Improved Honey Badger Algorithm Based on a Hybrid Strategy

Jiayao Wen, Yu Liu*, Yutong Li, Zhen Wang, Pengguo Yan and Tiefeng An

Abstract—**The Honey Badger Algorithm (HBA) represents a novel swarm intelligence optimization algorithm introduced in recent years. However, its predominant constraints are linked to inadequate convergence accuracy and a vulnerability to entrapment in local optima. In an effort to mitigate these challenges, this paper introduces an Improved Honey Badger Algorithm Based on a Hybrid Strategy (OHBA). Firstly, during the population initialization phase, a method involving the utilization of a good point set is introduced to enhance the diversity and introduce more randomness into the population. Secondly, in the position update phase, the Beta distribution is employed as an alternative to the Uniform distribution, aiming to strike a balance between global exploration and local exploitation capabilities. Thirdly, an improved adaptive density factor strategy is incorporated into both global and local position updates to enhance the algorithm's convergence precision and speed. Lastly, within the global exploration stage, a Cauchy mutation strategy based on the Sine chaotic mapping is introduced to facilitate the algorithm in overcoming local optima and reinforcing its optimization capabilities. The improved algorithm's performance has been evaluated through a comprehensive set of assessments, including CEC-2017 functions, CEC-2022 functions, Wilcoxon rank-sum tests, and practical engineering optimization problems. These evaluations were undertaken to assess the algorithm in comparison to classical intelligent optimization algorithms. The experimental results show that OHBA possesses significant advantages in terms of convergence speed, optimization accuracy, robustness and its practical utility and effectiveness in addressing complex optimization challenges. This establishes OHBA as a highly competitive option in these critical aspects of optimization.**

Index Terms—**Honey Badger Algorithm, Good point set, Adaptive Density Factor, Beta distribution, Cauchy Mutati on**

Manuscript received September 29, 2023; revised December 12, 2023.

Jia-Yao Wen is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: m17641243544@163.com).

Yu Liu is a supervisor of postgraduate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan,114051, P. R. China (Corresponding author, phone: 86-0412-5929747; fax:86-0412-5929747; lnasacl@126.com)

Yu-Tong Li is a postgraduate student at School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 13942871038@163.com).

Zhen Wang is a Junior Engineer of Shenyang Siwo Electric Appliance Co., Ltd, Liaoning, Shenyang, 110036, P. R. China (e-mail: 17641243669@163.com).

Peng-Guo Yan is an undergraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: yan407319226@163.com).

Tiefeng An is a Senior Engineer of Steelworks of Beiying Iron and Steel Group Co., Ltd of Benxi Steel Group Corporation, Liaoning, Benxi, 117000, P. R. China (phone: 86-13842446121; e-mail: tiefengan@126.com).

I. Introduction

The metaheuristic optimization algorithms have attracted widespread attention due to their simple principles, ease of implementation, and superior performance in the past few decades [1][2]. These algorithms have demonstrated substantial promise, primarily aimed at expeditiously resolving a variety of complex optimization problems and achieving more robust optimization outcomes [3][4]. The foundation of metaheuristic algorithms lies in diverse biological mechanisms and physical laws. Among metaheuristic algorithms, swarm intelligence algorithms play a significant role, as they attain optimal solutions through iterative adjustments within a population. Prominent swarm intelligence algorithms include the ant colony algorithm, which mimics the foraging patterns of ant colonies [5]; the particle swarm algorithm, inspired by the foraging behaviors of bird flocks [6]; the pelican optimization algorithm, which takes inspiration from the natural hunting behavior of pelicans [7]; and the grey wolf optimization algorithm, which is inspired by the hunting patterns of grey wolf packs [8]. In general, real-world optimization problems often exhibit several challenging characteristics, including nonlinearity, complexity, extensive computational demands, and vast search spaces [9]-[11]. Swarm intelligence algorithms have demonstrated their remarkable effectiveness in delivering superior solutions to a broad spectrum of intricate optimization problems, in contrast to conventional numerical methods. As a result, they find widespread application in tackling complex real-world engineering challenges, such as path planning [12][13], path tracking [14][15], combinatorial optimization [16][17], and feature selection [18][19], consistently yielding favorable results.

The Honey Badger Algorithm (HBA) [20], introduced by Fatma A. and her colleagues in 2021, stands as a novel swarm intelligence algorithm. Its fundamental principle revolves around modeling mathematical optimization problems by simulating the foraging behavior of honey badgers. The standard HBA algorithm demonstrates several notable strengths, including robust optimization capabilities, minimal tuning parameters, and high stability. In its initial applications within engineering practice, it has shown promising outcomes. Timur Düzenli and his colleagues introduced an enhanced Honey Badger Algorithm for the extraction of photovoltaic model parameters [21]. The proposed algorithm exhibits high performance in minimizing root mean square error (RMSE) and offers an effective alternative for addressing the challenge of photovoltaic parameter estimation, thereby contributing to the optimization of photovoltaic model parameters. Lei W and their research team presented a study titled "Solar Photovoltaic Cell Parameter Identification Based on the Improved Honey Badger Algorithm" [22]. The IHBA algorithm, as proposed, is capable of rapidly calculating the minimum objective function RMSE for three different models. It accurately identifies the parameters that need to be solved within these models, presenting a valuable tool for expeditious parameter identification in the context of solar photovoltaic cells. Ma and colleagues introduced a groundbreaking approach for diagnosing compound faults in rolling bearings [23]. This method harnesses the power of optimized variational mode decomposition (VMD) and the extreme learning machine (ELM) within the framework of the chaotic honey badger algorithm (CHBA). Notably, this approach demonstrates exceptional diagnostic capabilities for compound faults, achieving an impressive fault classification accuracy rate of up to 100%. It represents an innovative solution to the challenges associated with compound fault diagnosis. Reference [24] introduced a feature selection methodology grounded in wrapper techniques, which seamlessly integrates the multi-objective honey badger algorithm (MO-HBA) and the strength Pareto evolution algorithm-II. This approach effectively alleviates redundancy within extensive datasets while simultaneously enhancing classification accuracy. Khan et al. [25] advocate for the application of the Honey Badger Algorithm to optimize the placement of Distributed Generation (DG) in IEEE 33-node and 69-node distribution test systems. This optimization aims to minimize power losses while simultaneously improving voltage distribution and enhancing system stability. In reference[26], Chen et al. employ the improved HBA algorithm to optimize the frequency control sliding mode design within a multi-area power system that incorporates disturbance observers. The experimental results demonstrate that this control method exhibits exceptional frequency control capabilities, especially in handling uncertain conditions such as load disturbances, wind power fluctuations, and parameter uncertainties.

When compared to many intelligent optimization algorithms, the Honey Badger Algorithm (HBA) offers several advantages in problem optimization. However, it is not without its challenges, including slow convergence and vulnerability to premature convergence towards local optima. Currently, researchers have employed various methods to tackle these issues. In Reference [27], a multi-strategy improved Honey Badger Optimization Algorithm is introduced. This enhanced algorithm showcases substantial enhancements in convergence speed, target precision, and optimization search capabilities, thus demonstrating robustness in addressing these challenges. Chen R.F. and their colleagues [28] introduced an enhanced Honey Badger Algorithm that incorporates elite reverse learning, spiral updates, and wild dog survival strategies to optimize the systematic charging of electric vehicles (EVs). This approach significantly improves convergence speed and precision. Maintaining good population diversity and striking a balance between exploration and exploitation are consistently crucial for metaheuristic algorithms [29]. References [30][31] introduce the utilization of chaotic mapping for population initialization, thereby increasing population diversity and introducing randomness to help populations escape local optima. Furthermore, References [32][33] introduce dynamic inertia weight strategies aimed at enhancing the algorithm's performance. This results in improved convergence speed and dynamic balancing of the trade-off between exploration and exploitation.

To tackle challenges such as slow convergence, vulnerability to local optima, and limited convergence precision in the Honey Badger Algorithm (HBA), this paper presents an Improved Honey Badger Algorithm (OHBA) Based on a Hybrid Strategy. The outcomes illustrate the applicability and effectiveness of this algorithm in optimization. The contributions of this study can be summarized as follows:

1.Using the method of the good point set to initialize the initial population, helping the population escape local optima.

2.Replacing the Uniform distribution of the original algorithm with a Beta distribution to balance the search and exploitation capabilities.

3.Improving the adaptive density factor strategy to accelerate the algorithm's convergence speed and enhance its exploration capability.

4.Introducing a Cauchy mutation based on the Sine chaotic mapping to increase perturbation, improve the update of honey badger individuals' positions, and enhance both global exploration and local exploration capabilities.

5.The efficiency and performance of the proposed OHBA algorithm are evaluated through CEC-2017, and CEC-2022 benchmark test functions.

6.To validate the performance of the OHBA algorithm, it is compared to the original HBA, DBO, WOA, MFO, SCA, and BOA algorithms.

7.Wilcoxon rank-sum tests have been conducted to confirm the statistical superiority of the proposed OHBA.

8.The solving ability of the proposed OHBA algorithm is evaluated by solving two real-world engineering design problems (pressure vessel design problem and rolling bearing design problem).

The remaining structure of this research is organized as follows: Section 2: Introduces the basic concepts of the Honey Badger Algorithm (HBA), including its mathematical model, fundamental principles, and pseudocode. Section 3: Provides in-depth explanations of the strategies employed in this research, including the good point set, generation of random numbers using the Beta distribution, adaptive density factor, and the Cauchy mutation strategy based on the Sine chaotic mapping. It also includes the flowchart of the Improved Honey Badger Algorithm (OHBA). Section 4: Rigorously assesses the efficacy of the proposed Optimal Honey Badger Algorithm (OHBA) by subjecting it to two sets of diverse benchmark test functions, complemented by Wilcoxon rank-sum tests. Section 5: Evaluates the effectiveness of the proposed algorithm in addressing two distinct real-world engineering design problems. Section 6: Summarizes the conclusions drawn from this research and outlines potential future directions.

II. Honey Badger Algorithm

The Honey Badger Algorithm is a swarm intelligence algorithm inspired by the foraging behavior of honey badgers. These resilient creatures utilize a combination of two distinct methods to successfully locate food sources:

1、The Digging Phase: In this method, honey badgers rely on their acute sense of smell to approximate the location of prey. Once they identify a potential spot, they choose an appropriate location to dig in search of prey.

2、The Honey Phase: Alternatively, honey badgers utilize the guidance of honey guide birds to approximate the location of prey. These birds lead them to potential food sources.

A. Initialization phase

The population size and positions of honey badgers are initialized using a Uniform distribution. The population initialization is defined by Eq. (1):

$$x_i = lb_i + r_1 \times (ub_i - lb_i) \tag{1}$$

Where x_i is ith honey badger position; r_1 is a random number within the range (0, 1); ub_i and lb_i are the upper and lower bounds of the search space, respectively.

B. Defining intensity (I):

The intensity mainly relies on the concentration strength of the prey and the distance between the prey and the honey badger. I_i is smell intensity of the prey. The higher the smell, the faster the honey badger's movement speed will be, and vice versa. It is defined by Eq. (2)-(4):

$$I_i = r_2 \times \frac{S}{4\pi d_i^2} \tag{2}$$

$$S = (x_i - x_{i+1})^2 \tag{3}$$

$$d_i = x_{prey} - x_i \tag{4}$$

Where S is source strength or concentration strength, and di is defined as the distance between the prey and the ith honey badger.

C. Update density factor

The density factor α governs time-varying randomization to ensure a seamless transition from exploration to exploitation. With the progression of iterations, the density factor α also decreases correspondingly to reduce the randomness introduced by temporal changes, using Eq. (5):

$$\alpha = C \times \exp\left(\frac{-t}{t_{max}}\right) \tag{5}$$

Where t_{max} is the maximum number of iterations.

D. Digging phase.

The first way badgers find food sources is the badger performs a motion resembling Cardioid shape. This is simulated using Eq. (6):

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times \alpha \times d_i \times |\cos(2\pi r_4) \times [1 - \cos(2\pi r_5)]| \tag{6}$$

Where x_{prey} is the current global optimal location for prey found so far, "β" is the badger's food acquisition capability, $\beta \geq 1$, (default = 6) d_i is the distance between the prey and the badger, and r_3, r_4, r_5 are random numbers within the range (0, 1) respectively. "F" is a flag used to determine the change in the search direction, as determined by Eq. (7):

$$F = \begin{cases} 1, & if \quad r_6 < 0.5 \\ -1, & else \end{cases} \tag{7}$$

E. Honey phase

The second way badgers find food sources is by following honey guide bird to locate beehive. This is simulated using Eq. (8):

$$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i \tag{8}$$

Where x_{new} represents the current position of the honey badger, whereas x_{prey} is the position of the prey. F and α are determined by Eq. (7) and Eq. (5) respectively. According to Eq. (8), the badger performs search close to prey location x_{prey} found so far, based on distance information d_i. The search process is also influenced by the density factor α and disturbance factor F.

