Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(9): 9388–9411.
DOI: 10.3934/mbe.2022437
Received: 11 May 2022
Revised: 10 June 2022
Accepted: 19 June 2022
Published: 27 June 2022

Research article

Improved bat algorithm for roundness error evaluation problem

Guowen Li[1], Ying Xu[1,2], Chengbin Chang[1], Sainan Wang[1], Qian Zhang[1] and Dong An[1,*]

[1] School of Mechanical Engineering, Shenyang Jianzhu University, Shenyang 110168, China
[2] School of Electro-mechanical Engineering, Guangdong University of Technology, Guangdong 510006, China

* **Correspondence:** Email: andong@sjzu.edu.cn; Tel: +8602424690088.

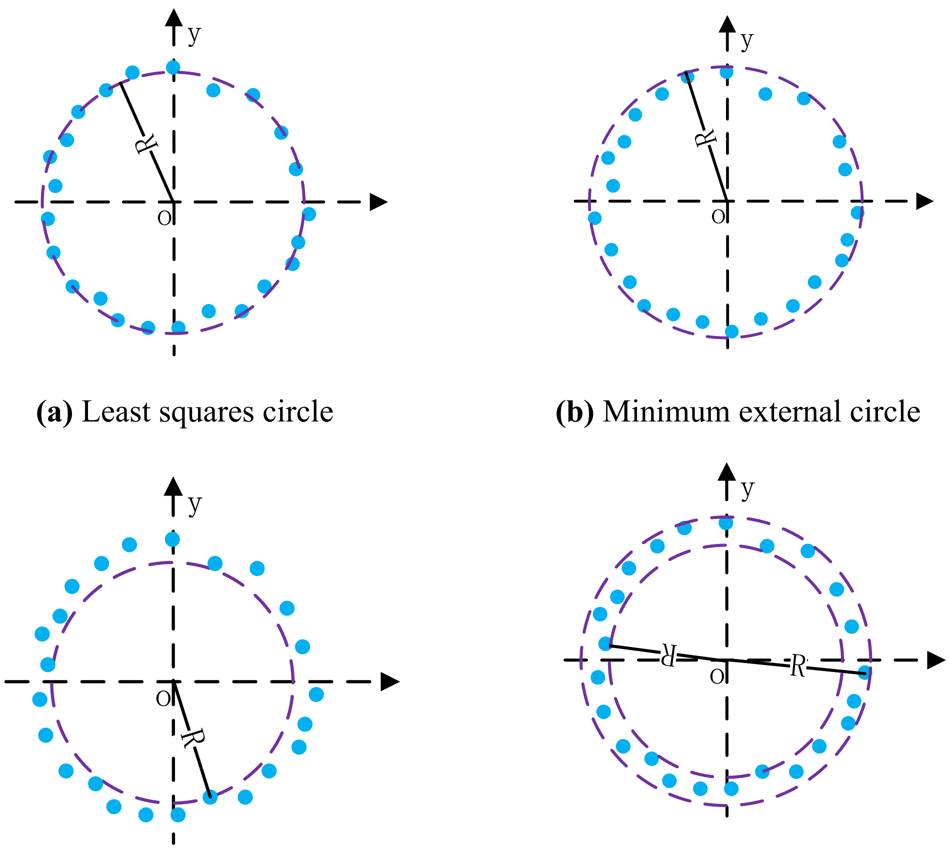
Abstract: In the production and processing of precision shaft-hole class parts, the wear of cutting tools, machine chatter, and insufficient lubrication can lead to changes in their roundness, which in turn affects the overall performance of the relevant products. To improve the accuracy of roundness error assessments, Bat algorithm (BA) is applied to roundness error assessments. An improved bat algorithm (IBA) is proposed to counteract the original lack of variational mechanisms, which can easily lead BA to fall into local extremes and induce premature convergence. First, logistic chaos initialisation is applied to the initial solution generation to enhance the variation mechanism of the population and improve the solution quality; second, a sinusoidal control factor is added to BA to control the nonlinear inertia weights during the iterative process, and the balance between the global search and local search of the algorithm is dynamically adjusted to improve the optimization-seeking accuracy and stability of the algorithm. Finally, the sparrow search algorithm (SSA) is integrated into BA, exploiting the ability of explorer bats to perform a large range search, so that the algorithm can jump out of local extremes and the convergence speed of the algorithm can be improved. The performance of IBA was tested against the classical metaheuristic algorithm on eight benchmark functions, and the results showed that IBA significantly outperformed the other algorithms in terms of solution accuracy, convergence speed, and stability. Simulation and example verification show that IBA can quickly find the centre of a minimum inclusion region when there are many or few sampling points, and the obtained roundness error value is more accurate than that of other algorithms, which verifies the feasibility and effectiveness of IBA in evaluating roundness errors.

Keywords: bat algorithm; roundness error; convergence speed; solution accuracy

1. Introduction

The roundness error is a fundamental geometric element in determining the quality of a machine part. High-precision shaft-hole class parts in various fields, such as in the aerospace, automotive, and military industries need to be produced under strict dimensional and shape tolerances in order to meet standards. An accurate evaluation of the dimensional and geometric characteristics of shaft-hole class parts ensures a high level of performance during use. Roundness is a fundamental geometric characteristic of shaft-hole class parts that can vary according to the actual manufacturing environment owing to inadequate lubrication, wear of cutting tools, machine chatter, and irregular spindle rotation. According to the standards of the American Society of Mechanical Engineers and the International Organization for Standardization, the roundness error is the minimum radius difference between the inner circle and the outer circle. Commonly used roundness error assessment methods are the least squares method (LSM), the minimum circumscribed circle method (MCC), the maximum internal circle method (MIC), and the minimum zone circle method (MZC) [1], as shown in Figure 1(a)–(d), respectively. The LSM has the advantages of simplicity and a theory that is easy to understand, simple programming, and simple algorithms; therefore, it is widely used. But the LSM does not meet the minimum bar defined by the international standard piece. MCC is often used to evaluate the roundness error of the outer surface, MIC is often used to evaluate the roundness error of the inner surface. In the roundness error evaluation, for a certain contour, sometimes three-point or two-point contact is satisfied. The inscribed circle or circumcircle of the form is not unique, which makes it possible to determine. MCC and MIC are very difficult and extremely unsuitable in application. MZC is an optimal evaluation method, and is commonly used internationally for evaluating roundness errors.

Many scholars have published different solutions for the roundness error assessment accuracy problem in recent years. Luo [2] proposed the use of information entropy to initialise the population, introduced a new search strategy, and proposed an improved cuckoo algorithm to assess roundness errors. Cai [3] introduced a roulette selection method based on the traditional cuckoo algorithm, optimised the step scaling factor and re-nesting probability in the traditional cuckoo search algorithm Lévy flight, and evaluated the roundness error using the improved cuckoo search algorithm. Li [4] introduced augmented search and interactive learning mechanisms to enhance the learning efficiency of the algorithm and maintain the diversity of the population while keeping the typical process of the basic Drosophila optimisation algorithm. Wang [5] designed a variable step size method based on the standard Drosophila whisker algorithm to further improve the computational accuracy and convergence speed of the algorithm. Srinivasu [6] combined LSM and the novel probabilistic global search lausanne (PGSL) technique. LSM was initially used to reduce the search space and within this reduced search space, PGSL performs an efficient, refined, and global search. Zhang [1] applied an improved genetic algorithm to the roundness error assessment. Liu [7] first proposed the intersecting chord method for implementing a minimum area model of the roundness error with coordinate data, which achieved satisfactory results in terms of computational accuracy and evaluation efficiency. Cao [8] found that in many industrial scenarios, a fine evaluation of local segment roundness variations is usually more practical than global evaluation; hence, a roundness error assessment method based on local least squares circle statistical analysis is preferable. Yue [9] proposed an asymptotic search method to obtain the concentric centre coordinates of the minimum area circle model and calculate the circularity error, and Yue [10] proposed a circularity error assessment method based on the minimum zone circle method. Samuel [11] proposed a method for processing coordinate measurement data and shape data based on computational geometry techniques. A new heuristic algorithm was proposed to solve the inner hull using the computational geometry concept of a convex hull. Gadelmawla [12] proposed a simple and effective algorithm to calculate the roundness error of a large number of points obtained by CMMs using three internationally defined methods: MCC, MIC, and MZC. Ray [13] proposed a method to evaluate the roundness error using geometric relations, but this method suffers from a large number of iterations, slow convergence, and the need to determine the direction and step size of the next circle centre movement. Because there is no consensus on the best method for roundness error assessments, the traditional geometric method, which is necessary to determine the direction in which the centre of the circle moves. The optimisation algorithm can solve the circularity error quickly, but the accuracy of the optimisation algorithm in evaluating the circularity error is not high enough. In order to improve the accuracy of the evaluation of the circularity error. In this paper, we apply the traditional BA to the roundness error evaluation. Due to the problems of the traditional bat algorithm, we propose to IBA to improve the evaluation accuracy of the roundness error.



(a) Least squares circle

(b) Minimum external circle

(c) Maximum inner joint circle

(d) Minimum zone circle

Figure 1. Common methods of roundness error assessment. **(a)** Least squares circle; **(b)** Minimum external circle; **(c)** Maximum inner joint circle; **(d)** Minimum zone circle.

In this study, the circularity error assessment accuracy problem is transformed into an optimisation problem of the circle centre of the minimum inclusion region, and the BA is applied to the roundness error assessment for this problem. An IBA is proposed because the traditional BA lacks a variational mechanism and is prone to fall into local extrema and premature convergence. Chaos initialisation is applied to the initial solution generation to enhance the variational mechanism of the population and improve the quality of the solution. During the iterative process, sinusoidal control factor is added to the BA to control the nonlinear inertia weights, which is used to dynamically adjust the balance between the global and local searches of the algorithm and improve the algorithm's search accuracy and stability. SSA is integrated into the BA to utilise the large range search capability of the explorer bat and make the algorithm jump out of local extremes and then quickly converge to a global optimum. The feasibility and effectiveness of the algorithm were verified the accuracy of IBA in roundness error evaluation.

