Fig.8, the convergence speed of IBWOA outranks the other algorithms in the case of similar optimization results. The IBWOA performs better in the Three-bar Truss design problem.

4) SPEED REDUCER DESIGN PROBLEM

In the mechanical system, the reducer is one of the important components of the gearbox, which can be used in a variety of applications. In this optimization problem, the weight of the reducer will be minimized under 11 constraints, which

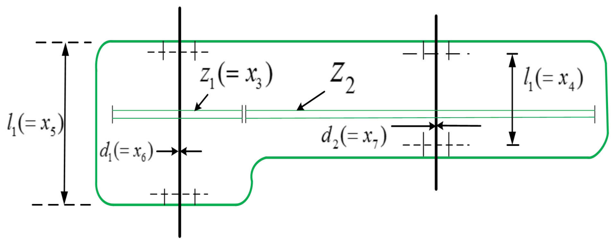

FIGURE 9. Schematic diagram of speed reducer.

is shown in Fig.9. There are seven variables in this problem, which are tooth width $b(= x_1)$, gear modulus $m(= x_2)$, number of teeth in pinion $z(= x_3)$, length of the first shaft between bearings $l_1(= x_4)$, length of the second shaft between bearings $l_2(= x_5)$, diameter of the first shaft $d_1(= x_6)$ and diameter of the second shaft $d_2(= x_7)$. The mathematical equation of this problem is as follows:

Objective function:

$$\begin{aligned} f(X) = &\ 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3) \\ &- 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) \\ &+ 0.7854(x_4x_6^2 + x_5x_7^2), \end{aligned} \tag{29}$$

Subject to:

$$\begin{aligned} g_1(X) &= \frac{27}{x_1x_2^2x_3} - 1 \le 0, \\ g_2(X) &= \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0, \\ g_3(X) &= \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0, \\ g_4(X) &= \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0, \\ g_5(X) &= \frac{\sqrt{(745x_4/x_2x_3)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \le 0, \\ g_6(X) &= \frac{\sqrt{(745x_5/x_2x_3)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \le 0, \\ g_7(X) &= \frac{x_1x_2}{40} - 1 \le 0, \\ g_8(X) &= \frac{5x_2}{x_1} - 1 \le 0, \\ g_9(X) &= \frac{x_1}{12x_2} - 1 \le 0, \\ g_{10}(X) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0, \\ g_{11}(X) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0, \end{aligned}$$

Boundary condition:

$$\begin{aligned} 2.6 &\le x_1 \le 3.6, \\ 0.7 &\le x_2 \le 0.8, \\ x_3 &\in \{17, 18, 19, \cdots, 28\} \\ 7.3 &\le x_4, x_5 \le 8.3 \\ 2.9 &\le x_6 \le 3.9, \\ 5 &\le x_7 \le 5.5. \end{aligned}$$

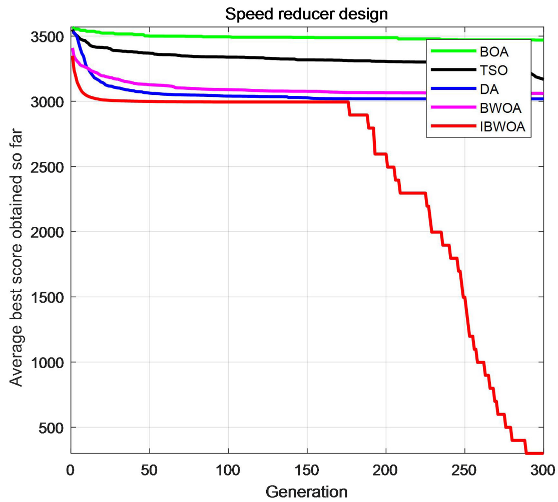

FIGURE 10. Average fitness curve of different algorithms for solving speed reducer problem.

The best solutions for speed reducer design problem obtained by IBWOA and other methods are listed in Table 10. It can be seen that the solution obtained by IBWOA outranks other methods. According to the statistical results of IBWOA and other methods listed in Table 11, the IBWOA outperforms the other methods in terms of the worst value, optimal solution, mean value and standard deviation. Therefore, optimization ability and stability of IBWOA outperforms the other methods. For speed reducer design problem, the mean value, optimal value and standard deviation obtained by DA are second only to IBWOA. By comparing the results, DA algorithm has better optimization accuracy and stability than BOA, TSO and BWOA when solving the speed reducer design problem.

It can be seen from the fitness curves of different algorithms given in Fig. 10 that the convergence speed of IBWOA outranks other algorithms. The advantages of IBWOA in solving speed reducer design problem are further verified.

5) PISTON LEVER DESIGN PROBLEM

The main purpose of this problem is to position the piston lever components $H(= x_1)$, $B(= x_2)$, $D(= x_3)$ and $X(= x_4)$ by minimizing the oil volume when the piston rod is raised from 0° to 45°, as shown in Fig. 11. The expression of this problem is as follows:

Objective function:

$$f(X) = \frac{1}{4}\pi x_3^2(L_2 - L_1) \tag{30}$$

Subject to:

$$\begin{aligned} g_1(X) &= QL\cos\theta - R \times F \le 0, \\ g_2(X) &= Q(L - x_4) - M_{\max} \le 0, \\ g_3(X) &= 1.2(L_2 - L_1) - L_1 \le 0, \end{aligned}$$

TABLE 8. Optimal results of three-bar truss design problem.