F. Pseudo-code

To clearly show the structure of the Honey Badger Algorithm, the HBA pseudo-code are shown in Algorithm.

HBA pseudo-code

Input: The population size N, the number of iterations t_{max} the variable dimension Dim

Output: The optimal position x_{prey}, the best fitness value f_{prey}

```
Initialize the honey badger population X using Eq. (1)
Calculate the fitness value f of each honey badger
Select the optimum position x_prey according to fitness values
while (t < t_max) do
Update α according to Eq. (5)
    for i = 1 to N do
        Update I_i according to Eq. (2)
        if rand < 0.5 do
            Update the current position x_new using Eq. (6).
        else if do
            Update the current position x_new using Eq. (8).
        end if
        Evaluate new position and assign to f_new
        if f_new ≤ f_i
            x_i = x_new and f_i = f_new
        end if
        if f_new ≤ f_prey
            x_prey = x_new and f_prey = f_new
        end if
    end for
end while
Return x_prey
```

III. Improved Honey Badger Algorithm

To enhance the precision of solutions, accelerate convergence, and bolster the robustness of the Honey Badger Algorithm, this paper introduces four key improvements to the algorithm: Population Initialization with the Good Point Set Method, Utilization of the Beta Distribution for Random Number Generation, Introduction of an Adaptive Density Factor, Application of a Cauchy Mutation Strategy Based on the Sine Chaotic Mapping.

A. Good point set

The conventional Honey Badger Algorithm usually begins by randomly initializing the population and subsequently identifying the best individual from this initial set. It is crucial to acknowledge that the quality of this initial population has a substantial impact on the optimization efficiency of the algorithm in subsequent iterations. Ideally, the initial population should be uniformly distributed across the solution space, thereby aiding the algorithm in exploring the global space during its early stages. However, random initialization methods fail to guarantee uniformity and diversity within the initial population.

In this paper, we employ the good point set method for initializing the honey badger population. This approach is designed to generate an initial population that is distributed more evenly across the solution space, thereby improving the algorithm's capacity to explore the global solution space in its initial iterations.

The basic definition and construction of the good point set method to initialize the population are as follows: the population is N, select n points in an s-dimensional space as $P_n(k)=\left\{\left(\left\{r_1^{(n)}*k\right\},\left\{r_2^{(n)}*k\right\},\cdots\left\{r_s^{(n)}*k\right\}\right),1\le k\le n\right\}$,and its deviation $\varphi(n)$ satisfies $\varphi(n)=C(r,\varepsilon)n^{-1+\varepsilon}$, where $C(r,\varepsilon)$ is a constant only related to r and ε (where ε is any positive number), then $P_n(k)$ is called a good point set, and r is a good point. Take the fractional part of $\left\{r_s^{(n)}*k\right\}$. Take $r=\left\{2\cos\left(\frac{2\pi k}{p}\right),1\le k\le s\right\}$(p is the smallest prime number that satisfies $(p-3)/2\ge s$.

A one-dimensional initial population consisting of 500 points is generated using the good point set method and compared with a random selection method, as depicted in Fig.1 and Fig.2. The graph clearly illustrates that when an equal number of points are chosen, the good point set sequence displays a more even distribution compared to the random point selection approach. Notably, the good point set selection method exhibits a high degree of stability, consistently producing the same distribution effect regardless of the number of computations conducted, as long as the population size remains constant. Consequently, when mapping these good points onto the target solution space within the Improved Honey Badger Algorithm (OHBA), it results in a more evenly distributed initial population of honey badgers. This significantly enhances the algorithm's ability to explore a broader range of solutions, thereby offering an effective method for uniform point selection. This, in turn, contributes to achieving superior global optimization.

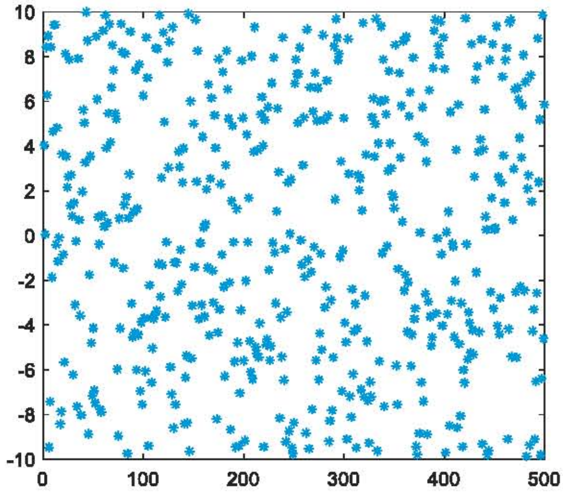


Fig.1. Randomly distributed n data points

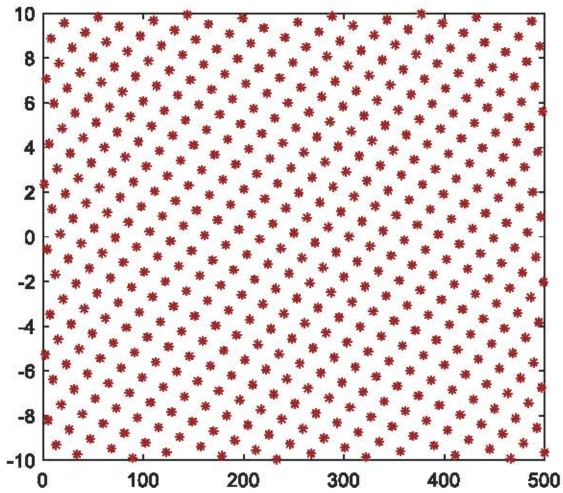


Fig.2. n data points distributed in good point set

B. Adaptive density factor

In the standard Honey Badger Algorithm, as the number of iterations increases, the honey badgers' exploration range becomes increasingly limited, hindering them from fully exploring optimal values. This reduction in population diversity can result in the algorithm becoming trapped in local optima, leading to diminished search accuracy. The density factor plays a critical role as a tuning parameter in optimizing the Honey Badger Algorithm. During the initial iterations, a strong global exploration capability enables it to explore a broad solution space, maintain population diversity, and steer clear of local optima. Conversely, in later iterations, the algorithm should exhibit robust local exploitation capabilities. A stronger local search capability contributes to higher precision in local optimization and faster convergence.

To strike the right balance between these two aspects, an adaptive density factor is introduced, which aids in harmonizing the algorithm's global exploration and local exploitation capabilities. The improved density factor formula is represented as Eq. (9):

$$w=a+b\times\sin(\pi+\frac{\pi\cdot t}{2\cdot t_{max}}) \tag{9}$$

TABLE I
TEST FUNCTION EXPERIMENT RESULTS

Parameter	0.5	1	1.5	2	2.5	3
1	2.7555E+03	2.7706E+03	2.8102E+03	**2.7276E+03**	2.8090E+03	2.7869E+03
1.5	2.7122E+03	2.7343E+03	2.7134E+03	**2.7006E+03**	2.7759E+03	2.7723E+03
2	2.7261E+03	2.7606E+03	2.7223E+03	**2.6334E+03**	2.7638E+03	2.7091E+03
2.5	2.7034E+03	**2.6500E+03**	2.6828E+03	2.6889E+03	2.6771E+03	2.6615E+03

In the equation, t represents the current iteration number, and t_{max} is the maximum iteration number. The parameters 'a' and 'b' are influential factors, and their careful selection is crucial as they have a significant impact on the performance of OHBA. To assess the influence of these parameters, we conducted a sensitivity analysis using Composition Function 3 (N=5). For this analysis, we created six scenarios by combining different values of 'a' and 'b'. From Table I and Fig.3, it becomes evident that the best fitness value is achieved when 'a' is set to 2 and 'b' is set to 2. Hence, the optimal parameter values for this algorithm are 'a=2' and 'b=2'. As a result, the position update formula for the Digging phase, after introducing the adaptive density factor in HBA, is presented in Eq. (10), while the position update formula for the Honey phase is presented in Eq. (11).

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times w \times d_i \times |\cos(2\pi r_4) \times [1-\cos(2\pi r_5)]| \quad (10)$$

$$x_{new} = x_{prey} + F \times r_7 \times w \times d_i \quad (11)$$

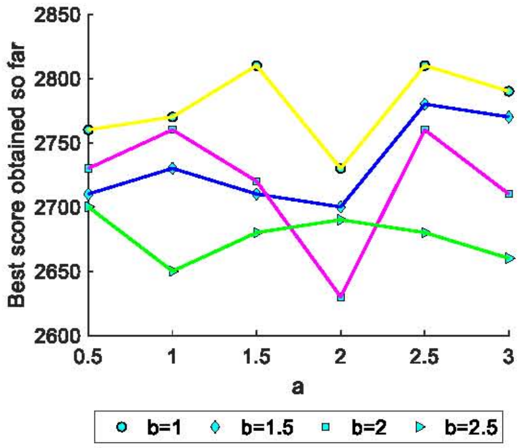


Fig.3. Fitness achieved by OHBA for parameters 'a' and 'b'

The HBA performs global exploration based on formula (10). When the honey badger is exploring for the global optimal solution, it employs a larger weight, enabling an extensive search to maintain the diversity of OHBA. In contrast, during local search based on formula (11), a smaller weight is utilized. This extends the search duration during later iterations, enhancing the algorithm's local search capabilities and, consequently, improving its optimization effectiveness.

C. Beta distribution

The HBA algorithm encounters challenges associated with the imbalance between exploration and exploitation, often requiring a considerable number of iterations to converge to the optimal value. In the realm of probability and statistics, the probability distribution function stands as one of the most critical characteristics of a random variable. It serves as a valuable tool for studying the statistical properties of random variables through mathematical analysis. Distributions extend the concept of functions beyond the ordinary sense, and even for functions that lack differentiability or are discontinuous in the conventional sense, they can possess derivatives in the distribution sense.

Hence, we have chosen several distributions, including the Beta distribution, Gamma distribution, Poisson distribution, Weibull distribution, Gaussian distribution, Exponential distribution, and Binomial distribution. Random numbers generated from these diverse distributions were incorporated into the HBA algorithm to evaluate their influence on the algorithm's performance. The convergence curves are illustrated in Fig.4.

Based on the observations from Fig.4, it is evident that the Beta distribution is better suited for enhancing the HBA algorithm's performance. Consequently, the Beta distribution is chosen to be integrated into the HBA algorithm. The Beta distribution is a continuous probability distribution and serves as the conjugate prior distribution for the Bernoulli and Binomial distributions. It finds significant applications in machine learning and mathematical statistics. The mathematical expression of the Beta distribution is presented in equation (12).

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \quad (12)$$

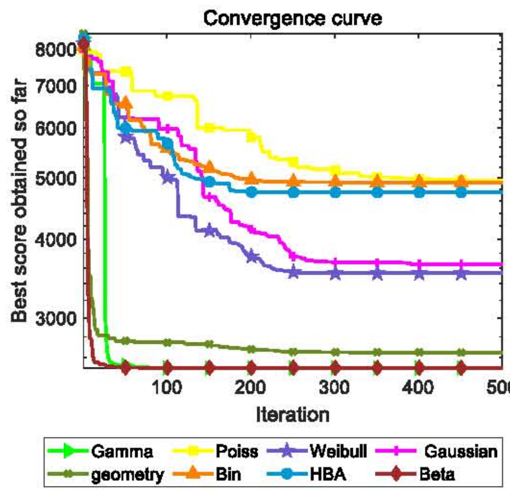


Fig.4. Convergence graphs for different distributions

Compared to the Uniform distribution, the Beta distribution provides the HBA algorithm with enhanced exploration capabilities. In the later stages of the algorithm, it can boost the algorithm's global exploration abilities and reduce the likelihood of the algorithm becoming trapped in local optima. When the algorithm encounters a local optimum, the Beta distribution can assist the HBA algorithm in escaping from it. Importantly, to preserve the core principles of the algorithm, the values generated by the Beta distribution ultimately fall within the range (0,1). The comparison between the Uniform distribution and the Beta distribution within the interval (0, 1) is illustrated in Fig.5 and Fig.6.

