A bat optimization algorithm with moderate orientation and perturbation of trend

Jingsen Liu1,2, Hongyuan Ji2, Qingqing Liu2 and Yu Li3

Abstract
In order to improve the convergence speed and optimization accuracy of the bat algorithm, a bat optimization algorithm with moderate optimal orientation and random perturbation of trend is proposed. The algorithm introduces the nonlinear variation factor into the velocity update formula of the global search stage to maintain a high diversity of bat populations, thereby enhanced the global exploration ability of the algorithm. At the same time, in the local search stage, the position update equation is changed, and a strategy that towards optimal value modestly is used to improve the ability of the algorithm to local search for deep mining. Finally, the adaptive decreasing random perturbation is performed on each bat individual that have been updated in position at each generation, which can improve the ability of the algorithm to jump out of the local extremum, and to balance the early global search extensiveness and the later local search accuracy. The simulating results show that the improved algorithm has a faster optimization speed and higher optimization accuracy.

Keywords
Bat algorithm, nonlinear variation factor, beta distribution, random perturbation, adaptive

Introduction
In recent years, many scholars, inspired by animal habits and laws in nature, have proposed some heuristic intelligent optimization algorithms. For example, genetic algorithm for simulating the genetic mechanism of biological evolution,1 particle swarm optimization algorithm for simulating the foraging behavior of a flock of birds,2,3 ant colony algorithm for simulating the ant colony finding the shortest path during the foraging process,4 cuckoo search algorithm for simulating cuckoo parasitism breeding habits,5,6 gray wolf optimizer algorithm for simulating gray wolf level hierarchies and predation processes,7 whale optimization algorithm for simulating humpback whales predation characteristics,8 dragonfly algorithm for simulating the foraging and migration process of dragonflies9 and so on.

In 2010, Yang, a scholar from University of Cambridge, proposed a new heuristic intelligent optimization algorithm called Bat Algorithm.10 The bat algorithm is proposed by simulating bats sending and receiving ultrasonic waves to prey. At present, bat algorithms have been practically applied. For example, interaction testing,11 the fuel arrangement optimization of reactor core,12 continuous optimization problems,13 multi-objective optimization of shell and tube heat exchangers,14 linear antenna array failure correction,15 the travelling salesman problem,16 large scale cloud manufacturing service composition,17 numerical optimization problems18 electric load forecasting19 and so on.

Although the bat algorithm has been applied in many fields, the bat algorithm still has the disadvantages that it is easy to fall into local extremum, the optimization
accuracy is not high, and the convergence speed of the algorithm is slow in the later period. To address these deficiencies, many scholars have proposed improvements to the bat algorithm. For example, the reference 20 introduced simulated annealing and Gaussian perturbations into the basic bat algorithm to improve optimization performance of the algorithm. The reference 21 applied the mechanism of crossover mutation to the basic bat algorithm, and improved the update formula of loudness and pulse emission rate, so that the algorithm breadth exploration and depth mining ability are balanced. The reference 22 introduced the acceleration factor and individual cognition in the particle swarm optimization algorithm into the basic bat algorithm, and expanded the search range of the bat population, and improved the convergence accuracy of the algorithm. The reference 23 incorporated starling flock behavior into the bat algorithm and expanded the optimization range of the algorithm. At the same time, it introduced the linear decreasing weight to balance global exploration and local search ability. The reference 24 introduced the idea of interpolation into the basic bat algorithm. According to the bat’s individual historical flight trajectory, it fitted the flight curve to predict the next position of the bat, thereby improved the global optimizing ability of the bat population. The reference 25 introduced a mutation switching function to maintain a high diversity of bat populations, integrated uniform mutations and Gaussian mutations at the same time, and improved local searching ability of the algorithm. The reference 26 adopted the strategy of average optimal position to improve the convergence rate of the algorithm, and introduced quantum behavior to enhance the flexibility of bats, so that the algorithm had a better ability to adapt to complex environment. The reference 27 introduced an iterative local search algorithm in the local search, which enhanced the ability of the algorithm to jump out of the local optimum, and introduced the random inertial weight to enhance the diversity and flexibility of the bat population. The reference 28 introduced the orientation strategy into bat algorithm to improve the optimization ability of the algorithm, and used the improved algorithm to optimize the constrained design of laminated composites. The reference 29 introduced the mechanism of backward learning and tangent random exploration into bat algorithm to improve the solve ability of the algorithm, and enhanced the feasibility and effectiveness for solving robot’s global path planning problem. The reference 30 introduced the invasive weed optimization algorithm into bat algorithm, which made the iwba algorithm more effective and accurate in image segmentation. The reference 31 introduced an improved bat algorithm, and the improved MBA algorithm improved the exploration and development ability of bat algorithm. The reference 32 introduced the multi-objective bat algorithm, and used the algorithm to solve the new configuration design problem of single-mode fiber Raman amplifier, which expanded the application range of the algorithm. The reference 33 proposed chaotic efficient bat algorithm, based on the chaotic, niche search, and evolution mechanisms and used to optimize the parameters of the hybrid kernel-based support vector regression model. The reference 34 proposed adaptive coevolutionary bat algorithm. The adaptive evolutionary population structure is used to improve the convergence ability and optimization accuracy of the algorithm. The reference 35 introduced the principle of bat algorithm and used the algorithm to solve the problem of the planning of distribution center.