2. Roundness error evaluation problem proposed

Roundness error assessment is a nonlinear optimisation problem, and the roundness error is the radius difference between two concentric circles that are inclusive of the actual circular contour, in other words, is the minimum radius difference, so the roundness error problem can be assessed by the BA. The discriminatory criterion of roundness error assessments using MZC is that when the measured contour is included by two concentric circles, the minimum radius difference between the two concentric circles is the roundness error. The coordinates of the sampling points of the circle section are $P_i(x_i, y_i)$ $(i = 1, 2, \ldots, n)$, n is the number of measurement points, the centre of the circle that meets the minimum zone circle is $O(a,b)$, and the distance between each sampling point $P_i(x_i, y_i)$ to the real-time circle centre $O_m(a_m, b_m)$ is R_m, as shown in Figure 2.

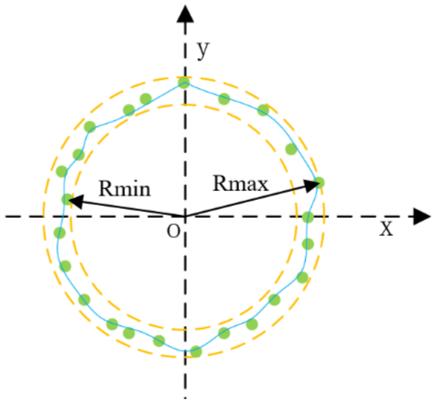


Figure 2. Schematic diagram of roundness error evaluation.

$$R_m = \sqrt{(x_i - a_m)^2 - (y_i - b_m)^2} \tag{1}$$

From Eq (1), it is known that the roundness error problem can be transformed into solving the concentric circle centre (a,b) that satisfies the minimum zone circle with the concentric circle radius difference f. T is the maximum number of iterations, and the objective function of the IBA optimisation is

$$f(a_m, b_m) = \min_{1 \le m \le T} \left(\max_{1 \le i \le n} R_m - \min_{1 \le i \le n} R_m \right) \tag{2}$$

3. Bat algorithm

The BA is a meta-heuristic search algorithm proposed by Xin-she Yang, a British scholar in 2010 [14], which includes a swarm intelligence optimisation algorithm that evolved from the behaviour of bats based on echolocation for searching and hunting. With good global search capability and fast convergence, the BA simulates a series of behavioural characteristics of bats, such as searching for locations and hunting using ultrasound. In the BA, an N bat individual updates the frequency, velocity, and position values during its flight. The entire algorithm consists of two parts: global and local updates. During the iterations, each bat can vary its pulse emission rate r$_i$, its loudness A$_i$, and its frequency f$_i$. Frequency variations or tuning can be carried out by. The global update equation is as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min}) * rand \tag{3}$$

$$v_i^t = v_i^{t-1} - (x_i^t - x_*) * f_i \tag{4}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{5}$$

where f_i , $f_{\max}$, and $f_{\min}$ denote the frequency, maximum frequency, and minimum frequency of i bats, respectively; $rand$ denotes a random number within the range of [0, 1], v_i^t denotes the velocity of the i^{th} bat at time t, x_* denotes the global optimum, and x_i^t denotes the position $(i = 1, 2, \cdots, N)$ of the i^{th} bat at time t. In the local search phase, if the following specific conditions are satisfied, the position update equation is:

$$x_{new} = x_{old} + \xi A^t \tag{6}$$

where x_{old} is the current optimum, ξ is a random number between [−1, 1], and A^t is the current average loudness of all bats $(t = 1, 2, \cdots, T)$.

As the number of iterations increases and approaches the target value, the loudness gradually decreases, while the pulse emissivity gradually increases. The loudness and pulse emissivity are updated as follows:

$$A_i^{t-1} = \alpha A_i^t \tag{7}$$

$$r_i^{t+1} = r_i^0 \left[1 - \exp(-\gamma t)\right] \tag{8}$$

where α is a random number between [−1, 1], and γ is a random number greater than 0.

The steps of a traditional BA are as follows:

1) Initialization of algorithm parameters: the number of bats is n, the frequency search range is $[f_{\min}, f_{\max}]$; the maximum pulse firing rate is r^0, the pulse rate coefficient is γ; the initial loudness is A , the loudness wave coefficient parameter is α , and the maximum number of iterations is T .
2) v_i^t and x_i^t are updated according to Eqs (4) and (5).
3) Start the local search and update the position according to Eq (6) when $rand > r_i$.

4) When $rand < A_i$ and $f_{new} < f(x_i)$, A_i and r_i are updated according to Eqs (7) and (8).
5) Determine if $f_{new} < f(x_*)$, and if yes, update the global optimal solution.
6) If the maximum number of iterations is reached, stop, and output the position of the best individual; otherwise, go back to Step (2).

In addition to the above advantages, traditional BA also has disadvantages, such as a lack of variation mechanisms, and the fact that it is not easy to maintain population diversity. An important point for an efficient swarm intelligence algorithm is to have an excellent mutation mechanism for maintaining population diversity and thus the ability to sustain evolution. From the optimisation process of the basic BA, the algorithm lacks an effective variation mechanism [15] and also lacks mobility, which in turn affects the algorithm's global search and local search imbalance [16]. Bat individuals are prone to be attracted to local extremes, leading to premature convergence. The basic algorithm uses learning from the current optimal individual for velocity updates, thus achieving a position update; if the current optimal individual, once attracted by local extremes, does not have an effective mechanism for eliminating the binding, the population rapidly loses diversity and thus the ability to evolve. This is particularly obvious in high-dimensional complex morphology performance. The pulse frequency r and pulse tone intensity A only determine whether the algorithm accepts the updated position with a certain probability, but it is not effective in overcoming the attraction of local attractors. Due to the above problems, an IBA is proposed.

4. Improved bat algorithm

4.1. Chaos initialization

In the initialisation process of intelligent optimisation algorithms, different generation methods produce different initial solution sequences. The basic BA generates the initial population randomly, therefore, the population lacks a variation mechanism, which has the problem of an uneven distribution of the original individuals, which in turn affects the optimisation performance of the algorithm. By adding chaotic initialisation to the traditional BA, the introduced chaotic sequence has the advantages of ergodicity, randomness, and regularity, which can be obtained by determining Eqs (9) and (10). The initial solution sequence generated by logistic chaotic mapping is proposed as:

$$x_{n+1} = \mu x_n (1 - x_n) \tag{9}$$

$$x_n^* = x_{\min} + x_n (x_{\max} - x_{\min}) \tag{10}$$

where μ is 4, indicating that the system is completely chaotic; $x_{\max}$ and $x_{\min}$ represent the upper and lower bounds of the variables in the solution space, respectively; x_n is the initialisation variable generated by chaos in (0, 1]; x_n^* is the variable mapped by x_n to a range of values between $x_{\max}$ and $x_{\min}$. Adding chaos initialisation makes the initial solution of the population more uniformly distributed and improves the quality of the initial solution.

4.2. Sinusoidal control factor

It can be seen from Eq (4) that the update rate of the BA is 1, which reduces the flexibility of individual bats in the bat algorithm to some extent, reduces the diversity of the population, and does not consider the experience accumulated in the early stages of the bat search, thus greatly reducing the mobility of the algorithm, which in turn affects the imbalance between the global search and the local search of the algorithm. Seyedali Mirjalili created the Sine Cosine Algorithm [17], which integrates multiple random variables and adaptive variables, emphasizing the exploration and utilization of the search space in different optimization stages. Inspired by the Sine Cosine Algorithm, the sinusoidal control factor is added to the BA to improve the search accuracy and stability of the algorithm. The expression of the sinusoidal control factor is:

$$\omega\text{=}\sin(\frac{\pi}{2}-\frac{\pi\times t}{2\times T}) \tag{11}$$

The speed update formula reads:

$$v_i^t=\omega\times v_i^{t-1}+(x_i^t-x_*)f_i \tag{12}$$

where t is the current iteration number, T is the maximum iteration number, ω is a random number between [0, 1], and ω is the sinusoidal control factor. The larger the value of ω is in the first period, the larger is the search range of the algorithm, which is conducive to finding the global optimal value. Moreover, with an increasing number of iterations, the value of ω gradually becomes smaller; at this time, to allow the bat to conduct a fine search. By changing the value of ω, the balance between the global and local searches of the algorithm can be dynamically adjusted to improve the accuracy of the algorithm.

4.3. Fusion SSA explorer ability

Inspired by the foraging strategy of sparrows in nature, the SSA was proposed by Xue et al., in 2020 [18]. The SSA algorithm has the advantages of higher stability, better convergence accuracy, and avoiding falling into the local optimum to some extent compared with other algorithms. In the process of foraging, the sparrow as explorer provides the search direction and area for the population, and the sparrow as follower is guided by the explorer. The search is conducted by the explorer. To address the problem that the local search ability and search accuracy are not high enough and easily fall into local extremes, leading to early convergence in the bat search algorithm, the ability of the explorer sparrow in the SSA is integrated into BA. Because the searcher sparrow has a larger search range compared with other algorithms and can quickly update its position, the ability of the explorer can be given to part of the bat search algorithm for guiding the entire population to achieve the goal of rapid convergence and jump out of local extremes. The process for fusing Explorer abilities in SSA is as follows.