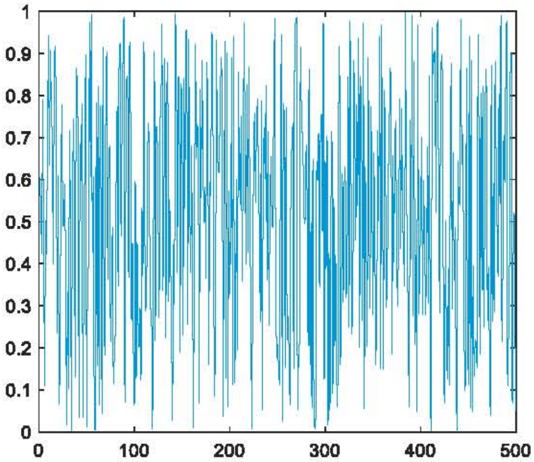


Fig.5. Generating random variates from a Uniform distribution

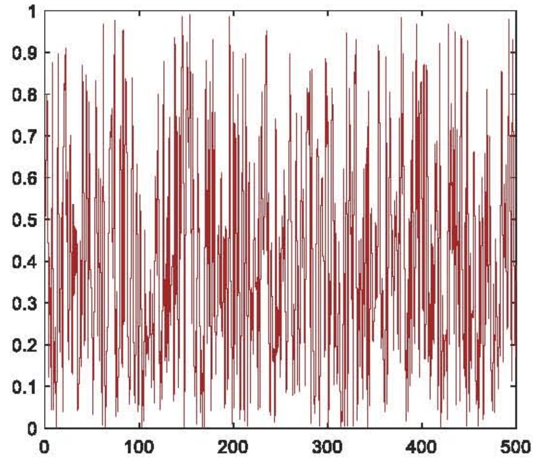


Fig.6. Generating random variates from a Beta distribution

D. Cauchy mutation based on the Sine chaotic mapping

To combat the problem of the HBA being susceptible to getting stuck in local optima, this paper introduces Cauchy mutation based on the Sine chaotic mapping during iterations. In each iteration, a sequence generated by the Sine chaotic mapping is utilized instead of randomly generated variables. The inherently chaotic and exploratory characteristics of the Sine chaotic mapping are leveraged to navigate the solution space. This enhancement effectively bolsters the algorithm's optimization performance and elevates its capability to uncover the optimal solution.

In the HBA, prior to convergence, the selection of the target individual consistently takes place from among the previous candidate solutions, often resulting in a lack of diversity. To address this limitation, the introduction of a mutation strategy that applies to the target individuals in each iteration proves beneficial. This strategy expands the search space, augments population diversity, and contributes to enhancing the algorithm's capacity for local exploration, enabling the discovery of new optimal solutions.

Cauchy mutation and Gaussian mutation represent two distinct strategies employed in optimization algorithms for generating new solutions. In this paper, both Cauchy mutation and Gaussian mutation are chosen to mutate candidate solutions, with the aim of determining which mutation strategy is better suited for the Honey Badger Algorithm. The formulas for the Cauchy distribution and Gaussian distribution functions are presented in equations (13) and (14), respectively. These two distribution functions are used to guide the mutation process, and their effectiveness in optimizing the HBA is being evaluated. Fig.7 and Fig.8 illustrate the comparative visualizations of the probability density functions and boxplots for the Cauchy and Gaussian distributions.

$$f(x)=\frac{1}{\pi(x^2+1)} \tag{13}$$

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{14}$$

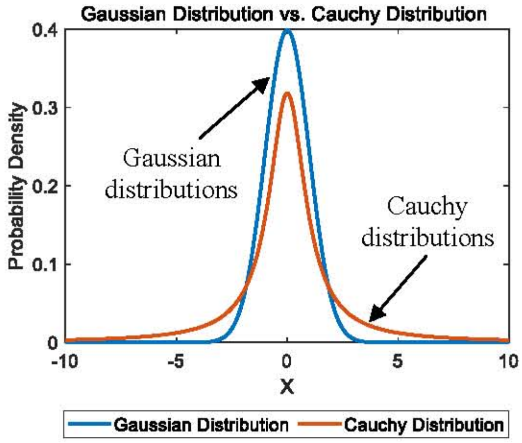


Fig.7. The probability density

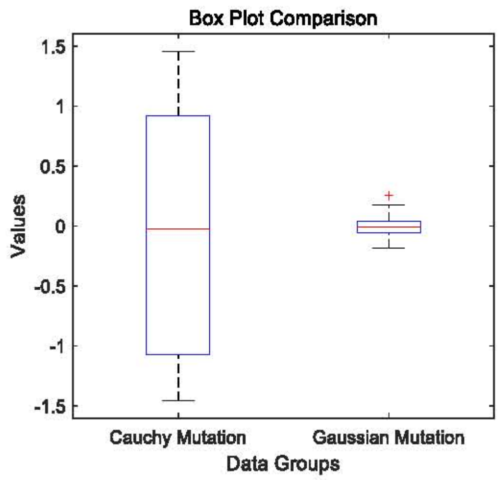


Fig.8. Box plot comparison

Based on Fig.7, it can be observed that the Cauchy distribution function has a smaller peak at the origin, and its distribution decreases more gradually towards both ends, while the Gaussian distribution decreases relatively quickly from the peak towards both ends. According to Fig.6, Cauchy mutation tends to produce larger mutations, while Gaussian mutation tends to produce smaller mutations. Cauchy mutation exhibits stronger perturbation capability compared to Gaussian mutation. By utilizing Cauchy mutation, the perturbation near the current mutated honey badger individual becomes stronger, allowing for rapid search of promising solutions in that region. This enhances population diversity, prevents the algorithm from becoming trapped in local optima, and improves its ability to search for the global optimal solution. The perturbation formula is shown in equation (14):

$$x_{newbest} = x_{prey} + x_{prey} \times Cauchy(0,1) \qquad (14)$$

Where x_{prey} represents the mutated position, $x_{newbest}$ is the current position, r is a random factor, and Cauchy (0,1) represents the Cauchy-distributed random variable.

E. The flowchart of OHBA

To offer a lucid portrayal of the proposed algorithm's structure, Fig.9 presents the flowchart of OHBA. This visual representation can significantly facilitate the comprehension of the program framework.

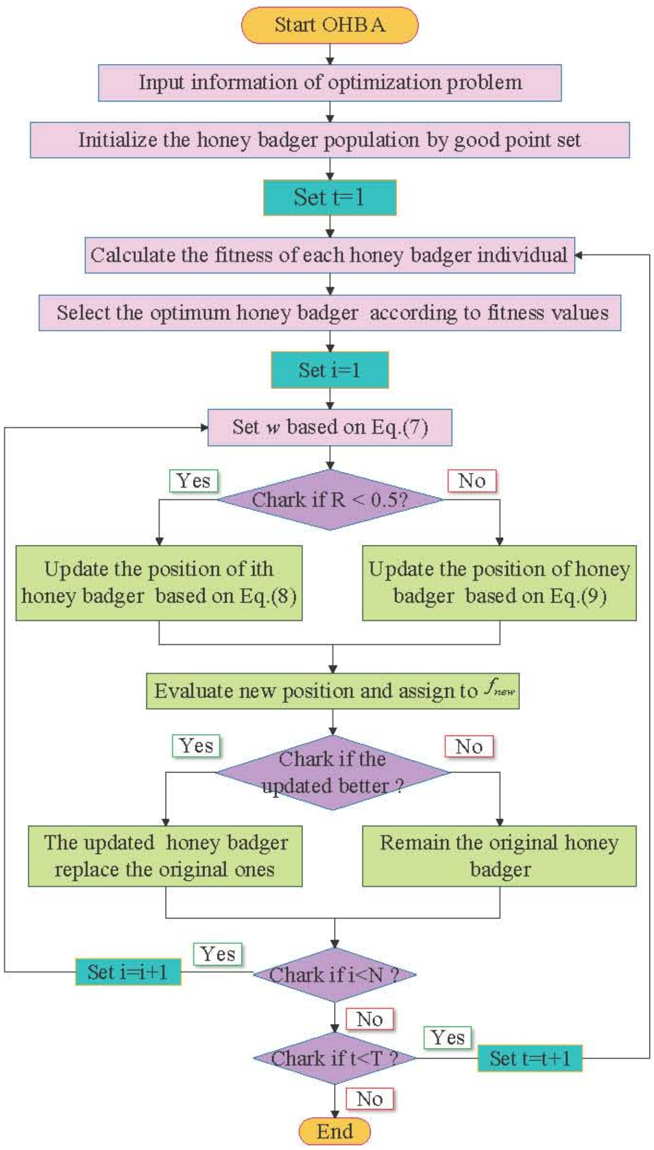


Fig.9. The flowchart of OHBA

IV. Experimental Results and Discussions

To assess the optimization performance of the OHBA, two distinct sets of test functions have been introduced. Additionally, the Wilcoxon rank-sum test is employed to evaluate the OHBA algorithm's performance. This comprehensive approach is designed to rigorously examine and validate the algorithm's capabilities across a range of test scenarios.

A. Benchmark test functions and algorithms for comparison

For this study, the experimental setup employed the following environment and hardware specifications: MATLAB 2021a simulation software, Windows 10 operating system. A machine with a clock speed of 2.60GHz. 8GB of RAM. These specifications outline the computational environment used for conducting the experiments and evaluating the OHBA algorithm's performance.

To validate the optimization performance of the Improved Honey Badger Algorithm Based on Hybrid Strategy, a comprehensive evaluation was conducted using a diverse set of benchmarks. Specifically, the study employed CEC-2017 and CEC-2022, along with the Wilcoxon rank-sum test to assess the algorithm's performance. In this comparative analysis, OHBA was pitted against several other algorithms, including HBA [20], Whale Optimization Algorithm(WOA) [34], Sine Cosine Algorithm (SCA) [35], Butterfly Optimization Algorithm (BOA) [36], Moth-Flame Optimization (MFO) [37], and Dung Beetle Optimizer (DBO) [38]. The primary objective of this experiment was to rigorously evaluate and compare the performance of OHBA against these established algorithms. To ensure the fairness and reliability of the experiments, each algorithm underwent independent runs 30 times for each test function. The experiments were conducted with a fixed population size of 30 individuals and a maximum iteration count of 500. This methodology ensured a robust and consistent assessment of algorithm performance across different test scenarios. The parameters used for each algorithm were selected in accordance with their respective original algorithm papers, and Table II presents a comparison of these different algorithm parameters.

To further demonstrate OHBA's superiority, we conducted a comprehensive comparison among various algorithms based on solution precision for each test function. This comparison included their best values (Best), average values (Mean), and standard deviations (Std). These metrics provide insights into different aspects of algorithm performance: the best value represents the algorithm's peak solving capability, a higher "Best" value signifies a superior proficiency in unearthing global optima. The average value indicates overall precision, a lower " Mean" value indicates a better convergence accuracy of the algorithm. The standard deviation signifies stability and robustness, a lower "Std" suggests greater reliability. In addition to these statistical measures, we recorded convergence curves, which offer a visual representation facilitating a more intuitive observation and comparison of each algorithm's convergence speed, precision, and ability to navigate away from local optima. This comprehensive evaluation aimed to showcase OHBA's advantages relative to other algorithms.

TABLE II
ALGORITHM PARAMETER SETTINGS

Algorithm	Parameter	Value
WOA	r	[0,1]
SCA	a	2
DBO	k and λ	0.1
	β	0.3
	S	0.5
MFO	a	[-2, -1]
	b	1
BOA	p	0.8
	c	0.1
	a	0.01
HBA	C	2
	β	6
OHBA	C	2
	β	6

B. The performance of the OHBA on the CEC-2017

To evaluate OHBA's capabilities in exploration, exploitation, and avoidance of local minima, this study selected 15 CEC-2017 test functions encompassing various characteristics. These functions encompass single-peaked, multi-modal, composite, and hybrid attributes, providing a comprehensive assessment of OHBA's performance. Functions F1 and F2 represent single-peaked functions, each possessing a single global optimum within the solution space. These functions serve as a litmus test for evaluating search accuracy and convergence speed. Conversely, functions F3 to F5 are multi-modal in nature, featuring multiple local optima, which pose a challenge for optimization algorithms prone to getting trapped in local minima. The results obtained from these multi-modal functions effectively showcase the algorithm's prowess in function optimization. Additionally, functions F6 to F10 are composite functions, while functions F11 to F15 are hybrid functions. The unique characteristics of each function are detailed in Table III.