These improvements improved the convergence and accuracy of basic bat algorithm from different aspects, but bat algorithm still has room for improvement. This paper proposes a bat optimization algorithm with moderate optimal orientation and random perturbation of trend (OPBA). The improved algorithm introduces nonlinear variation factor in the velocity update equation of the global search stage to improve the bat population diversity. In the local search stage, a mechanism of modestly trend towards optimal value is used to improve the local search ability of the algorithm. Finally, a self-adaptive decreasing random perturbation strategy is added to reduce the probability of the algorithm falling into local extremum, effectively balancing global search and local mining capability of the algorithm. By selecting several classical objective functions with different optimization characteristics, the improved algorithm is compared with several other algorithms. The simulating results fully prove the effectiveness of the improved algorithm.

**The basic bat algorithm**

In the basic bat algorithm, each bat is viewed as a “massless, no-size” particle that represents a feasible solution in the solution space. For different fitness functions, each bat has a corresponding function value and determines the current optimal individual by comparing the function values. Then, the frequency of acoustic wave, velocity, the rate of pulse emission, and loudness of each bat in the population are updated, iterative evolution is continued, and the current optimal solution is approached and generated, and the global optimum solution is finally found. The algorithm updates the frequency, velocity, and position of each bat as follows:

\[
h_i = h_{\min} + (h_{\max} - h_{\min}) \beta
\]

\[
v'_i = v'^{t-1}_i + (x'^{t-1}_i - x_i) h_i
\]
\[ x'_i = x_i^{t-1} + v'_i \]  

(3)

Where \( h_i \) denotes the frequency of acoustic waves emitted by the \( i \)th bat. Here \( h_{\text{max}} \) and \( h_{\text{min}} \) represent the maximum and minimum frequencies respectively. \( \beta \) is a number generated randomly in the range of \([0, 1] \), \( x'_i \) and \( x_i \) denote the speed and position of the \( i \)th bat of the \( t \)th generation respectively. \( x \) represents the current optimal position.

Once the bat finds the prey, which is close to the global optimal solution, a local search strategy is used near the current optimal individual. The generated uniform distribution random number \( p \) is used as the judgment threshold. If \( p < r_i \) (the rate of pulse emission of the \( i \)th bat), a local search strategy is used, on the contrary a global search is performed. The local search position update equation is:

\[ x_{\text{new}} = x_s + \varepsilon A^t \]  

(4)

Where \( \varepsilon \in [-1, 1] \) is a random number. Here \( A^t \) is the average of all bat loudness for the \( t \)th generation. \( x_s \) is the current optimal position, \( x_{\text{new}} \) is the new position generated after a local search.

As the number of iterations increases, the loudness of the bat \( A^t \) gradually decreases while the bat is approaching the prey. At the same time, the rate of pulse emission \( r_i \) continues to increase. The updated equation is as follows:

\[ A^{t+1} = \alpha A^t \] \[ r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \]  

(5) \( (6) \)

Where \( \alpha \) and \( \gamma \) are constants. For any \( 0 < \alpha < 1 \) and \( \gamma > 0 \), we have \( A^t \to 0, r_i^t \to r_i^0 \) as \( t \to \infty \), \( r_i^0 \) is the initial rate of pulse emission.