The proportion of bats within the population that acquire explorer bat capabilities is first determined based on the proportionality factor a:

$$a=\frac{X_{best}}{X_{N-best}}=\frac{P}{N-P} \tag{13}$$

X_{best} represents the best positioned bat in the bat population, P, as the explorer bat. X_{N-best} represents $N-P$ bats with bad positions in the bat population, as follower bats, and usually a

is set to 20%. In a wide-range search environment, the warning value r_2 for becoming an explorer bat is constantly smaller than the safety value ST, when the explorer bat performs a wide-range jump search, and the explorer bat's ability is updated according to Eq (14).

$$x_{i,j}^{t+1} = x_{i,j}^{t} \times \exp(\frac{-i}{\alpha \times T}) \tag{14}$$

where $x_{i,j}^{t+1}$ denotes the position of the i th bat in the $t+1$ th iteration in the j th dimension, α is a random number between (0, 1] and T is the maximum number of iterations. All other bats follow the explorer bat at normal speed. According to this change in the explorer bat's ability, an influence factor rr is generated, which in turn affects the influence of the bat's past position and speed on its present position and speed.

$$rr = \frac{1}{\alpha \times T} \tag{15}$$

where α is a random number between (0, 1] and T is the maximum number of iterations. The velocities and positions of the follower bats are updated using Eqs (16) and (17).

$$v_i^t = rr \times \omega \times v_i^{t-1} - (x_i^t - x_*) * f_i \tag{16}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{17}$$

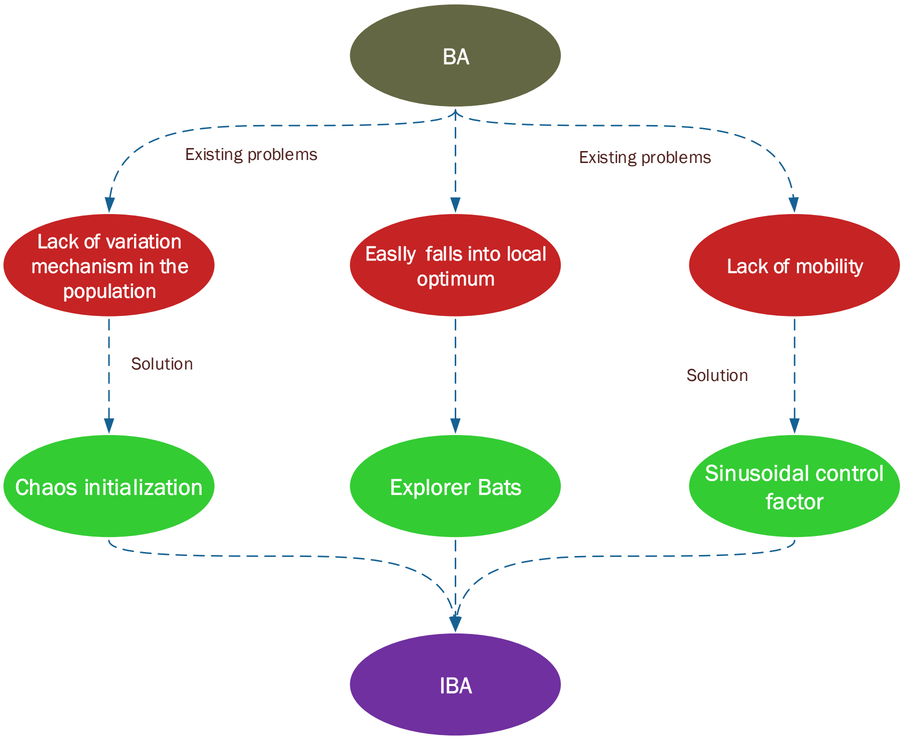


Figure 3. Diagram of improvement idea.

By fusing the SSA into the BA, the fused BA gains the ability of explorer bats to perform a wide range search, making the jump out of local extremes easier, and resulting in faster convergence, and higher accuracy. To clarify the IBA, a diagram of the improved idea is shown in Figure 3.

4.4. IBA algorithm steps

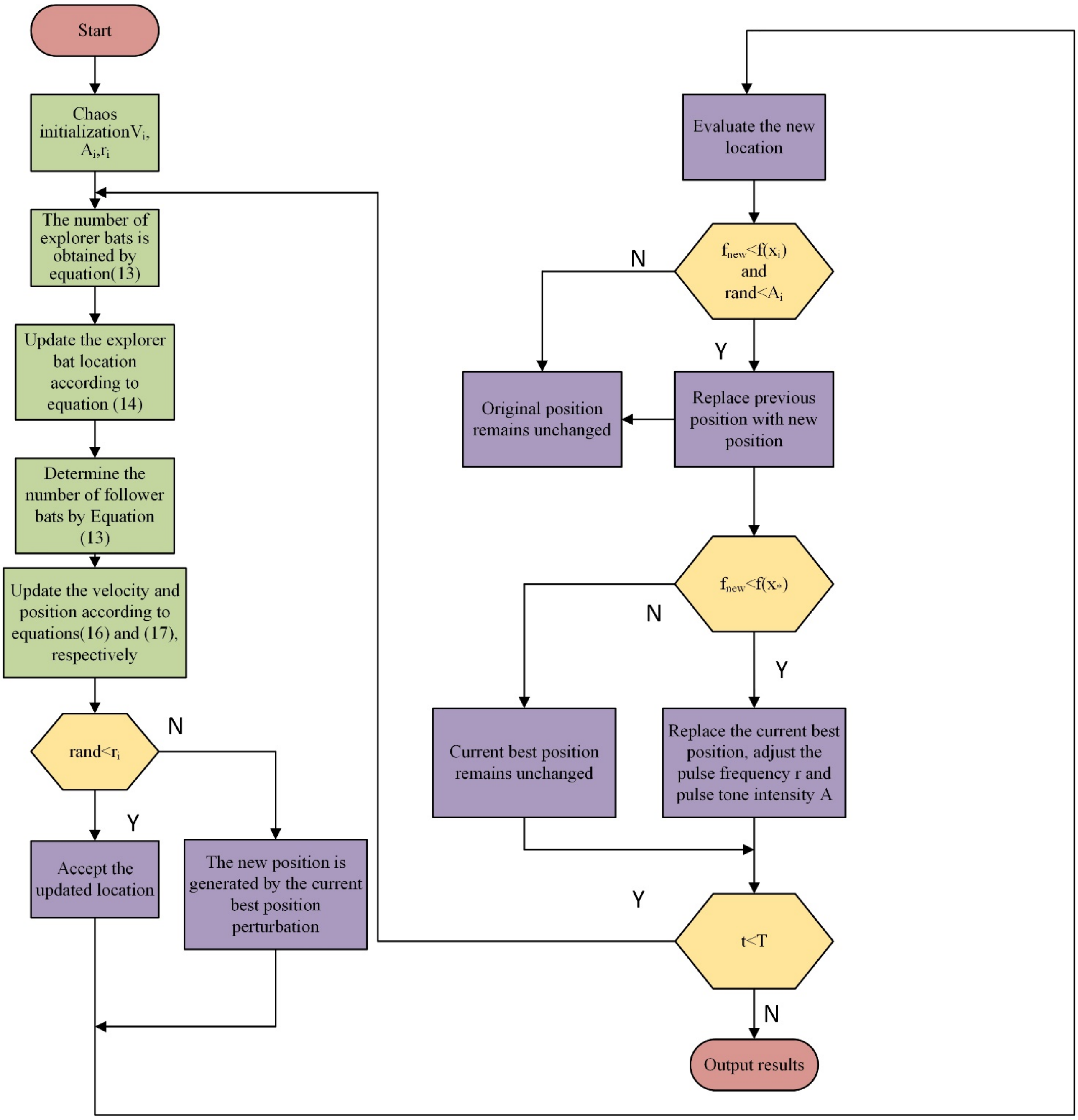

Start
Chaos initializationV$_i$, A$_i$,r$_i$
The number of explorer bats is obtained by equation(13)
Update the explorer bat location according to equation (14)
Determine the number of follower bats by Equation (13)
Update the velocity and position according to equations(16) and (17), respectively
rand<r$_i$
N
Y
Accept the updated location
The new position is generated by the current best position perturbation
Evaluate the new location
f$_{new}$<f(x$_i$) and rand<A$_i$
N
Y
Original position remains unchanged
Replace previous position with new position
f$_{new}$<f(x*)
N
Y
Current best position remains unchanged
Replace the current best position, adjust the pulse frequency r and pulse tone intensity A
t<T
Y
N
Output results

Figure 4. IBA program flow chart.

1) Initialization of algorithm parameters: the number of bats is n, the frequency search range is $[f_{\min}, f_{\max}]$; the maximum pulse firing rate is r^0, the pulse rate coefficient is γ; the initial loudness is A, the loudness wave coefficient parameter is α, and the maximum number of iterations is T.

2) Individual bat positions are assigned according to chaotic initialisation (9) and (10), and

according to the fitness function. The value of the fitness function is found for each bat, and the position of the bat for the optimal value x_* is recorded.

3) The population was divided into explorer bats and follower bats, and position updates were performed using Eq (14) when explorer bats performed a wide range search, while follower bats performed position and velocity updates using Eqs (3), (16), and (17).
4) When $rand < A_i$ and $f_{new} < f(x_i)$, A_i and r_i are updated according to Eqs (7) and (8).
5) Determine if $f_{new} < f(x_*)$, and if yes, update the global optimal solution.
6) If the maximum number of iterations is reached, stop, and output the position of the best individual; otherwise, go back to Step (2).

The flow chart of IBA is shown in Figure 4.

4.5. Algorithm complexity

Because the algorithm works with the same producer for each iteration, we therefore analyse the algorithm complexity of one iteration of the algorithm. The complexity of the proposed IBA algorithm in the worst case is as follows:

The complexity of the traditional BA algorithm is $O(N)$, where N is the population size, using chaos initialization to make the population uniformly distributed. The complexity of chaos initialization is $O(N)$, and the inertia weight is changed by the sinusoidal control factor, whereby the complexity of the sinusoidal control factor is $O(1)$. The explorer ability of the SSA is integrated into BA to change the position of the bat and the complexity is $O(D)$, where D is the dimensionality. Therefore, the total complexity of the IBA algorithm is $O(2N+D+1)$. From this equation we can see that chaos initialization, the sinusoidal control factor and explorer bat ability all increase the complexity of the IBA algorithm.