OHBA underwent testing on this diverse set of functions, and its performance was systematically compared with other state-of-the-art optimization algorithms. Fig.10 presents the average convergence curves obtained from 30 independent runs of OHBA and the comparative algorithms. Complementing this, Table IV provides the corresponding statistical metrics, including the mean, standard deviation, and best values, also averaged over 30 runs. In Table IV. solutions characterized by the highest precision have been accentuated by employing bold text.

Derived from Fig.10, it becomes apparent that OHBA consistently upholds robust exploration and exploitation capabilities throughout the optimization procedure. Upon reviewing the outcomes, it becomes evident that OHBA provides commendable convergence speed and accuracy in comparison to other optimization algorithms for the majority of functions. In the case of unimodal functions such as F1 and F2, OHBA's convergence speed and precision exhibit a marginal superiority over the original HBA. Conversely, for multimodal functions like F4 and F5, OHBA markedly surpasses other optimization algorithms. Composite functions, which are inherently more complex than basic multimodal functions, offer a superior simulation of intricate real-world scenarios. OHBA also demonstrates significant advantages when applied to hybrid and composite functions. It maintains robust exploration and exploitation capabilities throughout the entirety of the search process. Nevertheless, due to the intricacy of specific individual functions, OHBA may occasionally become ensnared in local optima, leading to suboptimal results.

As illustrated in Table IV, the OHBA algorithm showcases remarkable advantages in optimizing a wide array of functions. Specifically, for single-peaked functions F1 and F2, OHBA not only attains the best averages but also reaches the theoretical optimum in F2. In these cases, OHBA's overall performance outshines that of the comparison algorithms. In the realm of multi-modal functions, OHBA consistently achieves the lowest averages for functions F3, F4, and F5. Notably, for function F4, OHBA excels in terms of mean, standard deviation, and best value, marking a significant enhancement in its problem-solving capabilities compared to other algorithms. Among the composite functions, OHBA secures the top results in terms of mean, standard deviation, and best value for functions F6, F9, and F10. Furthermore, it exhibits excellent performance in functions F7 and F8 when compared to alternative algorithms, underscoring its robustness. Composite functions, characterized by their higher complexity compared to basic multi-modal and hybrid functions, also yield favorable outcomes with OHBA. Specifically, OHBA achieves the best averages for five functions: F11, F12, F13, F14, and F15. While, in the cases of functions F13 and F14, its standard deviation and best value slightly trail behind SCA and DBO, the comprehensive performance displayed by OHBA, as depicted in Table IV, is undeniably exceptional. It distinguishes OHBA as a highly advantageous algorithm when compared to its counterparts.

TABLE III
PROPERTIES AND SUMMARY OF THE CEC2017 TEST FUNCTIONS

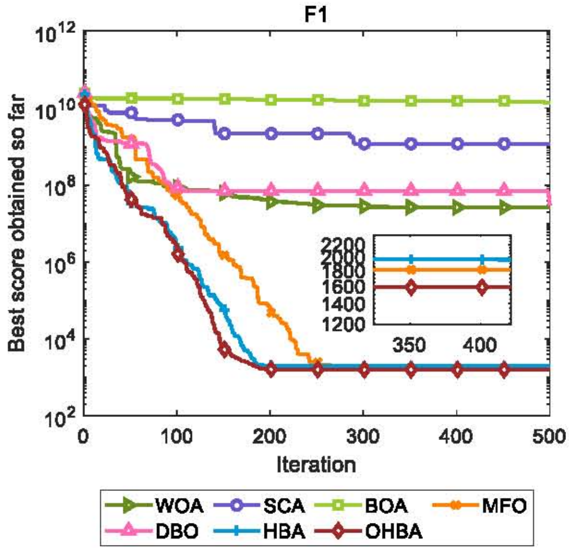
Benchmark functions	F	Descriptions	Dim	Range	f_{min}
Unimodal Function	F1	Shifted and Rotated Bent Cigar Function	10	[-100,100]	100
	F2	Shifted and Rotated Zakharov Function	10	[-100,100]	300
Multimodal Functions	F3	Shifted and Rotated Rosenbrock's Function	10	[-100,100]	400
	F4	Shifted and Rotated Rastrigin's Function	10	[-100,100]	500
	F5	Shifted and Rotated Schwefels Function	10	[-100,100]	1000
Hybrid Functions	F6	Hybrid Function 1 (N = 3)	10	[-100,100]	1100
	F7	Hybrid Function 2 (N = 3)	10	[-100,100]	1200
	F8	Hybrid Function 3 (N = 3)	10	[-100,100]	1300
	F9	Hybrid Function 6 (N = 5)	10	[-100,100]	1800
	F10	Hybrid Function 6 (N = 5)	10	[-100,100]	1900
Composition Functions	F11	Composition Function3 (N = 4)	10	[-100,100]	2300
	F12	Composition Function4 (N = 4)	10	[-100,100]	2400
	F13	Composition Function5 (N = 5)	10	[-100,100]	2500
	F14	Composition Function6 (N = 5)	10	[-100,100]	2600
	F15	Composition Function9 (N = 3)	10	[-100,100]	2900

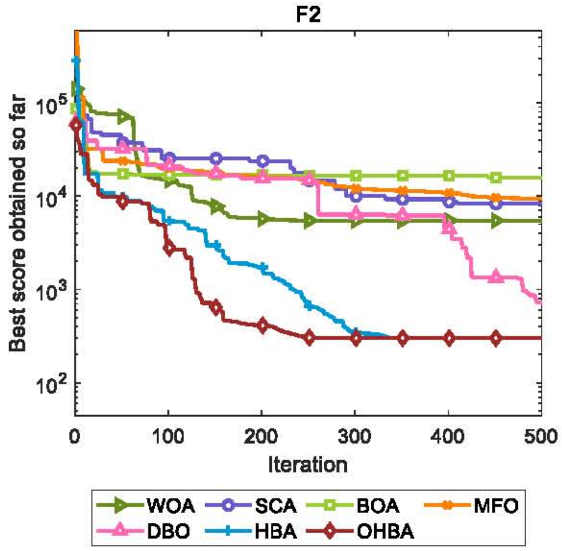
TABLE IV
PROPERTIES AND SUMMARY OF THE CEC2017 TEST FUNCTIONS

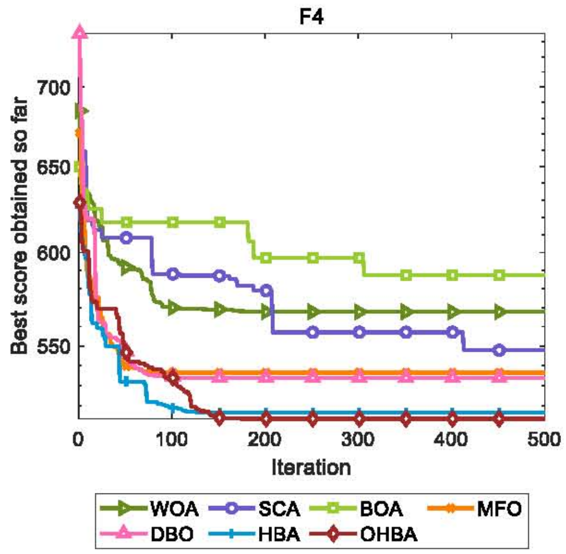
F	Statistics	HBA	WOA	SCA	BOA	MFO	DBO	OHBA
F1	Avg	4.7600E+03	6.6890E+06	8.4318E+08	7.3808E+09	1.8295E+08	1.8295E+08	**4.5191E+03**
	Std	**3.7178E+03**	8.9309E+06	2.9287E+08	3.9444E+09	4.7871E+08	4.7871E+08	3.9399E+03
	Best	1.4949E+02	9.0952E+05	4.4709E+08	1.5341E+09	1.2200E+02	1.2200E+02	**1.0929E+02**
F2	Avg	**3.0000E+02**	3.8830E+03	2.0558E+03	1.2902E+04	1.2742E+04	1.2742E+04	3.0001E+02
	Std	**3.4886E-07**	4.7565E+03	1.2653E+03	2.0674E+03	1.1694E+04	1.1694E+04	2.9200E-02
	Best	**3.0000E+02**	3.9792E+02	8.2562E+02	7.0807E+03	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**
F3	Avg	4.0816E+02	4.3995E+02	4.5414E+02	1.8013E+03	4.2371E+02	4.2371E+02	**4.0395E+02**
	Std	1.9187E+01	5.1965E+01	2.0878E+01	6.1605E+02	3.4896E+01	3.4896E+01	**9.8240E-01**
	Best	4.0009E+02	4.0076E+02	4.2299E+02	6.0889E+02	**4.0001E+02**	**4.0001E+02**	4.0057E+02
F4	Avg	5.2402E+02	5.5372E+02	5.5053E+02	5.9381E+02	5.3457E+02	5.3457E+02	**5.2202E+02**
	Std	1.0596E+01	2.5434E+01	5.8030E+00	1.6465E+01	1.1964E+01	1.1964E+01	**9.3486E+00**
	Best	5.0995E+02	5.2120E+02	5.3950E+02	5.5807E+02	5.1181E+02	5.1181E+02	**5.0796E+02**
F5	Avg	2.0154E+03	2.1992E+03	2.3700E+03	2.6715E+03	2.1039E+03	1.9900E+03	**1.7305E+03**
	Std	4.4151E+02	3.1509E+02	2.4621E+02	**2.1741E+02**	4.3308E+02	3.9668E+02	3.2900E+02
	Best	1.2487E+03	1.6842E+03	1.4690E+03	2.0142E+03	1.4198E+03	1.1677E+03	**1.1641E+03**
F6	Avg	1.1241E+03	1.2108E+03	1.2084E+03	1.8894E+03	1.3217E+03	1.2315E+03	**1.1222E+03**
	Std	2.4209E+01	8.6856E+01	5.2950E+01	7.5725E+02	4.9363E+02	1.3109E+02	**1.9412E+01**
	Best	1.1030E+03	1.1156E+03	1.1523E+03	1.2286E+03	1.1014E+03	1.1161E+03	**1.1013E+03**
F7	Avg	1.7240E+04	6.7146E+06	1.6501E+07	2.7722E+08	1.2499E+06	1.0071E+06	**1.5725E+04**
	Std	**1.6215E+04**	6.4151E+06	1.1541E+07	4.2599E+08	3.1221E+06	2.5377E+06	1.1628E+04
	Best	2.8904E+03	3.1119E+04	4.1481E+06	6.8646E+06	**2.4442E+03**	2.0770E+03	3.9370E+03
F8	Avg	**5.6008E+03**	1.5201E+04	5.0525E+04	5.6067E+05	1.2641E+04	1.3510E+04	6.3625E+03
	Std	5.9280E+03	1.0939E+04	3.8357E+04	7.9837E+05	1.1775E+04	1.2862E+04	**4.1189E+03**
	Best	1.5787E+03	2.6078E+03	8.5202E+03	1.8539E+04	**1.3324E+03**	1.3950E+03	1.4855E+03
F9	Avg	1.3167E+04	1.9305E+04	2.5042E+05	3.4292E+06	2.5792E+04	1.7216E+04	**5.2833E+03**
	Std	1.3315E+04	1.2956E+04	2.2439E+05	7.7103E+06	1.5784E+04	1.4083E+04	**2.1633E+03**
	Best	2.3406E+03	2.4914E+03	9.9949E+03	1.0939E+05	1.8555E+03	2.1452E+03	**1.9306E+03**
F10	Avg	3.0930E+03	5.5177E+04	8.2004E+03	1.0425E+05	1.3669E+04	8.4642E+03	**2.2282E+03**
	Std	5.6719E+03	1.2039E+05	5.9387E+03	1.4575E+05	1.2374E+04	1.3522E+04	**6.1245E+02**
	Best	**1.9151E+03**	2.0623E+03	2.0571E+03	2.4018E+03	2.0510E+03	1.9666E+03	1.9230E+03

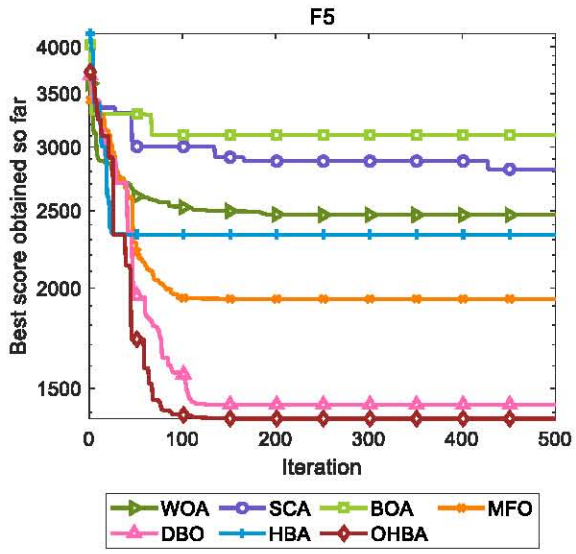
TABLE IV
PROPERTIES AND SUMMARY OF THE CEC2017 TEST FUNCTIONS