Algorithms	Variables		Constraint Condition			
	x_1	x_2	$g_1(x)$	$g_2(x)$	$g_3(x)$	$f(x)$
BOA	0.78831651	0.407502317	0.001335146	−1.46428312	−0.534381734	263.9067322
TSO	0.787845936	0.410555275	3.2781E−05	−1.461465394	−0.538501825	263.8965984
DA	0.788688552	0.408210341	2.22045E−15	−1.464144757	−0.535855243	263.8958435
BWOA	0.788672111	0.408256842	2.04695E−10	−1.464091894	−0.535908106	**263.8958434**
IBWOA	0.788674283	0.408250697	1.56422E−09	−1.464098879	−0.53590112	**263.8958434**

TABLE 9. Statistical results of three-bar truss design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	264.6913081	264.0414735	263.9067322	0.166108421
TSO	264.0031506	263.916243	263.8965984	0.025029295
DA	263.9032147	263.8965675	263.8958435	0.001466103
BWOA	**263.8958434**	**263.8958434**	**263.8958434**	**5.78152E−14**
IBWOA	**263.8958434**	**263.8958434**	**263.8958434**	**5.78152E−14**

TABLE 10. Optimal results of speed reducer design problem.

Algorithms/Variables	BOA	TSO	DA	BWOA	IBWOA
x_1	3.181294239	3.389714838	3.460738507	3.499893373	3.5
x_2	0.7	0.7	0.7	0.7	0.7
x_3	17.20798128	17	17	17	17
x_4	8.053298393	7.323023298	7.3	7.500648965	7.3
x_5	7.748952307	7.686544814	7.68350518	7.975406357	7.715319911
x_6	3.486937336	3.364256067	3.350540949	3.350917366	3.350540949
$g_1(x)$	5.307713674	5.290398615	5.286643739	5.286743596	5.286654465
$g_2(x)$	0.006546679	−0.043784898	−0.063409006	−0.073887066	−0.07391528
$g_3(x)$	−0.138852859	−0.171905222	−0.188899956	−0.197974093	−0.197998527
$g_4(x)$	−0.433924777	−0.502806528	−0.499367306	−0.457184805	−0.499367306
$g_5(x)$	−0.906064801	−0.905973503	−0.905817908	−0.894678796	−0.904643905
$g_6(x)$	−0.111802961	−0.012142287	4.44E−16	2.83E−06	−5.61E−13
$g_7(x)$	−0.011867146	−0.002127169	−9.99E−16	1.21E−07	−2.09E−12
$g_8(x)$	−0.698860328	−0.7025	−0.7025	−0.7025	−0.7025
$g_9(x)$	0.100181164	0.032535233	0.011344831	3.05E−05	−1.83E−13
$g_{10}(x)$	−0.621274495	−0.596462519	−0.588007321	−0.583346027	−0.583333333
$g_{11}(x)$	−0.114598062	−0.0514322	−0.051258709	−0.076563097	−0.051258709
$f(x)$	3105.562866	3005.501037	2996.714999	3002.064865	**2994.424466**

$$g_4(X) = \frac{x_3}{2} - x_2 \leq 0, \textit{ where}$$
$$R = \frac{|-x_4(x_4 \sin\theta + x_1) + x_1(x_2 - x_4\cos\theta)|}{\sqrt{(x_4 - x_2)^2 + x_1^2}},$$
$$F = \frac{\pi P x_3^2}{4},$$
$$L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2},$$
$$L_2 = \sqrt{(x_4 \sin\theta + x_1)^2 + (x_2 - x_4\cos\theta)^2},$$
$$\theta = 45^0,$$
$$Q = 10000\ lbs,$$
$$L = 240\ in,$$
$$M_{\max} = 1.8 \times 10^6 lbs\ in,$$
$$P = 1500\ psi,$$

Boundary condition:

$$0.05 \leq x_1, x_2, x_4 \leq 500,$$
$$0.05 \leq x_3 \leq 200.$$

The best solutions for piston lever design problem obtained by IBWOA and other methods are listed in Table 12. It can be seen that the IBWOA provides the best solution. According to the statistical results of IBWOA and other methods listed in Table 13, the IBWOA outperforms other methods in respect of the worst value, optimal solution, mean value and standard deviation. Therefore, the convergence accuracy and stability of IBWOA outranks those of BOA, TSO, BWOA, and DA. For piston lever design problem, the worst value, optimal value, mean value and standard deviation obtained by DA are 519.9923802, 211.1498409, 8.412698323 and 163.9011095, respectively, which are second only to IBWOA. Compared with the other three algorithms, DA provides the competitive results.

From the fitness curves of different algorithms for solving the piston lever problem given in Fig. 12, it can be seen that the convergence speed of IBWOA outranks that of other algorithms. The robustness and convergence speed of IBWOA is the best. It is an effective tool to solve the piston lever design problem.

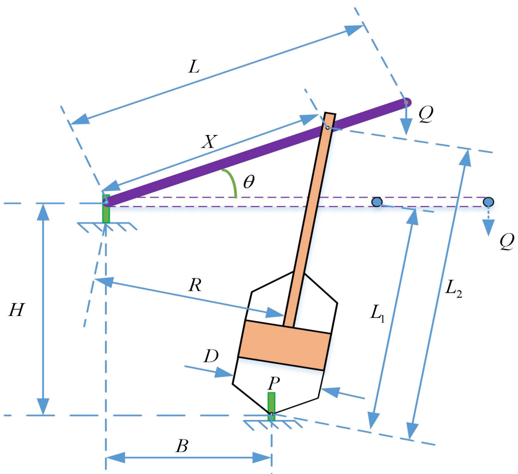

FIGURE 11. Schematic diagram of piston lever.