5. Performance testing with roundness error simulation and example verification

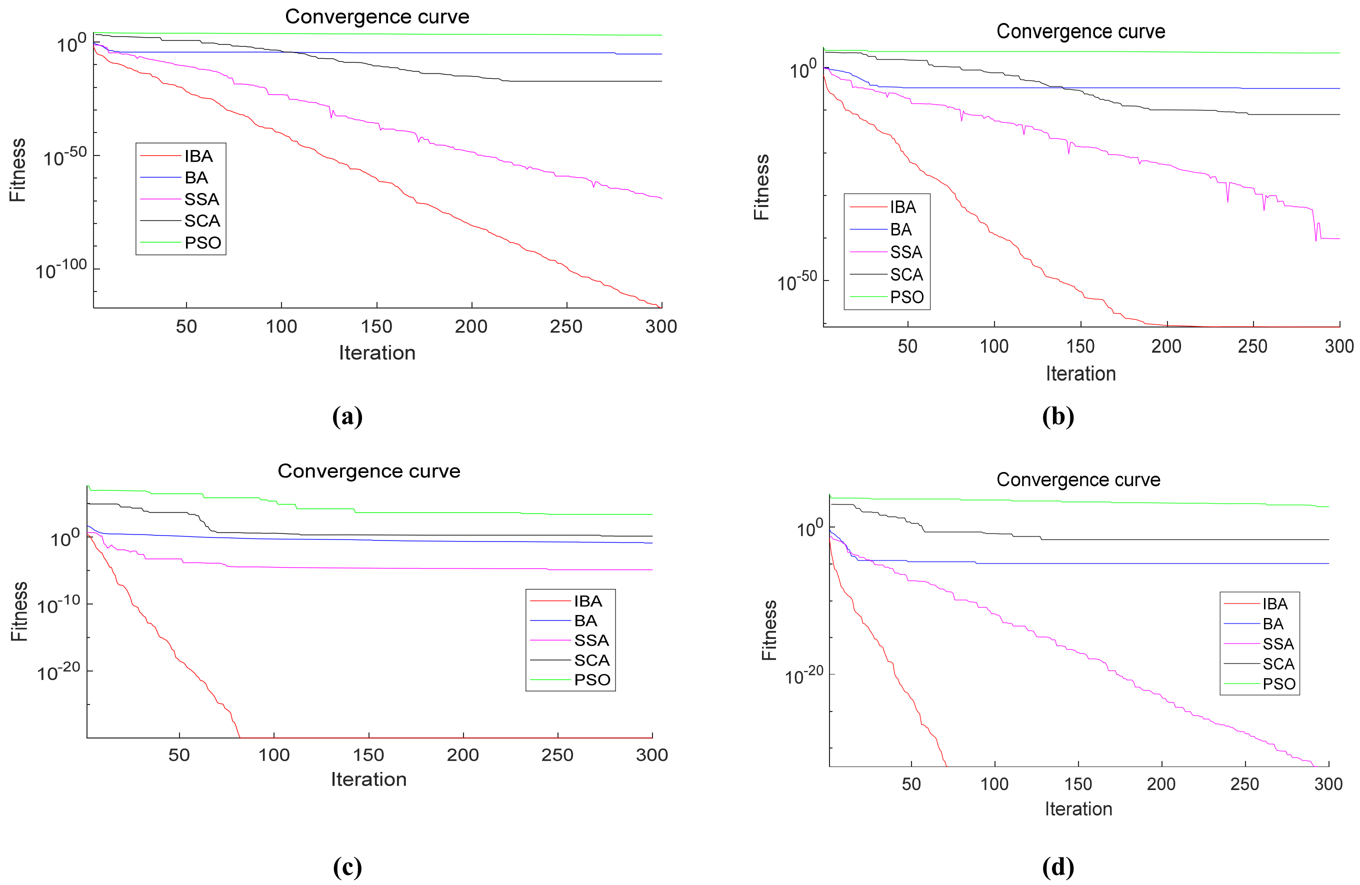
5.1. Performance Test

5.1.1. IBA vs. BA/PSO/SCA/SSA for performance comparison on F1–F4

To verify the performance of the IBA algorithm, four test functions were selected for performance testing in this study, and this experiment was performed using MATLAB 2018b software. The experimental platform was a PC with an operating system of win10 64-bit, a processor of Intel(R) core(TM) i5-7500CPU@3.40 GHz, and a 16.0 GB memory. The population size was set to 30, the maximum number of iterations was 300, the maximum value of loudness A was 0.25, and the maximum value of pulse emissivity r^0 was 0.5. Each group of experiments was calculated 20 times, and the averages and standard deviations of the optimal solutions were recorded and saved. The number of iterations for each group of experiments was 300, and the IBA was compared with PSO, SCA, SSA, and the traditional BA. Four groups of test functions were included: F1 and F2 for single-peak benchmark functions; F3 and F4 for multi-peak benchmark functions. The results obtained are exhibited in Table 1, and the optimal values are shown in bold. Algorithm settings: PSO was set to $c1=c2=1$, $\omega=1$; SSA was set to a = 20%, ST = 1.5.

Table 1. Test function results (dim = 5).

Algorithm	Function	Avg	Std	Algorithm	Function	Avg	Std
IBA	F1	**1.44E-70**	**6.42E-70**	IBA	F3	**9.88E-16**	**4.42E-15**
BA		1.35E-05	3.53E-06	BA		1.35E+00	1.83E+00
SSA		2.07E-62	8.61E-62	SSA		4.08E-04	1.19E-03
SCA		1.03E-17	2.99E-17	SCA		2.26E+00	4.35E-01
PSO		8.69E+03	1.40E+04	PSO		1.84E+04	1.48E+04
IBA	F2	**3.15E-73**	**1.41E-72**	IBA	F4	**0**	**0**
BA		1.07E-05	4.11E-06	BA		9.77E-06	4.55E-06
SSA		1.90E-37	5.87E-37	SSA		1.06E-32	3.85E-32
SCA		5.33E-10	2.28E-09	SCA		2.58E-02	1.55E-02
PSO		6.69E+03	4.67E+03	PSO		4.57E+03	3.39E+03



(a)

(b)

(c)

(d)

Figure 5. F1–F4 performance test curve.

In Figure 5(a), the convergence of BA and IBA tends to decrease when the number of iterations increases. After 20 iterations, BA falls into a local optimum, converges early, and cannot jump out. The IBA generates influence factors due to the explorer bats for a large range search and enhances the ability of the BA to jump out of local extremes by influencing the past positions and velocities of bats and therefore, the updates of the present positions and velocities. After 50 iterations, the convergence speed of IBA is obviously faster than that of BA, and it can be seen that the integration of the ability of explorer sparrows into SSA can improve the ability of BA to jump out of local extremes.

In Figure 5(b), BA converges to a local optimum approximately 40 times and falls into a local extremum, from which it cannot escape again. The IBA performs a wide-range search with explorer bats to generate an influence factor, which enhances the ability of the BA to jump out of a local extremum by influencing the past positions and velocities of the bats and therefore, the updates of the present positions and velocities. After jumping out of the local optimum by changing the sinusoidal control factor and thus, dynamically adjusting the inertia weights, IBA quickly converges to a more accurate solution.

In Figure 5(c), both BA and IBA show a decreasing trend as the number of iterations increases, but IBA decreases faster than BA. BA quickly falls into a local optimum and is then unable to jump out of the local optimum. IBA quickly jumps out of the local extreme with the sinusoidal control factor and the fused SSA explorer sparrow capability, and the algorithm converges to the global optimum at 80 iterations.

In Figure 5(d), when performing the search for the best value, the local optimum has a great impact on the convergence speed of the traditional BA because of a lack of variation mechanisms in the bat population once the population has been initialised. Therefore, chaos initialisation is added to enhance the random uniform distribution of the population. In subsequent iterations, the explorer bat performs a large search, generating an influence factor by influencing the present position and velocity updates through the bats' past positions and velocities. So, after the algorithm jumps out of local extremes, the sinusoidal control factor dynamically adjusts the inertia weights to regulate the balance between global and local searches, so that the convergence of the IBA has a fast decreasing trend.

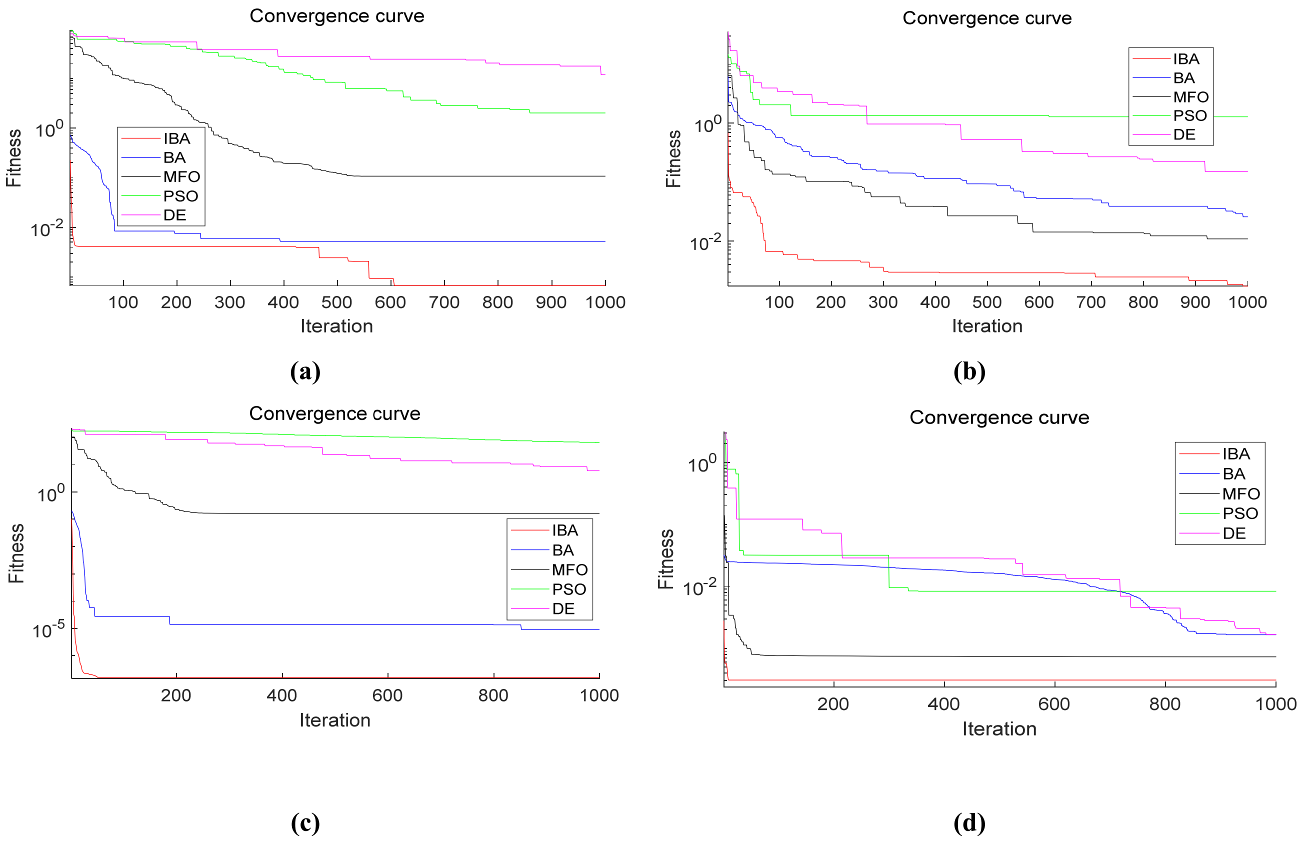
5.1.2. IBA vs BA/PSO/MFO/DE for performance comparison on F5–F8