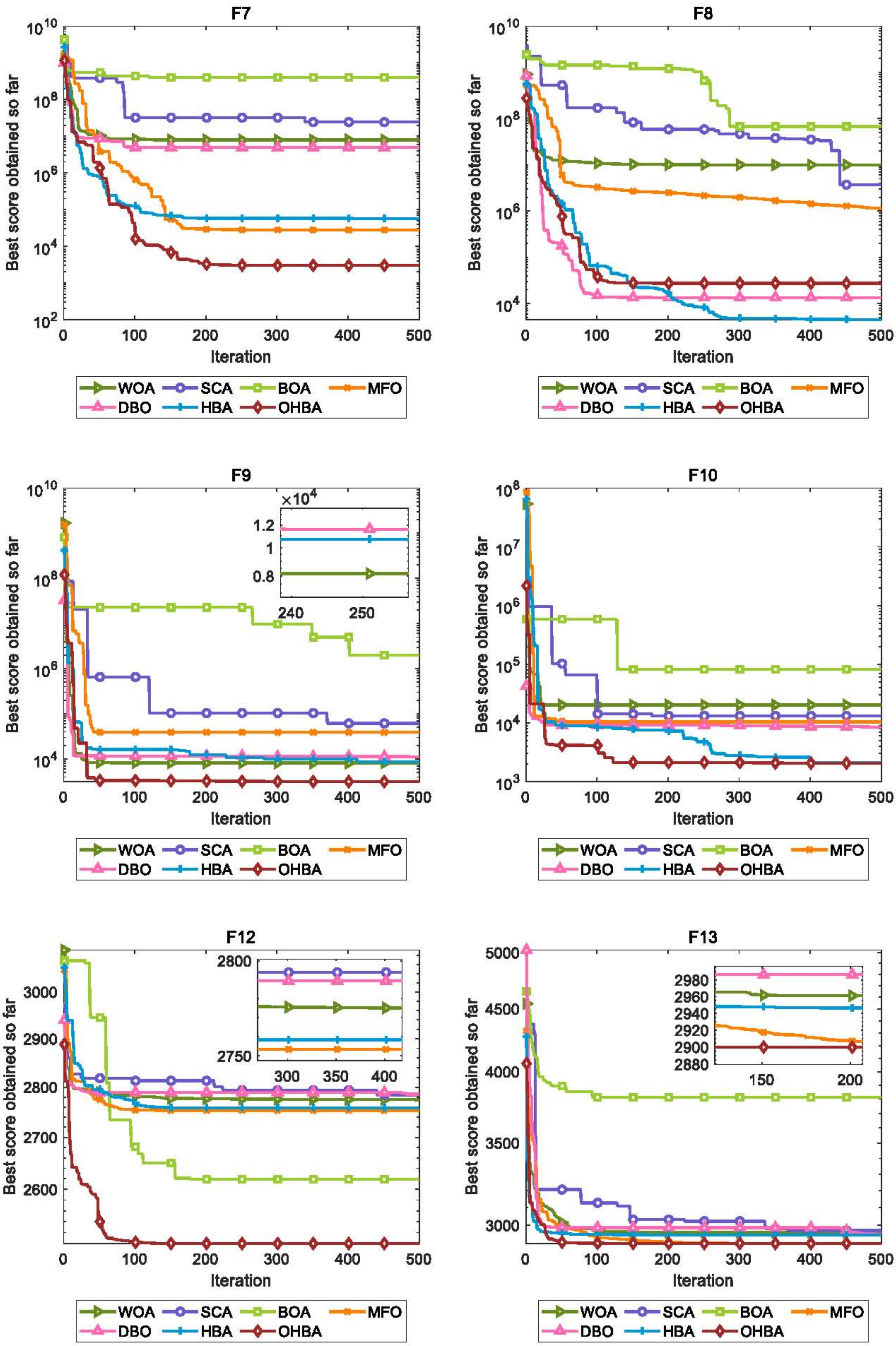
F	Statistics	HBA	WOA	SCA	BOA	MFO	DBO	OHBA
F11	Avg	2.6339E+03	2.6459E+03	2.6572E+03	2.6802E+03	2.6298E+03	2.6535E+03	**2.6281E+03**
	Std	2.1037E+01	2.4765E+01	**8.6603E+00**	2.5118E+01	1.1226E+01	1.9209E+01	1.2975E+01
	Best	2.6124E+03	2.6156E+03	2.6377E+03	2.6342E+03	2.6129E+03	2.6298E+03	**2.6054E+03**
F12	Avg	2.7403E+03	2.7834E+03	2.7794E+03	2.7072E+03	2.7609E+03	2.7034E+03	**2.7015E+03**
	Std	6.6676E+01	5.2187E+01	3.9211E+01	7.8761E+01	**1.0806E+01**	1.0370E+02	1.1377E+02
	Best	2.5000E+03	2.5701E+03	2.5772E+03	2.5558E+03	2.7427E+03	2.5000E+03	**2.5000E+03**
F13	Avg	2.9253E+03	2.9439E+03	2.9654E+03	3.6718E+03	2.9289E+03	2.9305E+03	**2.9128E+03**
	Std	2.3758E+01	2.7694E+01	**1.3703E+01**	2.3309E+02	2.4672E+01	6.7718E+01	2.1656E+01
	Best	2.8978E+03	2.9028E+03	2.9239E+03	3.1903E+03	2.8983E+03	**2.6001E+03**	2.8977E+03
F14	Avg	3.0252E+03	3.5055E+03	3.0826E+03	3.2242E+03	3.1278E+03	3.0647E+03	**3.0026E+03**
	Std	3.2472E+02	6.2095E+02	**3.9140E+01**	2.0923E+02	3.2918E+02	1.3953E+02	7.7882E+01
	Best	**2.6000E+03**	2.6166E+03	3.0212E+03	2.7990E+03	2.9000E+03	**2.6000E+03**	2.9000E+03
F15	Avg	3.2658E+03	3.3537E+03	3.2539E+03	3.3669E+03	3.2378E+03	3.2294E+03	**3.2228E+03**
	Std	8.6870E+01	9.7884E+01	3.5855E+01	6.9930E+01	6.3152E+01	5.1681E+01	**4.7004E+01**
	Best	3.1529E+03	3.2057E+03	3.2044E+03	3.2276E+03	**3.1431E+03**	3.1548E+03	3.1559E+03










F7
F8
F9
F10
F12
F13
Best score obtained so far
Iteration
WOA
SCA
BOA
MFO
DBO
HBA
OHBA

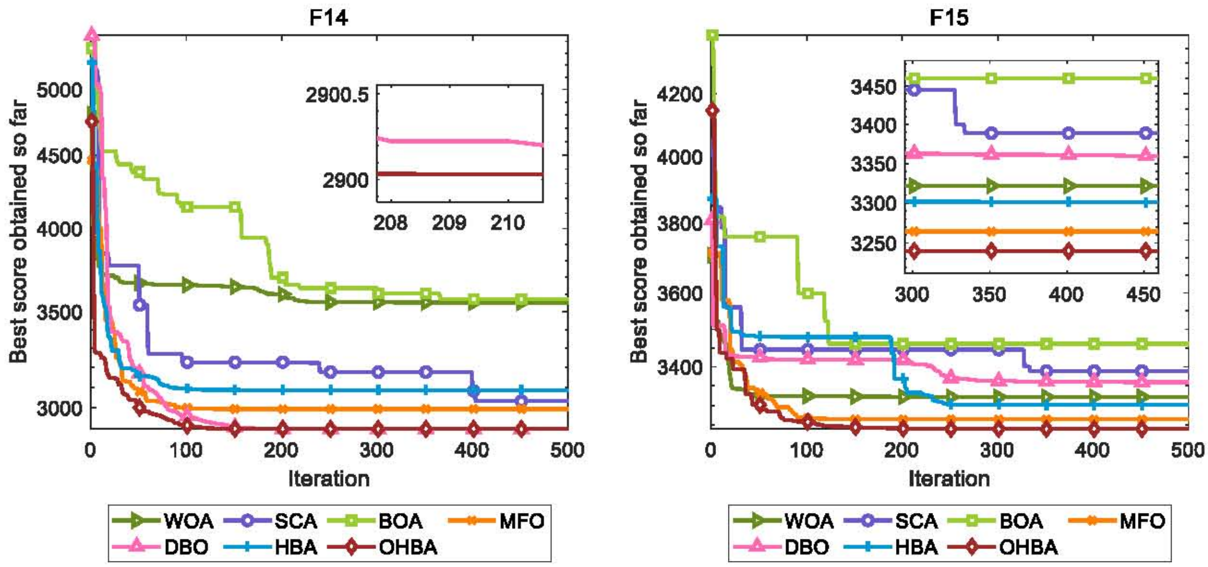


Fig.10. Comparison of convergence curves for CEC-2017 benchmark functions

C. The performance of the OHBA on the CEC-2022

To further validate the OHBA algorithm's proficiency in tackling intricate problems, rigorous testing was undertaken employing the CECC-2022 functions. CEC-2022, being a relatively recent test suite, presents distinctive function characteristics detailed in Table V. Fig.11 furnishes the average convergence curves, derived from 30 independent runs of OHBA and comparative algorithms, each comprising 500 iterations. Concurrently, Table VI encompasses the statistical metrics encompassing mean, standard deviation, and best values, acquired from 30 executions of each algorithm. In Table VI, solutions characterized by the highest precision have been accentuated by employing bold text.

Derived from Fig.11, it becomes evident that OHBA provides commendable convergence speed and accuracy in comparison to alternative optimization algorithms for the majority of functions. In the case of unimodal functions, it becomes apparent that OHBA consistently HBA's convergence speed and precision exhibit a marginal superiority over the original HBA. Conversely, for multimodal functions, OHBA markedly surpasses other optimization algorithms. OHBA also demonstrates significant advantages when applied to hybrid and composite functions. It maintains robust exploration and exploitation capabilities throughout the entirety of the search process. Nevertheless, due to the intricacy of specific individual functions, OHBA may occasionally become ensnared in local optima, leading to suboptimal results.

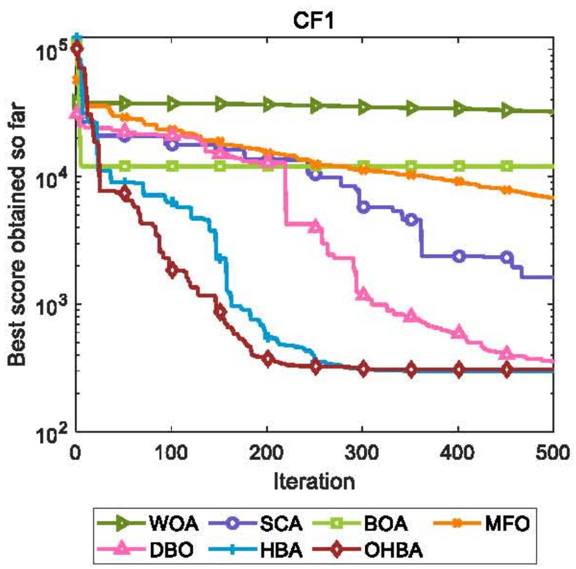
Analysis of the data presented in Table VI underscores OHBA's superior performance across several key metrics. Notably, OHBA excels in terms of mean values in a range of test functions, including F2, F5, F6, F8, F9, F10, and F11. It also attains the best mean and standard deviation results in functions F5 and F6. These findings collectively highlight OHBA's exceptional performance in comparison to HBA, with the exception of specific instances where it may not perform as strongly, as observed in functions F1, F4, and F7. Across the majority of the tested functions, OHBA consistently outperforms HBA and other optimization algorithms. This improvement is especially evident in optimization precision and convergence performance. Furthermore, OHBA's demonstrated robustness further solidifies its position as a highly effective optimization strategy. In conclusion, these results provide compelling evidence of the efficacy of the improvement strategy employed by OHBA.