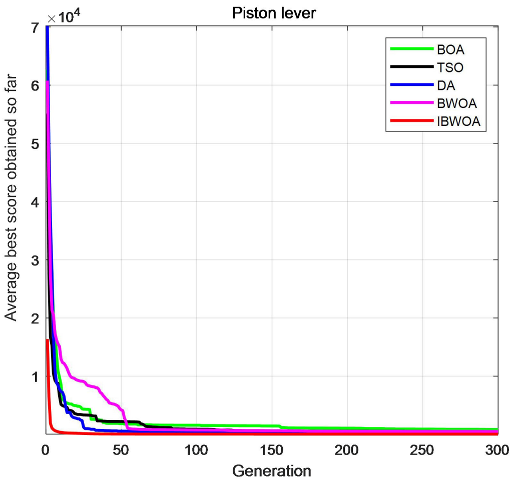

FIGURE 12. Average fitness curve of different algorithms for solving Piston lever problem.

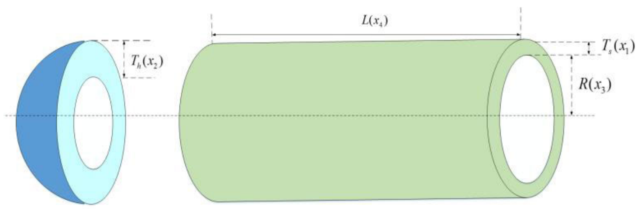

FIGURE 13. Schematic diagram of pressure vessel.

6) PRESSURE VESSEL DESIGN PROBLEM

As shown in Fig. 13, both ends of the cylindrical vessel are covered by hemispherical heads. The design goal of the pressure vessel is to minimize the total cost, including material, forming and welding costs. This problem involves four variables: shell thickness $T_s(x_1)$, head thickness $T_h(x_2)$, inner radius $R(x_3)$ and vessel length $L(x_4)$, where T_s and T_h

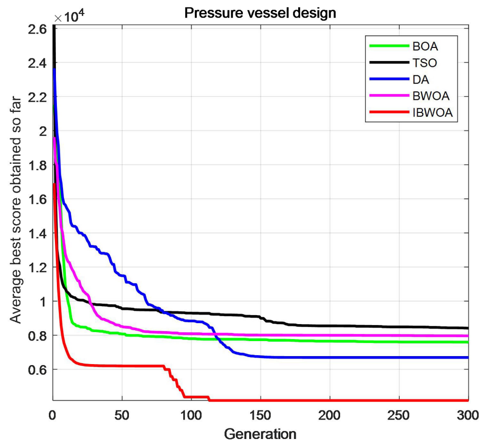

FIGURE 14. Average fitness curve of different algorithms for solving Pressure Vessel problem.

are integral multiples of 0.625 inch. R and L are continuous variables. The mathematical expression is as follows:

Objective function:

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{31}$$

Subject to:

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

Boundary condition:

$$\begin{aligned} x_1, x_2 &\in \{1 \times 0.0625, 2 \times 0.0625, 3 \times 0.0625, \cdots, 1600 \times 0.0625\}, \\ 10 &\le x_3, x_4 \le 200. \end{aligned}$$

The best solutions for pressure vessel design problem obtained by IBWOA and other methods are listed in Table 14. It can be seen that the optimal solution obtained by IBWOA is the best. According to the statistical results of IBWOA and other methods listed in Table 15, the IBWOA outperforms other methods in respect of the worst value, optimal solution, mean value and standard deviation. Therefore, the IBWOA possesses better convergence accuracy and stability when solving pressure vessel design problem. From the fitness curves of different algorithms to solve the pressure vessel problem given in Fig. 14, it can be seen that the convergence speed of IBWOA outranks other algorithms. Convergence speed, convergence accuracy and stability of IBWOA are the best in solving pressure vessel problem. Therefore, the IBWOA is a good choice in this problem.

TABLE 11. Statistical results of speed reducer design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	3851.937483	3469.363354	3105.562866	157.2338913
TSO	3425.333068	3167.781832	3005.501037	143.772895
DA	3056.000083	3018.242744	2996.714999	16.37631096
BWOA	3351.272101	3060.424161	3002.064865	87.58861506
IBWOA	**2994.424466**	**2994.424466**	**2994.424466**	**1.203E–08**

TABLE 12. Optimal results of piston lever design problem.

Algorithms/Variables	BOA	TSO	DA	BWOA	IBWOA
x_1	0.05	2.171306691	0.05	0.050152804	0.05
x_2	4.424024386	2.171306691	2.041513591	2.045658051	2.041513591
x_3	5.858859886	4.025863663	4.08302718	4.085367585	4.083027181
x_4	61.99517995	118.7718329	120	120	120
$g_1(x)$	–215492.9269	–7437.505565	–2.38E–05	–2010.901448	–0.001170462
$g_2(x)$	–19951.79947	–587718.3286	–600000	–600000	–600000
$g_3(x)$	–55.87198057	–113.9918059	–117.1874832	–117.1817068	–117.1874832
$g_4(x)$	–1.494594442	–0.15837486	–1.50E–09	–0.002974259	–1.16E–11
f(x)	**38.17495346**	**27.88731215**	**8.412698323**	**8.440173227**	**8.412698323**

TABLE 13. Statistical results of piston lever design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	6085.835949	840.0336453	38.17495346	1426.889975
TSO	2293.167311	450.3447518	27.88731215	423.5820243
DA	519.9923802	211.1498409	8.412698323	163.9011095
BWOA	4443.883217	504.2657334	8.440173227	815.5799698
IBWOA	**167.4727301**	**45.52670573**	**8.412698323**	**68.42493231**

TABLE 14. Optimal results of pressure vessel design problem.