Since the performance test 5.1.1 selected SSA, SCA, PSO and BA, however, the SSA and SCA algorithms still have low optimization accuracy.in order to further compare the performance of IBA and other algorithms, a performance test using IBA with traditional BA was used, and compared with the performance of the moth flame optimisation (MFO), the differential evolutionary algorithm (DE), and the PSO algorithm using four sets of benchmark functions in high dimensions, where F5 is the multi-peaked benchmark function, F6 is the fixed dimensional benchmark function, and F7 and F8 are the composite benchmark functions. The population size was set to 30, the maximum number of iterations was 1000, each group of experiments was calculated 20 times, and the averages and standard deviations of the optimal solutions were recorded and saved. The obtained results are exhibited in Table 2, and the optimal values are shown in bold. Algorithm settings: IBA, BA, and PSO with the same settings as described in the previous Section 4.1.1. DE: F = 0.5, CR = 0.9; MFO: a = 1, A = 1.

In Figure 6(a)–(d), the BA is caught in a local optimum and cannot jump out of the local optimum. IBA goes through population chaos initialisation to make the population distribution uniform and incorporates the ability of the explorer sparrow into SSA to make the algorithm jump out of local extremes quickly using the explorer bat to conduct a large-range search first and then, a global search using the follower bats. This ensures that the algorithm can jump out of local extremes through a sinusoidal control factor, which is used to control the non-inertial weight value change, and then dynamically adjust the global search and local search. Thus, IBA converges quickly and reaches the global optimum.

Table 2. Test function results (dim = 10).

Algorithm	Function	Avg	Std	Algorithm	Function	Avg	Std
IBA		**2.34E-03**	**9.14E-04**	IBA		**7.32E-07**	**8.01E-07**
BA		5.16E-02	7.60E-03	BA		1.03E-05	2.48E-06
SFO	F5	3.03E+00	3.85E+00	SFO	F7	1.81E-01	1.45E-01
PSO		2.88E+00	9.80E-01	PSO		7.07E+02	1.29E+02
DE		2.04E+02	6.36E+00	DE		6.13E+00	2.76E+00
IBA		**8.08E-04**	**4.13E-04**	IBA		**3.45E-04**	**1.44E-04**
BA		2.37E-02	9.91E-03	BA		7.65E-04	6.00E-04
SFO	F6	7.77E-03	6.48E-03	SFO	F8	1.06E-03	4.30E-04
PSO		4.06E+00	2.92E+00	PSO		2.66E-02	2.63E-02
DE		1.03E-01	6.46E-02	DE		2.45E-03	3.50E-03



(a)

(b)

(c)

(d)

Figure 6. F5–F8 performance test curve.

5.1.3. Performance test analysis

It is obvious from Tables 1 and 2 that IBA has a better optimisation effect on a single-peaked function, multi-peaked function, fixed-dimensional multi-peaked function, and composite benchmark test function. A good population initialisation can make the algorithm influence the process of finding the global optimum from the beginning of the algorithm. Chaos initialisation is introduced into BA to uniformly distribute the BA population and strengthen the optimisation performance of the algorithm. SSA explorer sparrows are fused into BA to enhance the exploration ability of the bats. Explorer bats are then made to perform a large range of fast searches, generating the influence factor rr , and influence the present position and velocity updates to make the algorithm jump out of local optimums through the past positions and velocities of the bats using rr . The sinusoidal control factor is also added into BA to enhance bat manoeuvrability, and make the algorithm converge to the global optimum quickly through changes of ω with the goal of obtaining a more accurate solution of IBA.

Table 3 shows the results of the statistical analysis of the algorithm's Wilkerson rank test. The p-value is the value of Wilkerson's test between IBA and other algorithms at a confidence level of 0.05. According to the results in Table 3, the calculated differences are all lower than 0.05, indicating that there is a significant difference between IBA and the other algorithms for the eight test functions; therefore, this conclusion is statistically significant.

Table 3. P-values of the Wilcoxon rank test.

	Function	BA	SSA	SCA	PSO
Performance Test 4.1.1	F1	7.4E-05	8.9E-05	8.9E-05	8.9E-05
	F2	8.8E-05	8.9E-05	8.9E-05	8.9E-05
	F3	8.9E-05	8.8E-05	8.9E-05	8.9E-05
	F4	8.8E-05	2.7E-02	8.9E-05	8.9E-05
	Function	BA	SFO	PSO	DE
Performance Test 4.1.2	F5	8.9E-05	8.9E-05	8.9E-05	8.9E-05
	F6	8.9E-05	1.0E-04	8.9E-05	8.9E-05
	F7	8.9E-05	8.9E-05	8.9E-05	8.9E-05
	F8	8.9E-05	8.3E-05	8.9E-05	8.9E-05

5.2. Roundness error simulation verification

To verify the feasibility of this method for roundness error evaluation, two specified size circles were simulated [19]: the first specified circle had a radius of 1 mm, a roundness error of 10 μm, an inner diameter size of 0.997 mm, and an outer diameter size of 1.007 mm. The centre of the concentric circle was defined as the coordinate origin, and other measurement points of the circle contour were simulated. The simulated data points are shown in Table 4; the second specified circle has a radius of 1 mm and a roundness error of 10 μm. The inner diameter was 0.990 m, and the outer diameter was 1.000 mm. The data points of the simulation are presented in Table 5. Algorithm settings: The set population size was 30, the maximum number of iterations was 50, the maximum value of loudness A was 0.25, and the maximum value of pulse emissivity r^0 was 0.5. The parameters used in the BA were the same as those described above.

Table 4. Simulation 1 data.

No.	x/mm	y/mm	No.	x/mm	y/mm	No.	x/mm	y/mm
1	0.704985	0.704985	5	0.868623	-0.5015	9	0.5	-0.866025
2	-0.5035	-0.872086	6	1.004	0	10	-0.711349	-0.711349
3	-0.997	0	7	-0.868623	-0.5015			
4	0	1.007	8	0.5	0.866025			

Table 5. Simulation 2 data.

No.	x/mm	y/mm	No.	x/mm	y/mm	No.	x/mm	y/mm
1	1	0	18	-0.56929	0.81612	35	-0.344039	-0.934865
2	0.986291	0.127168	19	-0.670278	0.73805	36	-0.221349	-0.969793
3	0.963874	0.252757	20	-0.760615	0.647521	37	-0.0954	-0.988926
4	0.924223	0.374176	21	-0.837747	0.545313	38	0.032052	-0.999486
5	0.869181	0.489514	22	-0.891959	0.429545	39	0.15933	-0.985512
6	0.795612	0.59378	23	-0.944759	0.313681	40	0.283348	-0.954692
7	0.716048	0.693454	24	-0.973104	0.189512	41	0.402961	-0.910295
8	0.621339	0.779135	25	-0.989456	0.063525	42	0.517963	-0.854435
9	0.514052	0.847982	26	-0.990536	-0.063595	43	0.619037	-0.776248
10	0.401217	0.906357	27	-0.979996	-0.190854	44	0.716605	-0.693993
11	0.2831	0.953859	28	-0.941978	-0.312758	45	0.79944	-0.596638
12	0.159536	0.986784	29	-0.899296	-0.433078	46	0.86592	-0.487677
13	0.03184	0.992893	30	-0.831748	-0.541408	47	0.917648	-0.371514
14	-0.095625	0.991251	31	-0.760907	-0.64777	48	0.963114	-0.252558
15	-0.220794	0.967361	32	-0.667931	-0.735466	49	0.982624	-0.126695
16	-0.344506	0.936134	33	-0.56752	-0.813583	50	0.99054	-0.024261
17	-0.459093	0.879995	34	-0.459074	-0.879959			

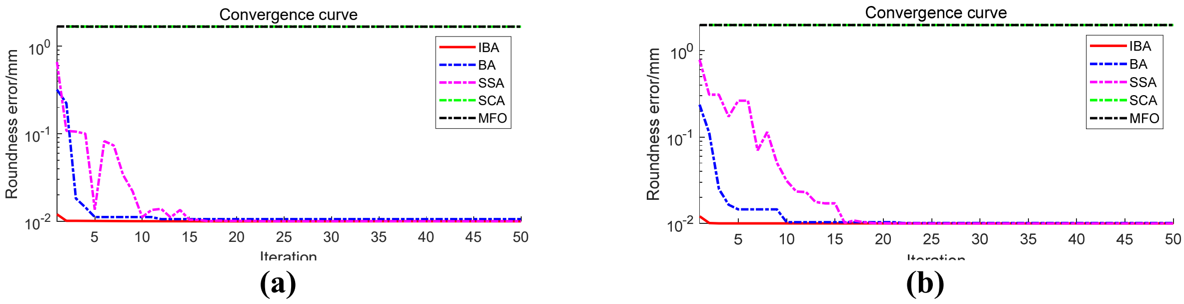
Table 6. Simulation 1 processing results.