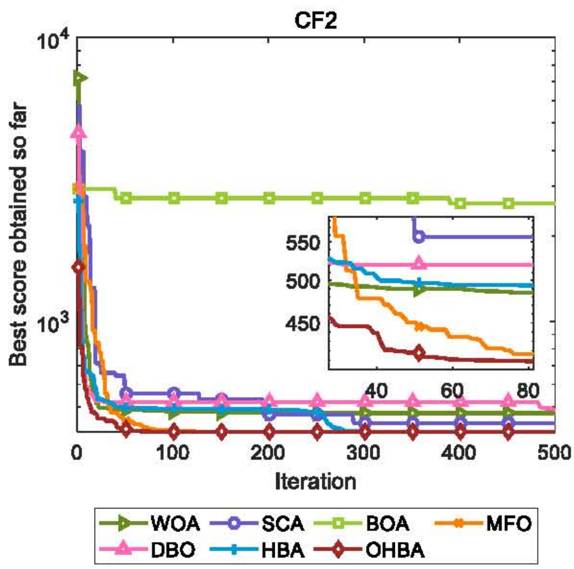
TABLE V
PROPERTIES AND SUMMARY OF THE CEC2022 TEST FUNCTIONS

Benchmark functions	No.	Descriptions	Dim	Range	f_{min}
Unimodal	CF1	Shifted and full Rotated Zakharov Function	10	[-100,100]	300
Basic Functions	CF2	Shifted and full Rotated Rosenbrock's Function	10	[-100,100]	400
	CF3	Shifted and full Rotated Rastrigin's Function	10	[-100,100]	600
	CF4	Shifted and full Rotated Non-Continuous Rastrigin's Function	10	[-100,100]	800
	CF5	Shifted and full Rotated Levy Function	10	[-100,100]	900
Hybrid Functions	CF6	Hybrid Function 1 (N=3)	10	[-100,100]	1800
	CF7	Hybrid Function 2 (N=6)	10	[-100,100]	2000
	CF8	Hybrid Function 3 (N=5)	10	[-100,100]	2200
Composition Functions	CF9	Composition Function 1 (N=5)	10	[-100,100]	2300
	CF10	Composition Function 2 (N=4)	10	[-100,100]	2400
	CF11	Composition Function 3 (N=5)	10	[-100,100]	2600
	CF12	Composition Function 4 (N=6)	10	[-100,100]	2700

TABLE VI
PROPERTIES AND SUMMARY OF THE CEC2022 TEST FUNCTIONS

CF	Statistics	HBA	WOA	SCA	BOA	MFO	DBO	OHBA
CF1	Avg	**3.0002E+02**	2.5413E+04	2.7079E+03	1.0278E+04	7.5094E+03	2.0135E+03	3.2701E+02
	Std	**5.1400E-02**	1.0420E+04	1.4535E+03	2.3604E+03	7.6546E+03	2.0130E+03	4.8751E+01
	Best	**3.0000E+02**	8.5832E+03	1.0452E+03	5.4519E+03	3.2421E+02	3.0141E+02	3.0050E+02
CF2	Avg	4.1228E+02	4.6366E+02	4.7986E+02	2.5594E+03	4.2063E+02	4.4196E+02	**4.0692E+02**
	Std	2.0412E+01	8.2639E+01	1.9255E+01	9.6231E+02	2.4180E+01	4.8688E+01	**3.8750E+00**
	Best	**4.0000E+02**	4.0103E+02	4.5364E+02	7.6146E+02	4.0399E+02	4.0039E+02	**4.0000E+02**
CF3	Avg	**6.0040E+02**	6.4302E+02	6.2295E+02	6.4462E+02	6.0501E+02	6.0988E+02	6.0519E+02
	Std	**8.7740E-01**	1.8505E+01	4.7089E+00	8.8220E+00	6.6786E+00	9.0351E+00	4.7512E+00
	Best	**6.0000E+02**	6.1203E+02	6.1446E+02	6.2594E+02	**6.0000E+02**	6.0004E+02	6.0013E+02
CF4	Avg	**8.1951E+02**	8.4683E+02	8.4557E+02	8.5340E+02	8.3161E+02	8.2948E+02	8.2368E+02
	Std	8.6578E+00	1.3004E+01	7.9599E+00	7.3085E+00	1.0886E+01	8.7991E+00	**6.6121E+00**
	Best	**8.0895E+02**	8.2321E+02	8.3156E+02	8.3479E+02	8.1293E+02	8.1293E+02	8.1194E+02
CF5	Avg	**9.4078E+02**	1.4891E+03	1.0607E+03	1.3637E+03	1.0650E+03	9.9716E+02	9.8611E+02
	Std	5.3992E+01	3.8144E+02	8.2954E+01	1.9183E+02	3.0300E+02	1.0803E+02	**5.1210E+01**
	Best	**9.0018E+02**	1.0318E+03	9.5574E+02	1.0193E+03	9.0000E+02	9.0500E+02	9.1194E+02
CF6	Avg	4.3724E+03	6.1390E+03	5.5127E+06	2.2643E+08	4.4522E+03	5.7304E+03	**4.2721E+03**
	Std	2.1743E+03	4.2178E+03	5.1170E+06	4.4822E+08	2.2165E+03	2.1168E+03	**1.8087E+03**
	Best	**1.8861E+03**	2.1949E+03	3.9587E+05	5.9673E+06	1.8960E+03	2.5308E+03	1.9265E+03
CF7	Avg	**2.0228E+03**	2.0907E+03	2.0628E+03	2.0942E+03	2.0305E+03	2.0407E+03	2.0446E+03
	Std	**5.1863E+00**	3.0713E+01	1.1150E+01	1.3708E+01	1.1769E+01	1.5919E+01	4.1109E+01
	Best	2.0096E+03	2.0436E+03	2.0388E+03	2.0680E+03	2.0201E+03	2.0201E+03	**2.0010E+03**
CF8	Avg	2.2278E+03	2.2404E+03	2.2361E+03	2.3937E+03	**2.2251E+03**	2.2300E+03	2.2297E+03
	Std	2.2690E+01	9.7616E+00	3.6298E+00	1.7443E+02	**3.9056E+00**	8.3653E+00	2.3652E+01
	Best	2.2204E+03	2.2275E+03	2.2281E+03	2.2376E+03	2.2205E+03	2.2207E+03	**2.2091E+03**
CF9	Avg	2.5319E+03	2.6090E+03	2.5803E+03	2.8282E+03	2.5361E+03	2.5554E+03	**2.5306E+03**
	Std	6.0303E+00	5.5967E+01	2.4135E+01	7.2089E+01	1.9277E+01	3.5848E+01	**3.8040E+00**
	Best	**2.5293E+033**	2.5305E+03	2.5463E+03	2.6267E+03	**2.5293E+03**	**2.5293E+03**	**2.5293E+03**
CF10	Avg	2.6348E+03	2.6203E+03	2.5127E+03	2.5263E+03	2.5679E+03	2.5389E+03	**2.5006E+03**
	Std	2.3011E+02	2.1056E+02	3.7739E+01	4.7029E+01	1.4874E+02	6.4369E+01	**2.0750E-01**
	Best	2.5004E+03	2.5005E+03	2.5014E+03	2.5018E+03	**2.5003E+03**	2.5005E+03	**2.5003E+03**
CF11	Avg	2.6800E+03	2.8362E+03	2.8044E+03	3.4371E+03	2.7726E+03	2.7977E+03	**2.6185E+03**
	Std	1.4948E+02	1.6795E+02	8.9465E+01	5.4012E+02	1.1829E+02	1.6929E+02	**7.7476E+01**
	Best	**2.6000E+03**	2.6557E+03	2.7558E+03	2.8225E+03	**2.6000E+03**	2.6100E+03	**2.6000E+03**
CF12	Avg	2.8910E+03	2.9082E+03	2.8721E+03	2.9535E+03	**2.8646E+03**	2.8696E+03	2.9434E+03
	Std	3.1328E+01	3.2361E+01	3.2441E+00	4.5581E+01	**2.5057E+00**	6.2048E+00	4.2404E+01
	Best	2.8626E+03	2.8653E+03	2.8642E+03	2.8744E+03	**2.8592E+03**	2.8632E+03	2.8644E+03





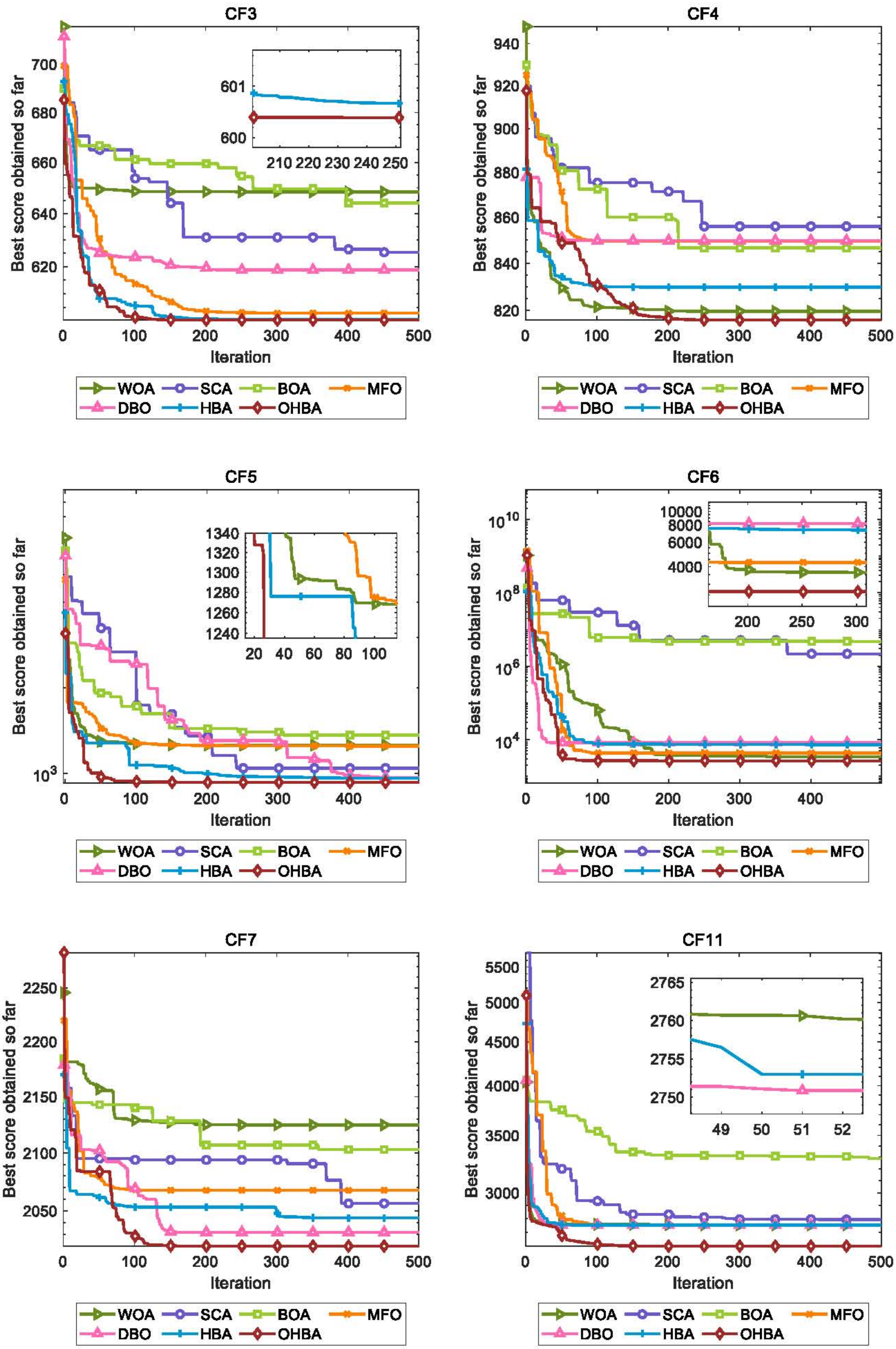


Fig.11. Comparison of convergence curves for CEC-2022 benchmark functions

D. Wilcoxon rank-sum tests

Robustness, in the context of an algorithm, refers to its ability to maintain stability when confronted with uncertainties or exceptional circumstances. It stands as a pivotal metric for assessing algorithmic performance. To further substantiate OHBA's robustness, an evaluation was conducted employing the Wilcoxon rank-sum test. This statistical analysis sought to determine whether OHBA exhibited statistically significant differences in performance compared to other algorithms. Calculations were carried out at a significance level of 5%. A resulting p-value less than 5% would signify a noteworthy disparity in optimization outcomes between the two algorithms, while a p-value greater than 5% would indicate that the optimization results of the two algorithms are generally similar.