Algorithms/Variables	BOA	TSO	DA	BWOA	IBWOA
x_1	15.45198215	16.21679631	11.93629887	12.78890915	12.67154314
x_2	8.445236392	7.626117159	6.205485988	6.772412192	6.954937506
x_3	48.56913216	51.62218131	40.31961872	42.09844518	42.0984456
x_4	113.955202	86.0158396	200	176.6366041	176.6365958
$g_1(x)$	–0.000115749	–0.003691901	0.028168641	–7.99E–09	4.93E–13
$g_2(x)$	–0.036650479	–0.00752439	0.009649163	–0.035880833	–0.035880829
$g_3(x)$	–28430.71553	–345.9818905	0	–0.017598918	–9.09E–07
$g_4(x)$	–126.044798	–153.9841604	–40	–63.36339587	–63.36340416
$f(x)$	6490.762405	6429.361814	6069.587335	6059.714459	**6059.714335**

TABLE 15. Statistical results of pressure vessel design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	8103.416923	7577.031993	6490.762405	386.9380589
TSO	12904.98906	8413.58021	6429.361814	1748.873686
DA	8213.282571	6694.534226	6069.587335	732.5516694
BWOA	17125.97271	7962.585819	6059.714459	2525.918052
IBWOA	**7046.575544**	**6191.614417**	**6059.714335**	**221.6468722**

7) TENSION/COMPRESSION SPRING DESIGN PROBLEM

As shown in Fig. 15, the goal of the design problem of the tension / compression spring is to minimize the weight of the tension / compression spring. The problem is constrained by the minimum deflection, shear stress, oscillation frequency and outer diameter. The problem includes three variables: the average diameter of the spring coil $D(x_1)$, the diameter of the spring wire $d(x_2)$, and the number of effective coils of the spring $N(x_3)$. The mathematical expression is as follows:

Objective function:

$$f(X) = (x_3 + 2)x_2x_1^2 \tag{32}$$

TABLE 16. Optimal results of tension/compression spring design problem.

Algorithms/Variables	BOA	TSO	DA	BWOA	IBWOA
x_1	0.055553661	0.052260082	0.051131126	0.051832804	0.051682254
x_2	0.456985345	0.370303919	0.343443024	0.360185213	0.356553986
x_3	7.204379597	10.59842857	12.11208033	11.08855739	11.29857501
$g_1(x)$	−0.005585398	−0.005083944	−1.78E−05	−3.61E−07	−1.31E−07
$g_2(x)$	−5.18E−05	−0.000671227	−1.90E−11	−1.05E−06	−5.90E−08
$g_3(x)$	−4.186009353	−4.050492608	−4.026653595	−4.060572404	−4.053461113
$g_4(x)$	−0.658307329	−0.718290665	−0.736950566	−0.725321322	−0.727842506
$f(x)$	0.012981419	0.012741333	0.012671162	0.012665638	**0.012665233**

TABLE 17. Statistical results of tension/compression spring design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	0.630756022	0.058422961	0.012981419	0.13185472
TSO	0.017180513	0.013719262	0.012741333	0.001043325
DA	0.030454968	0.015886904	0.012671162	0.005289225
BWOA	2.793542697	0.434937516	0.012665638	0.909564557
IBWOA	**0.012680259**	**0.012666253**	**0.012665233**	**2.83725E−06**

TABLE 18. Optimal results of welded beam design problem.

Algorithms/Variables	BOA	TSO	DA	BWOA	IBWOA
x_1	0.155213988	0.213878765	0.19888767	0.197922708	0.205729641
x_2	5.646153611	3.480069296	3.473015449	3.608876087	3.470488668
x_3	8.962109634	8.580987664	9.430677152	9.130576126	9.036623874
x_4	0.211358659	0.237730844	0.203843398	0.206022072	0.205729642
$g_1(x)$	−1041.390249	−0.613151717	−0.799252112	−0.223000019	−1.50E−05
$g_2(x)$	−311.3816541	−1208.091761	−2199.786561	−655.9251095	−1.75E−05
$g_3(x)$	−0.056144671	−0.023852079	−0.004955729	−0.008099364	−9.97E−10
$g_4(x)$	−3.183014371	−3.23392416	−3.340296582	−3.363126203	−3.390659093
$g_5(x)$	−0.030213988	−0.088878765	−0.07388767	−0.072922708	−0.080729641
$g_6(x)$	−0.23557143	−0.235385743	−0.237160479	−0.236002006	−0.235540323
$g_7(x)$	−470.6780533	−2943.040047	0	−66.58703627	−0.000139587
$f(x)$	1.940638318	1.891403646	1.767769855	1.749773548	**1.724852309**

TABLE 19. Statistical results of welded beam design problem.

Algorithms	Worst Value	Mean Value	Best Value	Standard Deviation
BOA	3.609391981	2.668938516	1.940638318	0.393862902
TSO	4.239627995	2.830744533	1.891403646	0.728730468
DA	2.460692476	2.059960384	1.767769855	0.254225694
BWOA	5.318498276	2.469834547	1.749773548	0.734001668
IBWOA	**1.724852309**	**1.724852309**	**1.724852309**	**6.77522E−16**

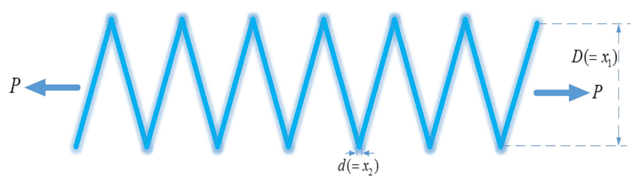

FIGURE 15. Schematic diagram of tension/compression spring.

Subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

Boundary condition:

$$0.05 \leq x_1 \leq 2,$$
$$0.25 \leq x_2 \leq 1.3,$$
$$2 \leq x_3 \leq 15.$$

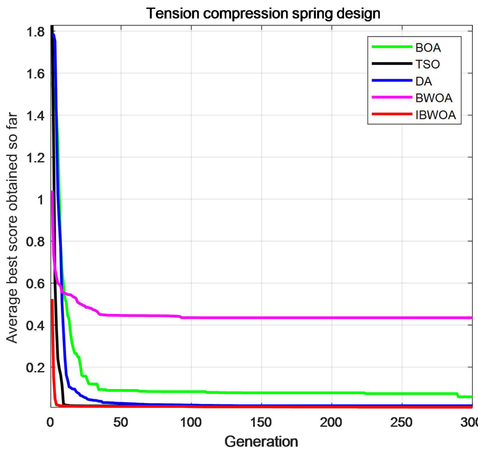

FIGURE 16. Average fitness curve of different algorithms for solving tension/compression spring problem.

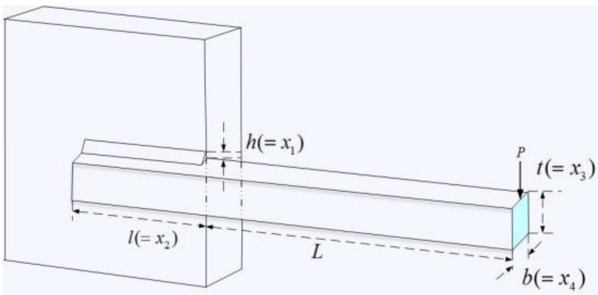

FIGURE 17. Schematic diagram of welded beam.

The best solutions for tension/compression spring design problem this problem obtained by IBWOA and other methods are listed in Table 16. It can be seen that the optimal solution obtained by IBWOA outranks other methods. According to the statistical results of IBWOA and other methods listed in Table 17, the IBWOA outperforms other methods in respect of the worst value, optimal solution, mean value and standard deviation. In other words, convergence speed and stability of IBWOA outrank other four algorithms. The indices of worst value, mean value and standard deviation obtained by TSO outperform the BOA, BWOA and DA in the case of similar optimization results. Therefore, compared with the other three algorithms, TSO gets competitive results in solving tension/compression spring design problem. From the fitness curves of different algorithms to solve the Tension/compression spring problem given in Fig. 16, it can be seen that the convergence speed of IBWOA outranks that of other algorithms. Therefore, it is further verified that the convergence speed, stability and convergence accuracy of the IBWOA are the best in solving tension/compression spring design problem.

8) WELDED BEAM DESIGN PROBLEM

As shown in Fig. 17, the goal of the beam bears vertical force problem is to find the minimum manufacturing cost of welded beams. The problem is constrained by seven constraints including shear stress, bending stress, beam bending load and end deviation. This problem involves four design variables, i.e. beam weld thickness $h(x_1)$, height $l(x_2)$, length $t(x_3)$ and thickness $b(x_4)$. The mathematical expression is as follows:

Objective function:

$$f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{33}$$

Subject to:

$$\begin{aligned}
&g_1(X) = \tau(X) - \tau_{\max} \le 0,\\
&g_2(X) = \sigma(X) - \sigma_{\max} \le 0,\\
&g_3(X) = \delta(X) - \delta_{\max} \le 0,\\
&g_4(X) = x_1 - x_4 \le 0,\\
&g_5(X) = P - P_c(X) \le 0,\\
&g_6(X) = 0.125 - x_1 \le 0,\\
&g_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0,\\
&\tau' = \frac{P}{\sqrt{2}\,x_1x_2},\\
&\tau'' = \frac{MR}{J},\\
&M = P(L + \frac{x_2}{2}),\\
&R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2},\\
&J = 2\left\{\sqrt{2}\,x_1x_2[\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2]\right\},\\
&\sigma(\vec{X}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{X}) = \frac{6PL^3}{Ex_3^2x_4},\\
&P_c(\vec{X}) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}),\\
&P = 6000lb, L = 14in, \delta_{\max} = 0.25in,\\
&E = 30 \times 10^6psi, G = 12 \times 10^6psi,\\
&\tau_{\max} = 13600psi, \sigma_{\max} = 30000psi,
\end{aligned}$$

Boundary condition:

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

The best solutions for welded beam design problem obtained by IBWOA and other methods are listed in Table 18. It can be seen that the optimal solution obtained by IBWOA outranks other methods. According to the statistical results of IBWOA and other methods listed in Table 19, the IBWOA outperforms other methods in the worst value, optimal solution, mean value and standard variance. Therefore, convergence accuracy and stability of IBWOA are the best when solving welded beam design problem. In addition, the BWOA outranks DA, TSO and BOA in terms of the optimization result. However, in the case of similar optimization

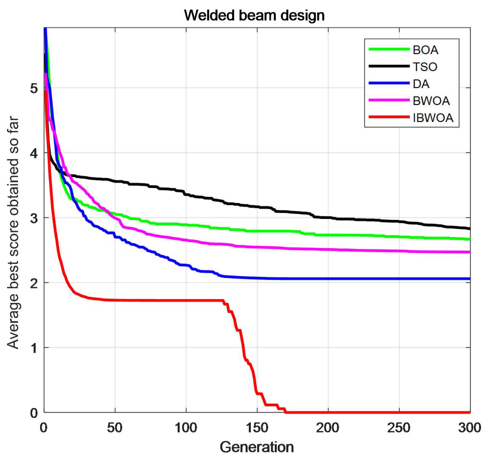

FIGURE 18. **Average fitness curve of different algorithms for solving Welded beam problem.**

results, the worst value, mean value and standard deviation obtained by DA outrank BWOA, TSO and BOA. Therefore, DA achieves competitive results in solving welded beam design problem. From the fitness curves of different algorithms to solve the welded beam problem given in Fig. 18, it can be seen that the convergence speed of IBWOA outranks that of other algorithms. It is proved that IBWOA is a powerful tool for solving welded beam design problem.