Method	Centre coordinates		Roundness error/mm
	x/mm	y/mm	
BA	2.881e-04	-4.301e-04	0.01033
IBA	-7.714e-07	9.407e-07	**0.01000**
MFO	-150	150	1.6798
SSA	-7.6363e-07	9.1728e-07	0.0099998
SCA	-47.8307	47.8573	1.6777

Table 7. Simulation 2 processing results.

Method	Centre coordinates		Roundness error/mm
	x/mm	y/mm	
BA	6.2030e-04	1.2203e-03	0.01138
IBA	2.0579e-07	3.3623e-09	**0.01000**
MFO	149.9997	-68.4027	1.98
SSA	2.0348e-07	-1.1253e-09	0.0099997
SCA	76.3035	-34.731	1.98

From the results in Tables 6 and 7, it can be seen that the result obtained by the IBA algorithm is 10 μm, while the results obtained by other algorithms are all greater than 10um, which proves that the accuracy of IBA is higher than other algorithms in the roundness error evaluation. From Figure 7(a),(b), it can be seen that IBA, after the initialization of population chaos, has a downward trend, and through the large range search capability of the explorer bat, the algorithm jumps out of the local optimum and continues to search before using the sinusoidal control factor to dynamically adjust the inertia weights to make the algorithm converge quickly to the global optimum solution. While BA is caught in a local optimum, it is unable to jump out of local extremes because of the lack of mobility of the population, and it is verified through simulations that IBA converges faster and has a higher search accuracy than traditional BA.



(a) **(b)**

Figure 7. Roundness error simulation verification curve.

5.3. Roundness error example verification

To further verify the feasibility and effectiveness of the algorithm in roundness error assessment, sampling point data from the literature [11,20] were used for a roundness error assessment with IBA and BA. The original data are shown in Tables 8–11, the main object of the roundness error assessment is the centre of the circle $O(a,b)$, so the search space of IBA is two-dimensional, the set population size is 30, the maximum number of iterations in Tables 7 and 9 is 50, and the maximum number of iterations in Tables 8 and 10 is 50. The maximum value of loudness A is 0.25, and the maximum value of pulse emissivity r^0 is 0.5. The parameters used for BA were the same as those described above in Section 4.2. The results of the IBA runs are compared with the literature results. The optimal value is shown in black.

Table 8. Eight data points from the literature [11].

No.	x/mm	y/mm	No.	x/mm	y/mm	No.	x/mm	y/mm
1	65.0038	30	4	22.3181	47.6818	7	40	5.0001
2	57.6809	47.6809	5	14.9978	30	8	57.6784	12.3215
3	40	55.0041	6	22.3207	12.3207			

Table 9. Hundred data points from the literature [20].

No.	x/mm	y/mm	No.	x/mm	y/mm	No.	x/mm	y/mm
1	0.57057	-0.95024	35	-0.5592	1.3352	69	-1.0411	-1.3206
2	1.56665	0.42132	36	-0.637	1.8676	70	-0.7890	-1.0823
3	1.01913	0.1475	37	1.1901	-1.3235	71	-0.0246	1.9005
4	0.9179	-0.8385	38	-0.9668	-1.1753	72	0.7139	0.799
5	-0.2496	1.7391	39	-0.7493	-1.0399	73	-0.4389	-1.1383
6	-0.0585	1.6545	40	-1.6001	0.6525	74	-0.4092	-1.6016
7	1.9574	0.0715	41	0.8048	-1.1432	75	1.3494	-0.1474
8	-0.6388	1.1281	42	-1.1019	-0.6525	76	-0.48152	-1.29893
9	-1.5951	0.8454	43	-0.7027	-0.7759	77	-0.6269	-1.2522
10	1.4209	0.5318	44	-0.7597	-0.9567	78	-1.2637	-0.9311
11	-1.0343	-0.467	45	1.5026	0.0313	79	-0.9584	-0.6098
12	0.4236	1.3945	46	-0.595	-1.2948	80	-0.8003	-1.4396
13	-1.0967	-1.3173	47	-0.1398	1.1773	81	-0.1384	1.3614
14	-1.6096	-1.0501	48	-1.4483	-0.4282	82	1.2736	0.4495
15	0.0079	-1.0619	49	-1.441	-0.5922	83	-0.6158	0.984
16	1.3099	1.0251	50	1.2789	-0.8130	84	0.1276	-1.4463
17	-0.9238	-1.2134	51	0.93272	1.04136	85	-0.7729	0.6518
18	-1.1727	-0.1253	52	-1.6604	-0.7575	86	0.4013	-1.2335
19	-1.0813	-0.2822	53	1.4369	0.0995	87	0.5666	-1.5757
20	-1.475	-0.1511	54	0.5633	-1.0538	88	-0.6014	-1.1251
21	0.29	1.0206	55	0.9646	1.2285	89	1.4585	-0.3672
22	0.261	-1.6876	56	0.462	0.9718	90	-1.3337	0.3174
23	-0.9285	-0.3927	57	1.0986	0.1462	91	-0.7503	-1.0336
24	0.0358	-1.6369	58	0.9888	0.8213	92	-1.4079	0.539
25	1.1664	0.0082	59	-0.5729	-1.8727	93	-1.7150	0.3622
26	0.12293	-1.00482	60	-1.1051	-0.6512	94	-0.3396	-1.6304
27	1.487	1.0866	61	1.4011	1.148	95	-1.5630	0.302
28	-0.3555	0.9602	62	1.4449	0.4777	96	-0.0920	-1.0383
29	-1.2198	0.1289	63	0.417	0.9303	97	0.4708	1.5051
30	0.5685	-1.7243	64	0.8847	-0.8546	98	1.6367	-0.5885
31	-0.0124	1.4314	65	-0.2688	-1.0733	99	1.7043	-0.1240
32	-0.156	1.0188	66	1.0129	-0.3987	100	-0.5162	0.9141
33	-0.9631	-0.3251	67	-1.3342	0.1675			
34	-0.3253	-1.1068	68	-0.5876	-1.4324			

Table 10. Twenty-four data points from the literature [20].

No.	x/mm	y/mm	No.	x/mm	y/mm	No.	x/mm	y/mm
1	107.5811	114.2119	9	55.7576	109.7039	17	85.6152	67.1081
2	102.2909	119.9906	10	53.4073	102.218	18	93.2669	68.7926
3	95.6848	124.2034	11	53.0774	94.3816	19	100.2245	72.4009
4	88.2128	126.5634	12	54.7849	86.7302	20	106.0093	77.6929
5	80.3826	126.9159	13	58.4107	79.7824	21	110.2199	84.3073
6	72.7251	125.2311	14	63.7075	74.0083	22	112.5676	91.7864
7	65.7612	121.6196	15	70.3176	69.8019	23	112.8977	99.6156
8	59.9721	116.3232	16	77.7899	67.4519	24	111.2129	107.2695

Table 11. Thirty-nine data points from the literature [20].

No.	x/mm	y/mm	No.	x/mm	y/mm	No.	x/mm	y/mm
1	1.0249	0.0863	14	-0.9394	0.1561	27	-0.4635	-0.9105
2	0.9991	0.2226	15	-0.2071	0.9218	28	0.4736	-0.9507
3	0.5974	0.7736	16	-0.3381	0.8782	29	0.5942	-0.8781
4	0.4731	0.8485	17	-0.4643	0.8132	30	-0.2059	-1.0269
5	0.8803	0.4794	18	-0.5771	0.7369	31	0.9950	-0.3272
6	0.8017	0.5899	19	-0.7763	0.5367	32	1.0218	-0.1921
7	0.9527	0.3551	20	-0.6838	-0.7485	33	-0.0686	-1.0512
8	0.7047	0.6884	21	-0.5795	-0.8424	34	0.0710	-1.0568
9	0.2101	0.9295	22	-0.9618	-0.017	35	0.2087	-1.0377
10	0.0708	0.9382	23	-0.9454	-0.2605	36	0.3445	-1.0078
11	-0.0683	0.9382	24	-0.9077	-0.3956	37	0.7082	-0.7982
12	-0.8432	0.4157	25	-0.8443	-0.5203	38	0.8873	-0.5832
13	-0.9022	0.2890	26	-0.7764	-0.6394	39	0.9510	-0.4578

Table 12. Comparison results for 8 data points.

Method	Centre coordinates		Roundness error/mm	SR
	x/mm	y/mm		
MCC [11]	40.0000	30.0014	0.0024147	7.4%
MIC [11]	40.0000	30.0010	0.0022945	2.5%
MZC [11]	39.9998	30.0022	0.0022430	0.2%
LSM [11]	40.0002	30.0012	0.0024657	9.3%
MFO	39.9976	29.9855	0.03324	93.3%
SCA	39.8158	30.5173	1.0304	98.8%
SSA	39.9998	30.0022	0.0022772	1.8%
BA	12.3789	41.4372	42.1202	99.8%
IBA	39.999682	30.002218	**0.002236716**	

Table 13. Comparison results for 100 data points.

Method	Centre coordinates		Roundness error/mm	SR
	x/mm	y/mm		
MIC [20]	0.00655	0.00277	0.95846	0.2%
MCC [20]	-0.00567	0.00772	0.96237	0.6%
MZC [20]	0.00535	0.00791	**0.95742**	0
MFO	-150	91.04567	3.507	72.7%
SCA	84.178	-51.4637	3.5063	72.7%
SSA	0.0053467	0.0079089	**0.9574200**	0
BA	0.0053885	0.0078137	0.9574537	0.1%
IBA	0.0053467	0.0079091	**0.9574200**	

Table 14. Comparison results of 24 data points.

Method	Centre coordinates		Roundness error/mm	SR
	x/mm	y/mm		
MIC [20]	82.9891	97.0096	0.0396	4.5%
MCC [20]	82.9907	97.0081	0.0386	1.0%
MZC [20]	82.9909	97.0084	**0.0382**	0
GA [21]	82.990941	97.008387	**0.00382309**	0
STA [22]	82.990941	97.008387	**0.00382309**	0
MFO	82.9909	97.0085	0.03833	0.3%
SCA	82.433	94.254	5.621	99.4%
SSA	82.9909	97.0084	0.03826	0.08%
BA	-16.5516	5.70016	59.4891	99.9%
IBA	82.990941	97.008387	**0.0382309**	

Table 15. Comparison results for 39 data points.