In this section, OHBA is compared with standard HBA, WOA, BOA, MFO, SCA, and DBO algorithms across 10 test functions. The function characteristics, including expressions, dimensions, search ranges, and theoretical optimal solutions, are shown in Table VII. The symbols "+," "-", and "=" were employed to respectively denote OHBA's superiority, inferiority, or equivalency to the comparison algorithm. The "NaN" label was used to signify cases where no conclusive determination could be made.

The outcomes of the Wilcoxon rank-sum test are outlined in Table VIII. Within this table, OHBA, in conjunction with the other six algorithms, demonstrated p-values below the 5% significance threshold in 55 instances. There were four cases with results deemed inconclusive and two cases with p-values exceeding 5%. This comprehensive analysis strongly indicates that OHBA demonstrates significant excellence in optimization compared to other algorithms. Consequently, OHBA exhibits superior robustness in comparison to the analyzed algorithms.

TABLE VII
BENCHMARK FUNCTIONS

Benchmark functions	Dim	Range	f_{min}
$f_1(x)=\sum_{i=1}^{Dim} x_i^2$	30/50/100	[-100,100]	0
$f_2(x)=\sum_{i=1}^{Dim}\lvert x_i\rvert+\prod_{i=1}^{Dim}\lvert x_i\rvert$	30/50/100	[-10,10]	0
$f_3(x)=\sum_{i=1}^{Dim}(\sum_{j=1}^{j} x_j^2)^2$	30/50/100	[-100,100]	0
$f_4(x)=max_i\{\lvert x_i\rvert,1\le i\le Dim\}$	30/50/100	[-100,100]	0
$f_5=(\sum_{i=1}^{Dim-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30/50/100	[-30,30]	0
$f_6=\sum_{i=1}^{Dim} ix_i^4+random[0,1)$	30/50/100	[-1.28,1.28]	0
$f_7=\sum_{i=1}^{Dim}[x_i^2-10\cos(2\pi x_i)+10]$	30/50/100	[-5.12,5.12]	0
$f_8=-20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{Dim} x_i^2}\right)-\exp\left(\frac{1}{30}\sum_{i=1}^{Dim} cos2\pi x_i\right)+20+e$	30/50/100	[-32,32]	0
$f_9=\frac{1}{4000}\sum_{i=1}^{Dim} x_i^2-\prod_{i=1}^{Dim} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30/50/100	[-600,600]	0
$f_{10}=0.1\{sin^2(\pi 3x_1)+\sum_{i=1}^{Dim-1}(x_i-1)^2[1+sin^2(\pi 3x_{i+1})]+(x_n-1)^2[1+sin^2(\pi 3x_{Dim})]\}+\sum_{i=1}^{Dim} u(x_i,5,100,4)$	30/50/100	[-50,50]	0

TABLE VIII
THE VALUES OF WILCOXON SIGNED RANK TEST

Benchmark functions	HBA	WOA	SCA	BOA	MFO	DBO
F1	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
F2	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
F3	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
F4	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
F5	6.6955E-11	2.3768E-07	5.4941E-11	3.0199E-11	3.0199E-11	7.6171E-03
F6	2.9215E-09	6.7220E-10	3.0199E-11	3.3384E-11	3.0199E-11	2.3715E-10
F7	NaN	NaN	1.2118E-12	3.1349E-04	1.2118E-12	4.1926E-02
F8	3.3371E-01	1.8210E-06	1.2118E-12	1.2118E-12	1.2118E-12	1.6074E-01
F9	NaN	4.1926E-02	1.2118E-12	5.7258E-11	1.2118E-12	NaN
F10	3.0199E-11	3.3384E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.7769E-10
+ / = / -	8/2/1	9/1/0	10/0/0	10/0/0	10/0/0	8/1/1

V. Engineering Optimization Design Problems

In order to evaluate OHBA's effectiveness in real-world engineering applications, we selected two optimization problems: the pressure vessel design problem and the rolling bearing design problem. OHBA was subjected to a systematic comparison with five alternative optimization algorithms, which included WOA, SCA, DBO, Golden Jackal Optimization (GJO) [39], and Aquila Optimizer (AO) [40], in addition to the conventional HBA algorithm. These comparative experiments were meticulously carried out to provide a comprehensive assessment of OHBA's performance and its suitability for engineering applications in the realm of academic optimization.

A. Pressure vessel problem

The primary objective of the pressure vessel optimization design problem is the minimization of economic costs while adhering to specific constraints[41]. These economic costs encompass material expenses, forming costs, and welding expenditures associated with the vessel's fabrication. A visual representation of the pressure vessel design problem is provided in Fig.12. Derived from Fig.12, this problem entails four critical constraints that must be satisfied, and it relies on four variables for the computation of the objective function. These variables include shell thick-ness (z_1), head thickness (z_2), inner radius (x_3), and the vessel's length, excluding the 138 head (x_4).

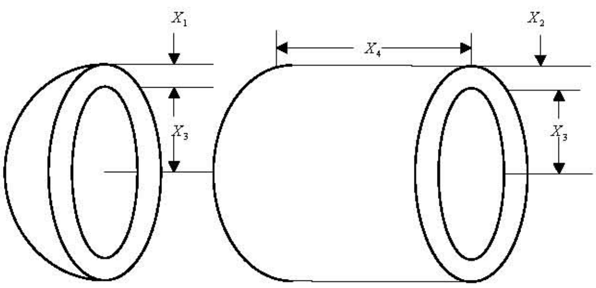


Fig.12. The model of pressure vessel problem

Minimize:

$$f(X) = 1.7781 z_2 x_3^2 + 0.6224 z_1 x_3 x_4 + 3.1661 z_1^2 x_4 + 19.84 z_1^2 x_3$$

Subject to:

$$g_1(X) = 0.00954 x_3 - z_2 \le 0$$

$$g_2(X) = 0.0193 x_3 - z_1 \le 0$$

$$g_3(X) = X_4 - 240 \le 0$$

$$g_4(X) = 1296000 - \pi x_3^2 x_4 - 4/3 \pi x_3^3 \le 0$$

Where, $z_1 = 0.0625 x_1, z_2 = 0.0625 x_2$

With bounds: $x_1 \le 99, \quad 1 \le x_2, \quad x_3 \le 200, \quad 10 \le x_4$

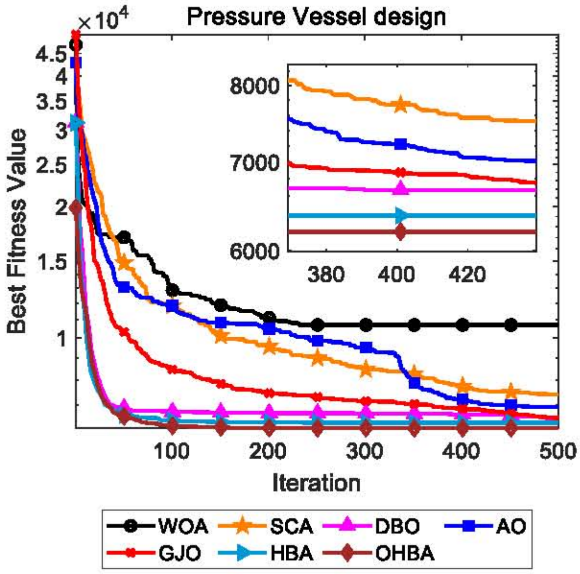


Fig.13. Comparison of convergence curves for pressure vessel problem

The experimental outcomes for OHBA and the comparative algorithms concerning the pressure vessel design problem are depicted in Fig.13. Each algorithm underwent 30 independent runs to address the pressure vessel design problem, and the ensuing results encompass means, standard deviations, optimal values, and worst values, all of which are compiled in Table IX. For the sake of clarity, the most favorable experimental data extracted from the table have been highlighted.

In the context of pressure vessel problems, the practicality and stability of the OHBA are demonstrated from multiple perspectives. The OHBA exhibits significantly lower standard deviation compared to other algorithms, indicating its strong robustness. Furthermore, the average and best values of the OHBA outperform those of other comparative algorithms, and the optimization results further illustrate the high efficiency of the OHBA in addressing complex real-world engineering problems.

Table IX
Comparison Results Of Six Algorithms Under The Pressure Vessel problem

	WOA	SCA	DBO	AO	GJO	HBA	OHBA
$f(x)$	11142.346	8038.696	6820.41	6240.676	7400.952	6059.714	6059.7154
x_1	16.22131	16.7732	18.3813	12.94245	20.3539	13.37172	12.5013
x_2	24.93488	11.9765	8.72841	6.904851	9.65965	6.963087	7.480618
x_3	47.8123	53.1361	58.2902	40.69833	63.6565	42.09845	42.09844
x_4	116.7081	79.6451	43.6927	194.9216	16.9702	176.6366	176.6367
Ave	9721.2	7551.5	6619.7	6848.5	6531.6	6340.1	**6285.1**
Std	2480.7	580.1611	536.228	529.9373	566.3045	452.5891	**330.9291**
Best	6351.6	6269.8	6059.7	6149	6064.7	6059.7	**6059.7**
Worst	7560.9	7680.5	7544.5	8715.9	17611	7544.5	**7332.8**

B. Rolling element bearing

The rolling bearing issue involves transforming sliding contact into rolling contact, thereby reducing friction losses, with the optimization objective of maximizing dynamic load capacity [42]. This optimization is formulated considering five design variables and five design parameters. Among the design variables are the pitch diameter (D_m)), ball diameter (D_b), outer raceway curvature coefficient (f_o)), inner raceway curvature coefficient (f_i), and the total number of balls (Z). Additionally, the five design parameters, namely e, $\in$, ζ, K_{Dmax}, K_{Dmin}, play a role primarily within the constraints of the problem. The problem encompasses nine non-linear constraints rooted in manufacturing and kinematic considerations, aiming to solve for the values of ten design variables and dynamic load capacity. The problem encompasses nine non-linear constraints rooted in manufacturing and kinematic considerations. A visual representation of the rolling bearing design problem can be found in Fig.14.

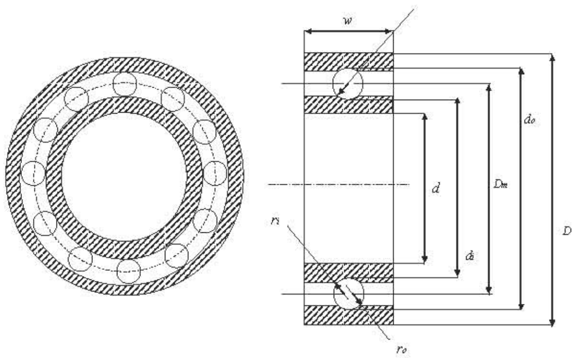


Fig.14. The model of rolling element bearing

Minimize:

$$f(X)=\begin{cases} f_c Z^{\frac{2}{3}} D_b^{1.8}, & if D_b \le 25.4mm \\ 3.647 f_c Z^{\frac{2}{3}} D_b^{1.4}, & otherwise \end{cases}$$