VII. CONCLUSION

In this paper, for solving engineering constrained optimization problems, an improved black widow optimization algorithm (IBWOA) is proposed with new hybrid strategies being employed in the IBWOA. Firstly, the quality of the initial population of the BWOA is improved based on the double chaotic map; secondly, in order to make full use of the difference information between the current and the optimal position, golden sine guidance strategy is introduced to improve optimization accuracy of the IBWOA; finally, the Cauchy barycenter reverse differential mutation operator is employed to increase the diversity of the population, avoid trucking local optimization and enhance the global search ability of the IBWOA.

Eight continuous / discrete hybrid engineering optimization problems and typical benchmark functions are utilized to evaluate the performance of the IBWOA. The optimization results show that IBWOA effectively improves the search accuracy, convergence speed and robustness of engineering constrained optimization problems, and overcomes the defect that IBWOA is easy to fall into local optimization. The global exploration, local exploitation and optimization ability of the IBWOA with new hybrid strategies can be improved. The new hybrid strategies can enable the algorithm to seek an appropriate balance between the exploration and exploitation stages. Moreover, it is proved that the IBWOA which retains the optimal solution after Cauchy barycentric reverse difference mutation strategy converges to the global optimal solution with a probability of 1 based on Markov theory. It opens up a future for online optimization calculation in a low grade hardware condition.

However, the convergence speed and accuracy of the algorithm are only analyzed based on experiments. Markov theory only proves the convergence performance of the algorithm in the sense of probability. There are some uncertainties in actual performance. The IBWOA is a meta- heuristic algorithm with random characteristics. It will still be a challenging task to achieve an appropriate balance between the exploration and exploitation stages in the future.

This paper extends the application of BWOA in engineering constrained optimization problems. The IBWOA will be evaluated on machine learning, pattern recognition and other complex real-world optimization problems in our future work.

REFERENCES

[1] L. Wang, *Intelligent Optimization Algorithms With Application*. Beijing, China: Tsinghua Univ. Press, 2001.

[2] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization*. Cambridge, MA, USA: MIT Press, 2019.

[3] S. S. Rao, *Engineering Optimization*. Hoboken, NJ, USA: Wiley, 2009.

[4] M. Azizi, S. Talatahari, and A. Giaralis, "Optimization of engineering design problems using atomic orbital search algorithm," *IEEE Access*, vol. 9, pp. 102497–102519, 2021.

[5] N. Chopra and M. M. Ansari, "Golden jackal optimization: A novel nature-inspired optimizer for engineering applications," *Exp. Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116924.

[6] Y. Yuan, J. Ren, S. Wang, Z. Wang, X. Mu, and W. Zhao, "Alpine skiing optimization: A new bio-inspired optimization algorithm," *Adv. Eng. Softw.*, vol. 170, Aug. 2022, Art. no. 103158.

[7] S.-P. Gong, M. Khishe, and M. Mohammadi, "Niching chimp optimization for constraint multimodal engineering optimization problems," *Exp. Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116887.

[8] S. Minocha and B. Singh, "A novel equilibrium optimizer based on Lévy flight and iterative cosine operator for engineering optimization problems," *Exp. Syst.*, vol. 39, no. 2, pp. 1–49, Oct. 2021.

[9] D. Dhawale, V. K. Kamboj, and P. Anand, "An improved chaotic Harris hawks optimizer for solving numerical and engineering optimization problems," *Eng. Comput.*, pp. 1–46, Sep. 2021, doi: 10.1007/s00366-021-01487-4.

[10] J. Ren, H. Wei, Y. Yuan, X. Li, F. Luo, and Z. Wu, "Boosting sparrow search algorithm for multi-strategy-assist engineering optimization problems," *AIP Adv.*, vol. 12, no. 9, Sep. 2022, Art. no. 095201.

[11] Q. He, L. Wang, and B. Liu, "Parameter estimation for chaotic systems by particle swarm optimization," *Chaos, Solitons Fractals*, vol. 34, no. 2, pp. 654–661, Oct. 2007.

[12] C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Comput. Oper. Res.*, vol. 22, no. 1, pp. 5–13, Jan. 1995.

[13] L. Poli, P. Rocca, L. Manica, and A. Massa, "Handling sideband radiations in time-modulated arrays through particle swarm optimization," *IEEE Trans. Antennas Propag.*, vol. 58, no. 4, pp. 1408–1411, Jan. 2010.

[14] A. A. Ayman, "PID parameters optimization using genetic algorithm technique for electrohydraulic servo control system," *Intell. Control Autom.*, vol. 2, no. 2, pp. 69–76, 2011.

[15] W. Zhao and L. Wang, "An effective bacterial foraging optimizer for global optimization," *Inf. Sci.*, vol. 329, pp. 719–735, Feb. 2016.

[16] G. Kourakos and A. Mantoglou, "Development of a multi-objective optimization algorithm using surrogate models for coastal aquifer management," *J. Hydrol.*, vol. 479, pp. 13–23, Feb. 2013.

[17] M.-C. Yuen, S.-C. Ng, and M.-F. Leung, "A competitive mechanism multi-objective particle swarm optimization algorithm and its application to signalized traffic problem," *Cybern. Syst.*, vol. 52, no. 1, pp. 73–104, Oct. 2020.

[18] Y. Yuan, X. Mu, X. Shao, J. Ren, Y. Zhao, and Z. Wang, "Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic k-best gravitational search strategy based grey wolf optimizer algorithm," *Appl. Soft Comput.*, vol. 123, Jul. 2022, Art. no. 108947.