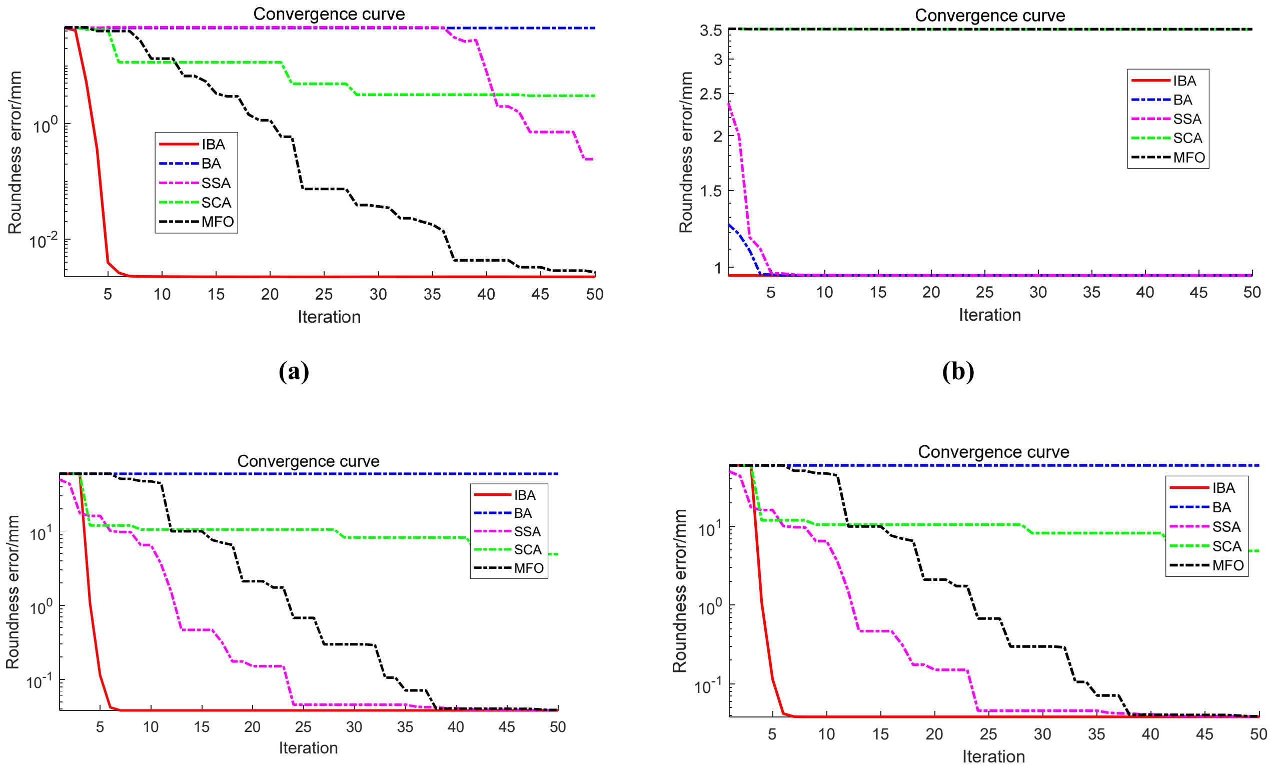
Method	Centre coordinates		Roundness error/mm	SR
	x/mm	y/mm		
LSM [2]	0.035535201	-0.053585512	0.0091672738	6.9%
GA [23]	0.035550359	-0.052929068	0.0085379971	0.01%
PSO [24]	0.035613523	-0.052928923	0.0085384689	0.02%
ABC [2]	0.035614964	-0.052929492	0.0085374723	0.01%
MZC [20]	0.0356	-0.0529	0.00856	0.27%
MZC [25]	0.035614972	-0.052929481	**0.0085374644**	0
MZC [10]	0.035614972	-0.052929481	**0.0085374644**	0
MFO	32.8326	-0.406248	1.9855	99.6%
SCA	0.03895159	-0.05118344	0.01446	40.9%
SSA	0.035614	-0.052929	0.0085385	0.01%
BA	0.035191159	-0.053098839	0.0088937156	5.1%
IBA	0.035614972	-0.052929481	**0.0085374644**	

In order to test the accuracy of the roundness error evaluation of IBA and other algorithms, the evaluation efficiency is expressed by SR:

$$SR = (1 - \frac{IBA}{\text{Contrast algorithm}}) \times 100\%$$

Among them, the Contrast algorithm is other algorithms and algorithms in the literature applied to the roundness error to be compared.

Figure 8(a),(c) show the circularity error-iteration number plots obtained for the data points in Tables 8 and 10, respectively. It can be seen that the IBA algorithm converges quickly at the very beginning of the algorithm and reaches the global optimum, whereas the BA algorithm cannot jump out after falling into a local extreme. From the results in Table 12, it can be seen that the roundness error obtained by the IBA algorithm is lower than that obtained by the literature [11] and other algorithms, and the roundness error obtained is 98.8% higher than the original algorithm. It can be seen from Table 13 that the roundness error value obtained by the IBA algorithm is lower than the roundness error obtained by the literature [20] and other algorithms, the obtained roundness error value is 0.1% higher than the original algorithm, and the circle center coordinate value is close to the actual circle center value. Figure 8(b),(d) show the circularity error-iteration number plots obtained from the data points in Tables 9 and 11, respectively, from which it can be seen that IBA first uses the explorer bat to perform a wide range search. The influence factor rr is generated in this wide-range search through the explorer bat. The past position and velocity of the bat are influenced by rr, which is used to update the present position and velocity, so that the algorithm jumps out of the local optimum. In subsequent iterations, the sinusoidal control factor is used to change the nonlinear inertia weights, and thus, the global optimum is reached quickly, while BA is trapped in the local optimum and cannot jump out of it. The results shown in Table 14 are similar to those in [20], and the roundness error values are equivalent to those obtained by MZC, and the roundness error obtained is 99.9% higher than the original algorithm. It can be seen from Table 15 that IBA yields smaller results than LSM, GA, PSO, ABC, and BA, and a higher accuracy for the roundness error, and the roundness error obtained is 5.1% higher than the original algorithm, Through the verification of the above four examples, we can see that the improved bat algorithm not only has the same results as the literature, but also has a more accurate roundness error value than the literature.it is verified that the roundness error obtained by IBA has a high accuracy when applied to the roundness error evaluation problem.



(a)

(b)

(c)

(d)

Figure 8. Example validation curve graph.

6. Conclusions

In order to improve the evaluation accuracy of the roundness errors of shaft-hole class parts, BA is applied to roundness error evaluation, and an improved bat algorithm is proposed for solving the problems of BA easily falling into local extremes, experiencing premature convergence, and the lack of a variation mechanism in the population. In addition to retaining the good characteristics of BA, chaos initialisation is applied to the population initialisation to uniformly distribute the population, add a sinusoidal control factor and fused SSA. The explorer bat enables the population to perform a large range search quickly, so that the algorithm can jump out of local extremes and obtains a great improvement in convergence speed, which in turn enhances the balance between the global and local searches of the algorithm. The excellent performance shown by IBA can be seen through the performance test, which is much higher than that of other optimisation algorithms and more stable. Although the algorithm is effective, there is still much room for improvements with regards to the optimisation effect in high-dimensional spaces. Simulation and example verifications show that IBA is an effective means for evaluating roundness errors, and that IBA is more accurate than other optimisation algorithms in evaluating roundness error values. The method proposed in this paper can not only evaluate roundness errors but can also be applied to straightness and flatness in the future.

Acknowledgments

The authors gratefully acknowledge the supports of the National Natural Science Foundation of China through Grant (No. 51975130,51975388,62173238), the Key Research and Development Project of Liaoning Province through Grant No. 2017225016.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. K. Zhang, Minimum zone evolution of circularity error based on an improved Genetic Algorithm, in *IEEE 2008 International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Thailand, (2008), 523–527. https://doi.org/10.1109/ICACTE.2008.168$
2. J. Luo, Y. Q. Lin, X. M. Liu, P. Zhang, D. Zhou, J. R, Chen, Research on roundness error evaluation based on the improved artificial bee colony algorithm, *Chin. J. Mech. Eng-En.*, **52** (2016), 27–32. https://doi.org/10.3901/JME.2016.16.027
3. Z. Cai, J. L. Wang, M. F. Lv, L. B. Zhu, Roundness error assessment based on improved cuckoo search algorithm, *Modular Mach. Tool Autom. Manuf. Tech.*, **7** (2020), 40–44. https://doi.org/10.13462/j.cnki.mmtamt.2020.07.009
4. G. Li, Y. Deng, Y. Yao, T. Zhou, Research on mechanical roundness error assessment based on improved Drosophila optimization, *Mech. Electr. Tech.*, **1** (2020), 28–32. https://doi.org/10.19508/j.cnki.1672-4801.2020.01.009
5. C. Wang, C. F. Xiang, S. H. Wang, Research on improving the accuracy of roundness error by searching the algorithm of beetle search, *Manuf. Tech. Mach. Tool.*, **11** (2019), 143–146. https://doi.org/10.19287/j.cnki.1005-2402.2019.11.029
6. D. S. Srinivasu, N. Venkaiah, Minimum zone evaluation of roundness using hybrid global search approach, *Int. J. Adv. Manuf. Tech.*, **92** (2017), 2743–2754. https://doi.org/10.1007/s00170-017-0325-y
7. F. Liu, G. H. Xu, L. Liang, Q. Zhang, D. Liu, Intersecting chord method for minimum zone evaluation of roundness deviation using Cartesian coordinate data, *Precis. Eng.*, **42** (2015), 242–252. https://doi.org/10.1016/j.precisioneng.2015.05.006