Subject to:

$$g_1(X)=Z-\frac{\phi_0}{2sin^{-1}(D_b/D_m)}-1\le 0$$

$$g_2(X)=K_{Dmin}(D-d)-2D_b\le 0$$

$$g_3(X)=2D_b-K_{Dmax}(D-d)\le 0$$

$$g_4(X)=D_{b-w}\le 0$$

$$g_5(X)=0.5(D+d)-D_m\le 0$$

$$g_6(X)=D_m-(0.5+e)(D+d)\le 0$$

$$g_7(X)=D_b-0.5(D-D_m-D_b)\le 0$$

$$g_8(X)=0.515-f_i\le 0$$

$$g_9(X)=0.515-f_0\le 0$$

With bounds:

$$0.5(D+d)\le D_m\le 0.6(D+d)$$

$$0.15(D-d)\le D_b\le 0.45(D-d)$$

$$4\le Z\le 50$$

$$0.515\le f_i\le 0.6$$

$$0.515\le f_0\le 0.6$$

$$0.4\le K_{Dmin}\le 0.5$$

$$0.6\le K_{Dmax}\le 0.7$$

$$0.3\le \quad \le 0.4$$

$$0.02\le e\le 0.1$$

$$0.06\le \zeta\le 0.85$$

Where,

$$f_c=37.91\left\{1+\left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_0-1)}{f_0(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right\}^{-0.3}$$

$$\gamma=\frac{D_b\cos(\alpha)}{D_b}\ ,\quad f_i=\frac{r_i}{D_b}\ ,\quad f_0=\frac{r_0}{D_b}$$

$$\phi_0=2\pi-2cos^{-1}\times$$

$$\left(\frac{\{(D-d)/2-3(T/4)\}^2+\{D/2-(T/4)-D_b\}^2-\{d/2+(T/4)\}^2}{2\{(D-d)/2-3(T/4)\}\{D/2-(T/4)-D_b\}}\right)$$

$$T=D-d-2D_b\ ,\quad D=160\ ,\quad d=90\ ,\quad B_w=30$$

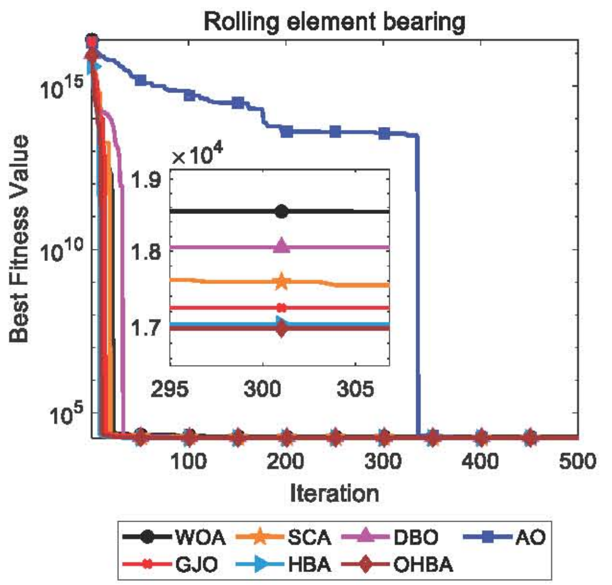


Fig.15. Comparison of convergence curves for Rolling element bearing

The OHBA algorithm underwent a rigorous systematic evaluation through a comparative analysis with prominent algorithms such as WOA, SCA, DBO, AO, GJO, and HBA. Each algorithm was executed independently in 30 instances to address the challenges associated with rolling bearing design problems. The findings from these experiments are depicted in Fig.15. The resulting statistics, including means, standard deviations, optimal values, and worst values, are detailed in Table X. To facilitate a clearer interpretation, the most favorable experimental data have been highlighted in bold within the table.

When addressing the rolling bearing design problems, the OHBA achieved the optimal optimization results. According to experimental findings, OHBA consistently exhibited the lowest averages, standard deviations, optimal values, and worst values compared to other comparative algorithms. In all aspects, OHBA has obtained the highest ranking, demonstrating the best overall performance and emphasizing the effectiveness of this method when applied to practical problem-solving scenarios.

TABLE X
COMPARISON RESULTS OF SIX ALGORITHMS UNDER THE ROLLING ELEMENT BEARING

	WOA	SCA	DBO	AO	GJO	HBA	OHBA
$f(x)$	17672.758	17513.146	17058.767	17045.251	17045.56	16958.202	16958.202
x_1	129.9129	125.2312	125	127.7706	125.9213	131.2	131.2
x_2	18	18.17614	18	18.01677	18.00068	18	18
x_3	4.51	4.893674	4.51	5.219708	5.164455	4.51	4.51
x_4	0.6	0.6	0.6	0.6	0.6	0.6	0.6
x_5	0.5402832	0.587077	0.6	0.6	0.6	0.6	0.6
x_6	0.4855557	0.5	0.4	0.4	0.494259	0.5	0.4
x_7	0.7	0.6	0.7	0.6	0.658191	0.6	0.7
x_8	0.3330697	0.3739817	0.300028	0.3	0.327157	0.3	0.3
x_9	0.02	0.02	0.0288869	0.1	0.051192	0.1	0.0997939
x_{10}	0.6	0.6	0.6	0.6	0.6	0.6	0.6
Ave	19049	17507	17515	17540	17046	17262	**16974**
Std	2433.3	315.3726	1601.4	1453.7	31.3332	1565.2	**21.3759**
Best	16976	17103	16958	17045	16977	16958	**16958**
Worst	24559	18300	25749	23741	17096	25548	**17018**

VI. CONCLUSIONS

In this study, we have introduced an Improved Honey Badger Algorithm Based on a Hybrid Strategy. This paper outlines four key enhancement strategies built upon the foundation of the standard Honey Badger Algorithm. Initially, the 'good point set' method is integrated for population initialization, introducing a randomized population distribution. Subsequently, the conventional Uniform random number generation is replaced with the Beta distribution to achieve a balance between global exploration and local exploitation capabilities. In the following step, the adaptive density factor is refined, effectively facilitating a seamless transition between global and local search phases. Lastly, the Cauchy mutation is introduced, leveraging the Sine chaotic mapping, to enhance the algorithm's diversity, thereby augmenting its search capabilities and ultimately improving convergence accuracy. Performance evaluation involves a thorough assessment across the CEC-2017 and CEC-2022 benchmark functions, simultaneously benchmarking OHBA against five other prominent metaheuristic algorithms. To determine OHBA's statistical significance, we employ the Wilcoxon rank-sum test for comparison with other optimization algorithms. Finally, we apply the OHBA algorithm to address two real-world engineering problems. The obtained results unequivocally affirm the superiority of the OHBA algorithm over other competing algorithms. Therefore, it can be confidently concluded that OHBA, the newly developed algorithm, surpasses state-of-the-art metaheuristic algorithms, demonstrating its excellence, effectiveness, and promising potential for practical applications. In the future, we will continue to enhance OHBA's performance and extend its application to a broader spectrum of practical engineering challenges.

REFERENCES

[1] Talbi E G. Machine learning into metaheuristics: A survey and taxonomy[J]. ACM Computing Surveys (CSUR), vol.54, no.6, pp.1-32, 2021.

[2] Lu C, Gao L, and Yi J. Grey wolf optimizer with cellular topological structure[J]. Expert Systems with Applications, vol.107, pp. 89-114, 2018.

[3] Abualigah L, Diabat A, Mirjalili S, et al. The arithmetic optimization algorithm[J]. Computer methods in applied mechanics and engineering, vol.376, pp. 113609, 2021.

[4] Sattar D, and Salim R. A smart metaheuristic algorithm for solving engineering problems[J]. Engineering with Computers, vol.37, no.3, pp. 2389-2417, 2021,

[5] Blum C. Ant colony optimization: Introduction and recent trends[J]. Physics of Life reviews, vol.2, no.4, pp.353-373, 2005.

[6] Gad A G. Particle swarm optimization algorithm and its applications: a systematic review[J]. Archives of computational methods in engineering, vol.29, no.5, pp. 2531-2561, 2022.

[7] Trojovský P, Dehghani M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications[J]. Sensors, vol.22, no.3, pp.855, 2022.

[8] Mirjalili S, Mirjalili S M, and Lewis A. Grey wolf optimizer[J]. Advances in engineering software, vol.69, pp.46-61, 2014.

[9] Krentel M W. The complexity of optimization problems[C]// Proceedings of the eighteenth annual ACM symposium on Theory of computing, pp.69-76,1986.

[10] Osaba E, Villar-Rodriguez E, Del Ser J, et al. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems[J]. Swarm and Evolutionary Com putation, vol.64, pp.100888, 2021.

[11] Tang J, Liu G, and Pan Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends[J]. IEEE/CAA Journal of Automatica Sinica, vol.8, no.10, pp. 1627-1643, 2021.

[12] Du N, Zhou Y, Deng W, et al. Improved chimp optimization algorithm for three-dimensional path planning problem[J]. Multimedia Tools and Applications, vol.81, no.19, pp. 27397-27422, 2022.

[13] Brand M, Masuda M, Wehner N, et al. Ant colony optimization algorithm for robot path planning[C]//2010 international conference on computer design and applications. IEEE, vol.3, pp. V3-436-V3-440, 2010.

[14] Levenhagen J, Bortfeldt A, and Gehring H. Path tracing in genetic algorithms applied to the multiconstrained knapsack problem[C]// Workshops on Applications of Evolutionary Computation. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.40-49, 2001.

[15] Wang R, Li Y, Fan J, et al. A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller[J]. IEEE Access, vol.8, pp.166525-166540, 2020.

[16] Yang J, and Zhuang Y. An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem[J]. Applied soft computing, vol.10, no. 2, pp. 653-660, 2010.

[17] Xu X, Rong H, Trovati M, et al. CS-PSO: chaotic particle swarm optimization algorithm for solving combinatorial optimization problems[J]. Soft Computing, vol.22, pp.783-795, 2018.

[18] Sharawi M, Zawbaa H M, and Emary E. Feature selection approach based on whale optimization algorithm[C]//2017 Ninth international conference on advanced computational intelligence (ICACI). IEEE: pp.163-168, 2017.
[19] Hu G, Du B, Wang X, et al. An enhanced black widow optimization algorithm for feature selection[J]. Knowledge-based systems, vol.23 5, pp.107638, 2022.
[20] Hashim F A, Houssein E H, Hussain K, et al. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems[J]. Mathematics and Computers in Simulation, vol.192, pp.84-110, 2022.
[21] Düzenli T, Onay F K, and Aydemir S B. Improved honey badger algorithms for parameter extraction in photovoltaic models[J]. Optik, vol.268, pp.169731, 2022.
[22] Lei W, He Q, Yang L, et al. Solar photovoltaic cell parameter identification based on improved honey badger algorithm[J]. Sustainability, vol.14, no.14, pp.8897, 2022.
[23] Ma J, Yu S, and Cheng W. Composite fault diagnosis of rolling bearing based on chaotic honey badger algorithm optimizing VMD and ELM[J]. Machines, vol.10, no.6, pp. 469, 2022.
[24] Papasani A, and Devarakonda N. A novel feature selection algorithm using multi-objective improved honey badger algorithm (mo-hba) and spea-ii[J]. Journal of Engineering Research, 2022.
[25] Khan M H, Ulasyar A, Khattak A, et al. Optimal Sizing and Allocation of Distributed Generation in the Radial Power Distribution System Using Honey Badger Algorithm[J]. Energies, vol.15, no.16, pp. 5891, 2022.
[26] Gonggui Chen, Zhijin Jiang, Hao Tan, Chaohua Zhang, Jinzhu Wei, Wenchen Chen, and Hongyu Long, "Sliding-Mode Design with Disturbance Observer Optimized by Improved HBA Algorithm for Frequency Regulation of Multi-Area Power System," IAENG International Journal of Applied Mathematics, vol. 52, no.4, pp994-1013, 2022.
[27] Xiang Haiyun, LI Hongxin, FU Xiao, and SU Xiaoping. Improved honey badger algorithm based on multiple strategies and its application[J]. Computer Engineering, doi: 10.19678/j.issn.pp.1000-3428.00 66465.
[28] Chen R F, Luo H, Huang K C, et al. An improved honey badger algorithm for electric vehicle charge orderly planning[J]. J. Netw. Intell, vol.7, no.2, pp.332-346, 2022
[29] Liu S H, Mernik M, Hrnčič D, et al. A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model[J]. Applied Soft Computing, vol.13, no.9, pp.3792-3805, 2013.
[30] Arora S, and Singh S. An improved butterfly optimization algorithm with chaos[J]. Journal of Intelligent & Fuzzy Systems, vol.32, no.1, pp.1079-1088, 2017.
[31] Fan J, Li Y, and Wang T. An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism[J]. PLoS One, vol.16, no.11, pp.e0260725, 2021.
[32] Nagra A A, Han F, and Ling Q H. An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search[J]. Engineering Optimization, vol.51, no.7, pp.1115-1132, 2019.
[33] Ding H, Wu Z, and Zhao L. Whale optimization algorithm based on nonlinear convergence factor and chaotic inertial weight[J]. Concurrency and Computation: Practice and Experience, vol.32, no.24, pp. e5949, 2020.
[34] Mirjalili S, and Lewis A. The whale optimization algorithm[J]. Advances in engineering software, vol.95, pp.51-67, 2016.
[35] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-based systems, vol.96, pp.120-133, 2016.
[36] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing,vol.23, pp.715-734, 2019.
[37] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-based systems, vol.89, pp.228-249, 2015.
[38] Xue J, Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing, vol.79, no.7, pp.7305-7336, 2023.
[39] Chopra N, and Ansari M M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications[J]. Expert Systems with Applications, vol.198, pp.116924, 2022.
[40] Abualigah L, Yousri D, Abd Elaziz M, et al. Aquila optimizer: a novel meta-heuristic optimization algorithm[J]. Computers & Industrial Engineering, vol.157, no.107250, 2021.
[41] He, Xiaoyu, and Yuren Zhou. "Enhancing the performance of differential evolution with covariance matrix self-adaptation." Applied Soft Computing, vol.64, pp.227-243, 2018.
[42] Rao R V, Savsani V J, and Vakharia D P. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. Computer-aided design, vol.43, no.3, pp.303-315, 2011.