[19] M.-C. Yuen, S.-C. Ng, M.-F. Leung, and H. Che, "A metaheuristic-based framework for index tracking with practical constraints," *Complex Intell. Syst.*, vol. 8, no. 6, pp. 4571–4586, Dec. 2022.

[20] M. Frank, D. Drikakis, and V. Charissis, "Machine-learning methods for computational science and engineering," *Computation*, vol. 8, no. 1, pp. 1–35, Mar. 2020.

[21] S. Arunkumar and T. Chockalingam, "Genetic search algorithms and their randomized operators," *Comput. Math. Appl.*, vol. 25, no. 5, pp. 91–100, Mar. 1993.

[22] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, 1995, pp. 1942–1948.

[24] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[25] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[26] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 169–196, Feb. 2019.

[27] T. Wang, L. Yang, and Q. Liu, "Beetle swarm optimization algorithm: Theory and application," *Filomat*, vol. 34, no. 15, pp. 5121–5137, 2020.

[28] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.

[29] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bioinspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intel.*, vol. 90, pp. 1–29, Mar. 2020.

[30] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[31] S. Mirjalili, "The Ant Lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.

[32] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[33] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016.

[34] J. Pierezan and L. D. S. Coelho, "Coyote optimization algorithm: A new metaheuristic for global optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Oct. 2018, pp. 1–8.

[35] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, vol. 284. Berlin, Germany: Springer, 2010, pp. 65–74.

[36] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Comput. Eng. Dept., Eng. Fac., Erciyes Univ., Ayseri, Turkey, Tech. Rep. TR-06, Jan. 2005.

[37] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," *Eng. Optim.*, vol. 38, no. 2, pp. 129–154, 2006.

[38] S. J. Liu, Y. Yang, and Y. Q. Zhou, "A swarm intelligence algorithm-lion swarm optimization," *Pattern Recognit. Artif. Intell.*, vol. 31, no. 5, pp. 431–441, May 2018.

[39] S. A. Uymaz, G. Tezel, and E. Yel, "Artificial algae algorithm (AAA) for nonlinear global optimization," *Appl. Soft Comput.*, vol. 31, pp. 153–171, Jun. 2015.

[40] X. S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.*, vol. 7445, no. 2, pp. 240–249, Dec. 2012.

[41] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[42] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[43] R. A. Formato, "Central force optimization: A new gradient-like metaheuristic for multidimensional search and optimisation," *Int. J. Bio-Inspir. Com.*, vol. 1, no. 4, pp. 217–238, Apr. 2009.

[44] I. Birbil and S.-C. Fang, "An electromagnetism-like mechanism for global optimization," *J. Global Optim.*, vol. 25, no. 3, pp. 263–282, 2003.

[45] L. P. Xie, J. C. Zeng, and Z. H. Cui, "On mass effects to artificial physics optimization algorithm for global optimization problems," *Int. J. Innov. Comput. Appl.*, vol. 2, no. 2, pp. 69–76, Feb. 2009.

[46] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Transient search optimization: A new meta-heuristic optimization algorithm," *Int. J. Speech Technol.*, vol. 50, no. 11, pp. 3926–3941, Nov. 2020.

[47] W. Zhao, L. Wang, and Z. Zhang, "A novel atom search optimization for dispersion coefficient estimation in groundwater," *Future Gener. Comput. Syst.*, vol. 91, pp. 601–610, Feb. 2019.

[48] X. Jiang and S. Li, "BAS: Beetle antennae search algorithm for optimization problems," *Int. J. Robot. Control*, vol. 1, no. 1, pp. 1–5, Apr. 2018.

[49] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation," *Soft Comput.*, vol. 13, no. 2, pp. 157–168, Jan. 2009.

[50] S.-K.-S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *Eur. J. Oper. Res.*, vol. 181, no. 2, pp. 527–548, Sep. 2007.

[51] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.

[52] A. E. Hassanien, R. M. Rizk-Allah, and M. Elhoseny, "A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems," *J. Ambient Intell. Hum. Comput.*, pp. 1–25, Jun. 2018, doi: 10.1007/s12652-018-0924-y.

[53] Q. Liu, Y. H. Feng, and Y. Y. Chen, "Moth-flame optimization algorithm based on chaotic initialization and Gaussian mutation," *J. Zhengzhou Univ. Eng. Sci.*, vol. 42, no. 3, pp. 53–58, May 2021.

[54] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSOGSA algorithm for function optimization," in *Proc. Int. Conf. Comput. Inf. Appl.*, Tianjin, China, Dec. 2010, pp. 374–377.

[55] O. Abdel-Raouf, I. El-Henawy, and M. Abdel-Baset, "A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles," *Int. J. Mod. Educ. Comput. Sci.*, vol. 6, no. 3, pp. 38–44, Mar. 2014.

[56] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.

[57] L. Wang and B. Liu, *Particle Swarm Optimization and Scheduling Algorithms*. Beijing, China: Tsinghua Univ. Press, 2008.

[58] L. J. Sun, B. B. Feng, and T. F. Chen, "Global convergence analysis of grey wolf optimization algorithm based on martingale theory," *Control Decis.*, vol. 37, no. 11, pp. 2839–2848, Nov. 2022.

[59] X. S. Xu, S. J. Yang, R. Y. Chen, W. Liang, and W. J. Jiang, "The convergence and performmance analyses of immune evolutionary algorithm based on stochastic functional theory," *Control Decis.*, vol. 33, no. 6, pp. 1100–1106, Jun. 2018.

[60] Z. H. Ren, J. Wang, and Y. L. Gao, "The global convergence analysis of particle swarm optimization algorithm based on Markov chain," *Control Theory Appl.*, vol. 28, no. 4, pp. 462–466, Apr. 2011.