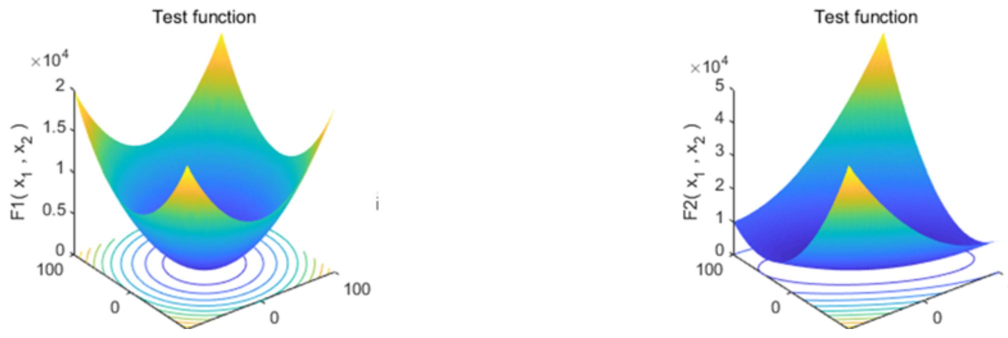
8. Z. M. Cao, Y. Wu, J. Han, Roundness deviation evaluation method based on statistical analysis of local least square circles, *Meas. Sci. Technol.*, **28** (2017), 105017. https://doi.org/10.1088/1361-6501/aa770f
9. Q. Huang, J. Mei, L. Yue, Cheng R, L. Zhang, C. Fang, et al., A simple method for estimating the roundness of minimum zone circle, *Materialwiss. Werkstofftech.*, **51** (2020), 38–46. https://doi.org/10.1002/mawe.201900012
10. L. L. Yue, Q. X. Huang, J. Mei, J. R. Cheng, L. S. Zhang, L. J. Chen, Method for roundness error evaluation based on minimum zone method, *Chin. J. Mech. Eng-En.*, **6** (2020), 42–48. https://doi.org/10.3901/JME.2020.04.042
11. G. L. Samuel, M. S. Shunmugam, Evaluation of circularity from coordinate and form data using computational geometric techniques, *Precis. Eng.*, **24** (2000), 251–263. https://doi.org/10.1016/s0141-6359(00)00039-8
12. E. S. Gadelmawla, Simple and efficient algorithms for roundness evaluation from the coordinate measurement data, *Measurement*, **43** (2010), 223–235. https://doi.org/10.1016/j.measurement.2009.10.001
13. X. Q. Lei, W. M. Pan, X. P. Tu, S. F. Wang, Minimum zone evaluation for roundness error based on Geometric Approximating Searching Algorithm, *MAPAN*, **29** (2014), 143–149. https://doi.org/10.1007/s12647-013-0078-5
14. X. S. Yang, A new metaheuristic bat-inspired algorithm, in *Nature Inspired Cooperative Strategies for Optimization (NIC-SO 2010)*, Springer, (2010), 65–74. https://doi.org/10.1016/B978-0-12-819714-1.00011-7
15. A. Kanso, N. Smaoui, Logistic chaotic maps for binary numbers generations, *Chaos Solitons Fractals*, **40** (2009), 2557–2568. https://doi.org/10.1016/j.chaos.2007.10.049
16. H. Zheng, J. Y. Yu, S. F. Wei, Bat optimization algorithm based on cosine control factor and iterative local search, *Comput. Sci.*, **47** (2020), 68–72.
17. S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowledge Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
18. J. K. Xue, Research and application of a novel swarm intelligence optimization technique, Ph.D thesis, Donghua University, 2020.
19. Q. Huang, J. Mei, L. Yue, R. Cheng, L. Zhang, C. Fang, et al., A simple method for estimating the roundness of minimum zone circle, *Materialwiss. Werkstofftech.*, **51** (2020), 38–46. https://doi.org/10.1002/mawe.201900012
20. X. M. Li, Z. Y. Shi, The relationship between the minimum zone circle and the maximum inscribed circle and the minimum circumscribed circle, *Precis. Eng.*, **33** (2009), 284–290. https://doi.org/10.1016/j.precisioneng.2008.04.005
21. X. Wen, Q. Xia, Y. Zhao, An effective genetic algorithm for circularity error unified evaluation, *Int. J. Mach. Tools Manuf.*, **46** (2006), 1770–1777. https://doi.org/10.1016/j.ijmachtools.2005.11.015
22. L. M. Zhu, H. Ding, Y. L. Xiong, A steepest descent algorithm for circularity evaluation, *Comput. Aided Des.*, **35** (2003), 255–265. https://doi.org/10.1016/S0010-4485(01)00210-X
23. W. L. Yue, Y. Wu, Fast and accurate evaluation of the roundness error based on quasi-incremental algorithm, *Chin. J. Mech. Eng-En.*, **1** (2008), 87–91. https://doi.org/10.3901/JME.2008.01.087

24. R. Andrea, A. Michele, B. L. M. Michele, Fast genetic algorithm for roundness evaluation by the minimum zone tolerance (MZT) method, *Measurement*, **44** (2011), 1243–1252. https://doi.org/10.1016/j.measurement.2011.03.031 01
25. R. Calvo, E. Gómez, Accurate evaluation of functional roundness from point coordinates, *Measurement*, **73** (2015), 211–225. https://doi.org/ 10.1016/j.measurement.2015.04.009

Appendix.

Table A1. F1–F8 performance test function table.

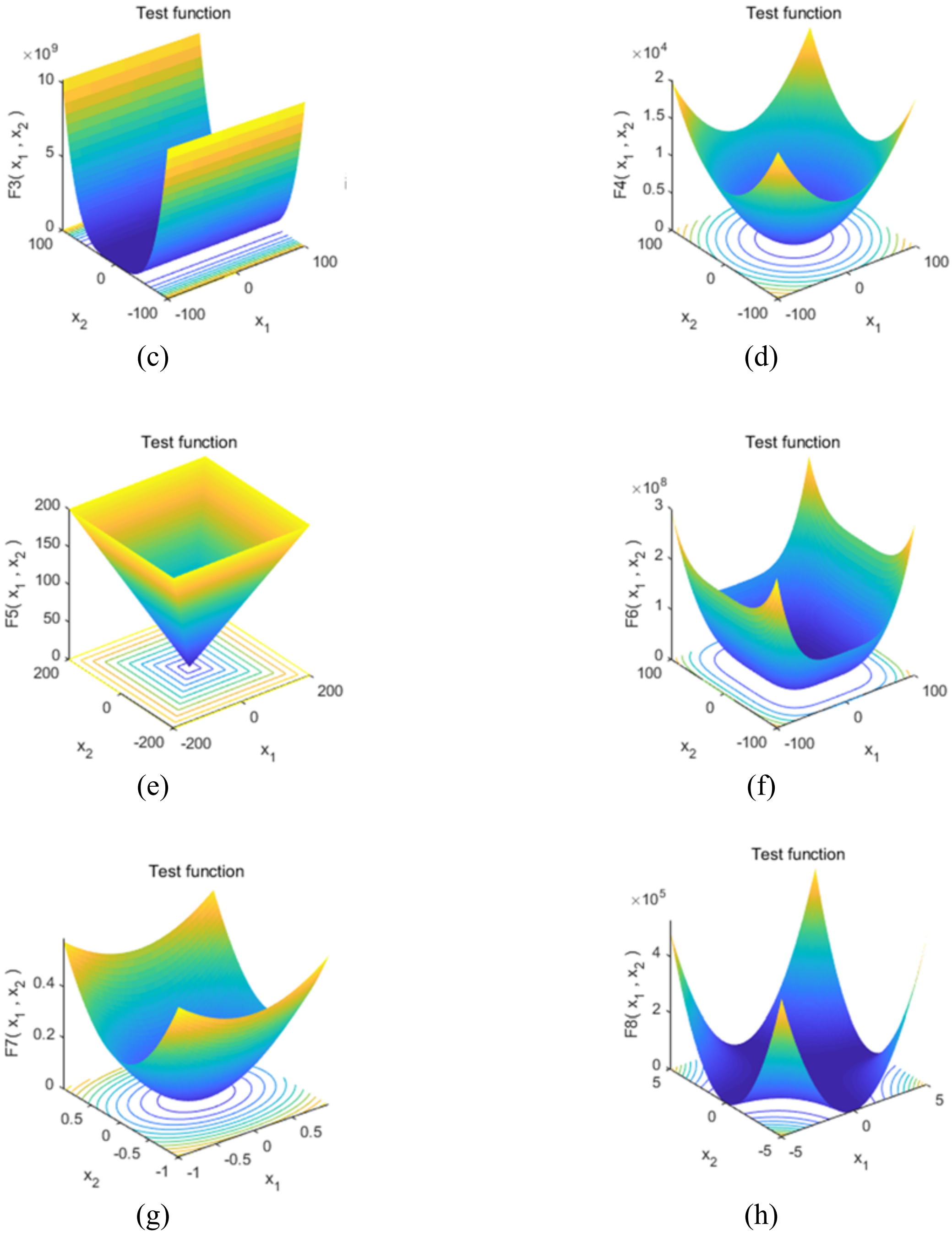
No.	Function	Scope	Min
F1	$F_1(x)=\sum_{i=1}^{n}x_i^2$	[-100,100]	0
F2	$F_2(x)=\sum_{i}^{n}\left(\sum_{j-1}^{i}x_j\right)^2$	[-10,10]	0
F3	$F_3(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	[-30,30]	0
F4	$F_4(x)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$	[-100,100]	0
F5	$F_5(x)=\max_i\left\{\lvert x_i\rvert,1\le i\le n\right\}$	[-100,100]	0
F6	$F_6(x)=\sum_{i=1}^{n}ix_i^4+random\left[0,1\right)$	[-1.28,1.28]	0
F7	$F_7(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600]	0
F8	$F_8(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_1x_2)}{b_i+b_1x_3+x_4}\right]^2$	[-5,5]	0.0003



(a)

(b)

Continued on next page



(c) (d) (e) (f) (g) (h)

Figure A1. (a–h) is the test function (F1–F8) 3D plot.