**The improved algorithm**

**Moderately optimal orientation**

(1) In the global search stage, it can be seen from the equation (3) that the velocity \( v'_i \) is the step size of the bat individual’s updated position. Therefore, the velocity update equation (2) determines the bat population’s global exploration capability. In order to enhance the global search ability of the algorithm, a nonlinear variation factor \( L \) is added to the equation (2). The speed update equation becomes:

\[ v'_i = v_i^{t-1} + L \cdot (x_i^{t-1} - x_s) h_i \]  

(7)

In the equation (7), \( L = 2a \cdot b - a \) nonlinearly expands the search range of bat populations to ensure diversity of bat populations, where \( b \) is a random number that obeys beta distribution, and \( a \) is a convergence factoring whose equation is as follows:

\[ a = 2 \left( \frac{t}{T_{\text{max}}} \right)^2 \cdot (e - 1) \]  

(8)

Where \( t \) is the current number of iterations, \( T_{\text{max}} \) is the maximum number of iterations, and \( e \) is the natural constant. The convergence factor \( a \) decreases nonlinearly and dynamically with the increase of the number of evolutionary iterations. The decay degree of the convergence factor \( a \) is light at the beginning of the algorithm iteration, and the \( L \) value is larger, so it can move with a larger step to find the global optimal solution. At the later stage of iteration, the decay degree of \( a \) increased, the \( L \) value is smaller, so the bat’s individual movement pace decreases, which is helpful for the algorithm to find the optimal solution more accurately. At the same time, the randomness of \( b \) based on beta distribution increases the diversity of \( L \) nonlinear decline trend. As a result, the development ability and exploration ability of the algorithm global search are more effectively balanced.

(2) In the local search stage, bat individuals search for the optimal solution around the current optimal individual. During this process, on the one hand, the bat individual’s search cannot be too far away from the current optimal individual, because if the distance is too far, it becomes a global search, losing the deep mining ability of the local search. On the other hand, the distance cannot be too close, because if the distance is too close, it is easy to get the algorithm into a local optimum. Based on the above considerations, integrating some optimization ideas of the whale algorithm, in the local search stage, the following position update equation is proposed:

\[ x_{\text{new}} = x_s + D_t \cdot e^{-l} \cdot \sin(\pi l) \]  

(9)

In the equation (9), \( D_t = |x_i^{t-1} - c \cdot x_s| \) is the distance between the bat individual before the location update and the current optimal individual, which can control the bat individual distance from the current optimal individual not too far away. \( c = 2 \cdot b \) influences the distance between the bat individual \( x_i^{t-1} \) before the position update and the current optimal individual \( x_s \). When \( c > 1 \), the degree of influence increases, and when \( c < 1 \), the degree of influence decreases. \( b \) is a random number that obeys influence distribution. \( e \) is the natural constant. \( l \) is a number generated randomly in the range of \([-1, 1] \), \( e^{-l} \) and \( \sin(\pi l) \) control the bat individual not too close to the current optimal individual.
Random perturbation of trend

In order to further improve the global convergence accuracy and avoid the algorithm falling into local extremum, a random perturbation mechanism was added after the bat individuals were updated. The specific way of perturbation is as follows:

\[
\theta = \theta_{\text{max}} - (\theta_{\text{max}} - \theta_{\text{min}}) \cdot \cos \left( \frac{\pi}{2} \left( \frac{t}{T_{\text{max}}} - 1 \right) \right) \tag{10}
\]

\[
x^t_i = x_i + \theta \cdot \mu \cdot (x_i - x_{\text{rand}}) \tag{11}
\]

where \(x^t_i\) is the position of the \(i\)th bat individual before the disturbance, and \(x_i\) is the position of the \(i\)th bat individual after the disturbance. \(t\) is the current number of iterations. \(T_{\text{max}}\) is the maximum number of iterations. \(\theta\) is the nonlinear perturbation coefficient. \(\theta_{\text{max}}\) and \(\theta_{\text{min}}\) are the values of \(\theta\) respectively. After many simulating tests, the best algorithm optimization effect is when \(\theta_{\text{max}} = 0.9\), \(\theta_{\text{min}} = 0.3\). \(\mu\) is a uniform random number in the interval \([0, 1]\), \(x_{\text{rand}}\) is a randomly chosen bat individual from the previous generation of bat populations. It can be seen from equation (10) that the perturbation coefficient \(\theta\) decreases nonlinearly with the increase of iteration times, which makes the position disturbance item in equation (11) decrease randomly. It not only increases the diversity of the population, improves the ability to jump out of local extremum, but also maintains the ability of fine mining near the optimal solution in the late iteration.