[61] A. P. Ning and X. Y. Zhang, "Convergence analysis of artificial bee colony algorithm," *Control Decis.*, vol. 28, no. 10, pp. 1554–1558, Oct. 2013.

[62] D. H. Wu, F. Kong, and Z. C. Ji, "Convergence analysis of chicken swarm optimization algorithm," *J. Cent. South. Univ. Sci. Tech.*, vol. 48, no. 8, pp. 2105–2112, Aug. 2017.

[63] J. N. Shang, T. Cheng, K. Q. Yue, and L. Sheng, "Markov chain model analysis of bat algorithm," *Comput. Eng.*, vol. 43, no. 7, pp. 198–202, Jul. 2017.

[64] B. L. Li, X. L. Shi, C. X. Gou, D. J. Lv, Z. Z. An, and Y. F. Zhang, "Multivariant optimization algorithm and its convergence analysis," *Acta Autom. Sin.*, vol. 41, no. 5, pp. 949–959, May 2015.

[65] A. F. Peña-Delgado, H. Peraza-Vázquez, J. H. Almazán-Covarrubias, N. T. Cruz, P. M. García-Vite, A. B. Morales-Cepeda, and J. M. Ramirez-Arredondo, "A novel bio-inspired algorithm applied to selective harmonic elimination in a three-phase eleven-level inverter," *Math. Problems Eng.*, vol. 2020, pp. 1–10, Dec. 2020.

[66] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[67] S. Duan, H. Luo, and H. Liu, "A multi-strategy seeker optimization algorithm for optimization constrained engineering problems," *IEEE Access*, vol. 10, pp. 7165–7195, 2022.

[68] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst., Man Cybern., C, Appl. Rev.*, vol. 35, no. 2, pp. 233–243, May 2005.
[69] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 629–640, Mar. 2010.
[70] Y. Wang, Z.-X. Cai, Y.-R. Zhou, and C.-X. Xiao, "Constrained optimization evolutionary algorithms," *J. Softw.*, vol. 20, no. 1, pp. 11–29, Apr. 2009.
[71] C. Lin, "An adaptive genetic algorithm based on population diversity strategy," in *Proc. 3rd Int. Conf. Genetic Evol. Comput.*, Guilin, China, Oct. 2009, pp. 93–96.
[72] J. Luo, J. Zhou, and X. Jiang, "A modification of the imperialist competitive algorithm with hybrid methods for constrained optimization problems," *IEEE Access*, vol. 9, pp. 161745–161760, Dec. 2021.
[73] Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu, and C. Li, "Chaos enhanced bacterial foraging optimization for global optimization," *IEEE Access*, vol. 6, pp. 64905–64919, Oct. 2018.
[74] C. G. Fei and Z. Z. Han, "A novel chaotic optimization algorithm and its applications," *J. Harbin Inst. Technol.*, vol. 17, no. 2, pp. 254–258, Apr. 2010.
[75] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field," *IEEE Access*, vol. 8, pp. 186638–186652, Oct. 2020.
[76] X. T. Cui, Y. Li, and J. H. Fan, "Global chaotic bat optimization algorithm," *J. Northeastern Univ. Nat. Sci.*, vol. 41, no. 4, pp. 488–491+498, Apr. 2020.
[77] J. Y. Liu, J. K. Ge, and J. T. Tang, "A fast chaotic image encryption algorithm based on improved sine map," *J. Chongqing Univ. Sci. Technol. Nat. Sci.*, vol. 22, no. 5, pp. 75–80+90, Oct. 2020.
[78] E. Tanyildizi and G. Demir, "Golden sine algorithm: A novel math-inspired algorithm," *Adv. Electr. Comput. Eng.*, vol. 17, no. 2, pp. 71–78, 2017.
[79] J. Li, J. Zou, B. Li, and J. Q. Liu, "A different evolution algorithm using multi-neighborhood strategy and neighborhood centroid opposition-based learning," *J. Wuhan Univ. Sci. Technol.*, vol. 41, no. 3, pp. 232–240, May 2018.
[80] Y. C. Chi, J. Fang, and G. B. Cai, "Center mutation based differential evolution," *Syst. Eng. Electron.*, vol. 32, no. 5, pp. 1105–1108, May 2010.
[81] P. Y. Fan, *Stochastic Process Theory and Application*. Beijing, China: Tsinghua Univ. Press, 2005.
[82] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl-Based Syst.*, vol. 96, no. 3, pp. 120–133, Jan. 2016.
[83] H. Bayzidi, S. Talatahari, M. Saraee, and C.-P. Lamarche, "Social network search for solving engineering optimization problems," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–32, Sep. 2021.
[84] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2019.

DONGXING XU received the B.S. degree from Guangdong Ocean University, China, in 2013, and the M.S. degree from Dalian Maritime University, China, in 2016. He is currently a Lecturer with Naval Architecture and Shipping College, Guangdong Ocean University. His research interests include marine intelligent control and intelligent optimization algorithms.

JIANCHUAN YIN received the B.E. degree in marine navigation technique, the M.E. degree in vehicle operation engineering, and the Ph.D. degree in traffic transportation engineering from Dalian Maritime University, China, in 1995, 2001, and 2007, respectively. From December 2010 to March 2013, he was a Postdoctoral Researcher with the School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, China. From October 2016 to October 2017, he was a Visiting Scholar with the Department of Naval Architecture and Marine Engineering, University of Michigan, USA. He is currently a Professor with Naval Architecture and Shipping College, Guangdong Ocean University, China. His research interests include machine learning, intelligent marine navigation, and marine autonomous systems.