Pseudo code of the improved algorithm

begin

Objective function \(f(x), x = (x_1, \ldots, x_d)^T\);

Generate initial population of bat individuals, \(x_i = (1, 2, \ldots, n)\);

Define each parameter \(v_i, A_i, r_i, h_{\text{max}}, h_{\text{min}}, T_{\text{max}}, d_i\);

Evaluate the fitness function value for each bat;

Compare these fitness values and find the current optimal value \(f_{\text{min}}\) and the current optimal bat position \(x^*\);

\(t = 1\);

while (\(t \leq T_{\text{max}}\)) do

Use equations (1), (7) and (3) to update every bat’s rate of pulse emission, velocity and position. So as to get the new bat position \(x^t_i\);

Evaluate the fitness function value for \(x^t_i\);

Generate uniform random number \(p\);

if \((P < r_i)\) then

Use the equation (9) to perform a local search near the current optimal individual to create a new position \(x_{\text{new}}\). Replace the value of \(x^t_i\) with the value of \(x_{\text{new}}\) at this time;

end if

Use equations (10) and (11) to perturb the bat position \(x^t_i\) to create a new bat position \(x'_i\);

Evaluate the fitness function value for \(x'_i\);

if \((f(x'_i) < f(x^t_i))\) then

Replace the value of \(x^t_i\) with the value of \(x'_i\);

end if

Generate uniform random number \(r\);

if \((r > A_i \& f(x'_i) < f(x^t_i - 1))\) then

Accept the bat individual position \(x'_i\) resulting from this iteration updated;

Update the loudness \(A_i\) and the rate of pulse emission \(r\) based on the equations (5) and (6);

end if

Accept \(f(x'_i)\) as the current optimal solution and receive \(x'_{\text{new}}\) as the current best individual;

end if

end while

Output the global optimal solution;

end

Experiment and result analysis

Optimization test function

In order to verify the optimization ability of the improved algorithm in this paper, the particle swarm optimization algorithm (PSO), basic bat algorithm (BA), a novel bat algorithm with habitat selection and Doppler effect in echoes for optimization (NBA) and improved algorithm of this paper (OPBA) were used to compare simulating experiments on 10 classical test functions. The specific test function is as follows:

Sphere function.

\[
f_1(x) = \sum_{i=1}^{n} x_i^2
\]

This function gets a minimum value 0 at \((0, \cdots, 0)\).

Schwefel’s problem 2.21 function.

\[
f_2(x) = \text{MAX}_i \{|x_i|, 1 \leq i \leq n\}
\]

This function gets a minimum value 0 at \((0, \cdots, 0)\).

Griewank function.

\[
f_3(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right)
\]

This function gets a minimum value 0 at \((0, \cdots, 0)\).
Ackley function.
\[ f_4(x) = \sum_{i=1}^{n} i \cdot x_i^2 \]
This function gets a minimum value 0 at \((0, \cdots, 0)\).

Alpine function.
\[ f_5(x) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1x_i| \]
This function gets a minimum value 0 at \((0, \cdots, 0)\).

Rotated hyper-ellipsoid function.
\[ f_6(x) = \sum_{i=1}^{n} \sum_{j=1}^{i} x_j^2 \]
This function gets a minimum value 0 at \((0, \cdots, 0)\).

Branins'srcos function.
\[ f_7(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(x_1) + 10 \]
This function gets a minimum value 0.3979 at. \((-\pi, 12.275), (\pi, 12.275)\) and \((9.42478, 2.275)\).

Ackley function.
\[ f_8(x) = 20 + e - 20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) \right) \]
This function gets a minimum value 0 at \((0, \cdots, 0)\).

Easom function.
\[ f_9(x) = -\cos(x_1)\cos(x_2) \times \exp \left[ -\left( (x_1 - \pi)^2 + (x_2 - \pi)^2 \right) \right] \]
This function gets a minimum value 0 at \((\pi, \pi)\).

Powell function.
\[ f_{10}(x) = \sum_{i=1}^{n/4} \left[ (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-2} - x_{4i})^2 \right. \\
+ \left. (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \right] \]
This function gets a minimum value 0 at \((0, \cdots, 0)\).

Among these 10 test functions, \(f_1(x), f_2(x), f_4(x), f_6(x)\) and \(f_9(x)\) are unimodal functions which focus on monitoring the accuracy of the algorithm. \(f_3(x), f_5(x), f_7(x)\) and \(f_{10}(x)\) are complex multi-peak functions which contain more local minimum values, and can fully test the ability of the algorithm to jump from local minimum. \(f_3(x)\) and \(f_7(x)\) have many local minimum values and are easy to converge prematurely. It can verify the overall optimization ability of the algorithm. Therefore, these 10 test functions all have some difficulty in solving and are suitable for the optimization performance of test algorithms.

**Experimental environment and parameter settings**

In order to ensure the objectiveness and fairness of the algorithm comparison, the PSO, BA, NBA, and OPBA algorithms all use the same hardware and software platforms. The operating environment is Windows7 and the programming environment is Matlab R2014a. In the simulating experiment, the population size, maximum number of iterations, and space dimension of the four algorithms are consistent, that is, \(n = 50, T_{\text{max}} = 1000, d = 2/50/100\). The acoustic loudness and pulse emission rate in BA, NBA and OPBA algorithms mainly control the local search ability of the algorithm. Based on the fairness, the settings of the above two parameters are consistent with those in the original literature and source code of the basic bat algorithm. That is \(A_0 = 0.25, r_0 = 0.5\). The disturbance coefficient \(\theta\) in OPBA algorithm determines the step size of nonlinear disturbance, and has an important impact on the diversity and convergence accuracy of the algorithm. \(\theta_{\text{max}}\) and \(\theta_{\text{min}}\) are the maximum and minimum values of \(\theta\). The optimization effect is the best when \(\theta_{\text{max}} = 0.9, \theta_{\text{min}} = 0.3\). The learning factors \(c_1\) and \(c_2\) in PSO algorithm determines the influence of particle individual experience and colony experience on particle trajectory respectively. In standard PSO algorithm, individual experience and group experience have the same important influence. Generally set to \(c_1 = c_2 = 1.5\), the same values are used in this paper. The inertial weight \(\omega\) is used to control the development and exploration ability of the algorithm. When the value of \(\omega\) is larger, the global optimization ability is stronger, the local optimization ability is weak, and when \(\omega\) value is small, the global optimization ability is weak, and the local optimization ability is stronger. In the standard PSO algorithm, the value of \(\omega\) is generally set to \(\omega = 0.5\).

**Optimization accuracy analysis**

The four algorithms PSO, BA, NBA, and OPBA run independently 30 times in the same environment.
The dimension of the high-dimensional functions \( f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x), f_8(x) \) and \( f_{10}(x) \) is \( d = 50, 100 \), and the dimension of the low-dimensional functions \( f_7(x) \) and \( f_9(x) \) is \( d = 2 \). Tables 1 to 10 show the worst solutions, optimal solutions, averages, and variances of the four algorithms, with the best solutions being bolded.

It can be seen from the data in Tables 1 to 10 that the accuracy and stability of the OPBA algorithm proposed in this paper are better than that of the other three algorithms under the same dimension of the same function in higher dimensions. In the high-dimensional single-peak function \( f_1(x), f_2(x), f_4(x), f_6(x) \), the accuracy of the four algorithms will decrease with the increase of the dimension, but the accuracy and stability of the OPBA algorithm in different dimensions is better than that of the other three algorithms. In the high-dimensional multimodal function \( f_3(x), f_5(x), f_8(x), f_{10}(x) \), for the function \( f_3(x), f_8(x) \), the OPBA algorithm can obtain the global optimal solution in both 50-dimensional and 100-dimensional, the variances are also the smallest, while the other three algorithms are not only unable to obtain the global optimal solution but also relatively less stable. For the function \( f_5(x), f_{10}(x) \), the optimization accuracy of the OPBA algorithm is higher than that of the other three algorithms by several tens or even one hundred orders of magnitude, and the solution effect is very obvious. Moreover, the variances of the OPBA algorithm are smaller than the other three algorithms, and the stability also is better. This is because the trend random perturbation mechanism proposed in this

### Table 1. Four algorithms simulating results in function \( f_1(x) \).

| D  | Algorithm | Worst  | Best   | Mean       | Variance |
|----|-----------|--------|--------|------------|----------|
| 50 | PSO       | 3.8445 | 1.4057 | 2.8630     | 0.5302   |
|    | BA        | 5.2989e-05 | 3.4793e-05 | 4.4399e-05 | 1.8647e-11 |
|    | NBA       | 1.7038e-89 | 1.1708e-120 | 6.0215e-91 | 9.6559e-180 |
|    | OPBA      | 1.4616e-120 | 1.0880e-127 | 1.3164e-121 | 1.3594e-241 |
| 100| PSO       | 14.4179 | 7.0086 | 9.5182     | 2.8739   |
|    | BA        | 2.6493e-04 | 1.8340e-04 | 2.1984e-04 | 4.1784e-10 |
|    | NBA       | 8.2630e-58 | 1.9340e-91 | 2.7543e-59 | 2.2759e-116 |
|    | OPBA      | 3.7932e-117 | 7.0181e-124 | 2.2744e-118 | 5.7473e-235 |

### Table 2. Four algorithms simulating results in function \( f_2(x) \).

| D  | Algorithm | Worst  | Best   | Mean       | Variance |
|----|-----------|--------|--------|------------|----------|
| 50 | PSO       | 0.7784 | 0.4091 | 0.5762     | 0.0073   |
|    | BA        | 0.5191 | 0.2668 | 0.3859     | 0.0049   |
|    | NBA       | 0.3375 | 0.0025 | 0.1390     | 0.0090   |
|    | OPBA      | 2.7690e-06 | 2.7231e-22 | 1.6565e-07 | 3.1647e-13 |
| 100| PSO       | 0.9016 | 0.6076 | 0.7339     | 0.0051   |
|    | BA        | 0.6951 | 0.4036 | 0.5456     | 0.0056   |
|    | NBA       | 0.5577 | 0.3314 | 0.4561     | 0.0043   |
|    | OPBA      | 0.3239 | 5.3147e-19 | 0.0108    | 0.0035   |

### Table 3. Four algorithms simulating results in function \( f_3(x) \).

| D  | Algorithm | Worst  | Best   | Mean       | Variance |
|----|-----------|--------|--------|------------|----------|
| 50 | PSO       | 0.1360 | 0.0391 | 0.0764     | 3.1672e-04 |
|    | BA        | 2.3699e-06 | 1.3070e-06 | 1.7386e-06 | 6.6648e-14 |
|    | NBA       | 0       | 0      | 0          | 0        |
|    | OPBA      | 0       | 0      | 0          | 0        |
| 100| PSO       | 0.1604 | 0.0854 | 0.1204     | 3.2494e-04 |
|    | BA        | 8.8035e-06 | 4.8195e-06 | 6.3584e-06 | 1.0671e-12 |
|    | NBA       | 1.1102e-16 | 3.7007e-18 | 4.1087e-34 | 0        |
|    | OPBA      | 0       | 0      | 0          | 0        |
paper makes it easier for the algorithm to jump out of the local extremum, thus improving the optimization accuracy of the algorithm. For the low-dimensional function $f_7(x)$, PSO, BA, NBA, and OPBA can all explore the theoretical optimal solution. There is no difference in the optimization accuracy, and the variances are not much different. However, from the analysis of the convergence curve in the next section, the OPBA algorithm has faster convergence speed than other algorithms. This is because the moderately optimal orientation strategy adopted in this paper makes the bat population fast to the optimal direction, thus greatly improving the algorithm optimization speed. The above experimental results show that the improved algorithm OPBA has more advantages than PSO, BA and NBA algorithms and has a better stability in different dimensions.

**Analysis of convergence curve**

The convergence curve can visually show the convergence speed of the algorithm and the ability of the algorithm to jump out of the local extremum. It is an important standard to measure the performance of the

---

**Table 4.** Four algorithms simulating results in function $f_4(x)$.

| D  | Algorithm | Worst   | Best    | Mean    | Variance |
|----|-----------|---------|---------|---------|----------|
| 50 | PSO       | 108.4736| 25.4200 | 67.3696 | 403.8732 |
|    | BA        | 0.0044  | 0.0011  | 0.0027  | 8.1424e-07 |
|    | NBA       | 4.8972e-86| 3.8450e-121| 1.6482e-87| 7.9892e-173 |
|    | OPBA      | 6.5336e-119| 1.2526e-125| 3.5454e-120| 1.7164e-238 |
| 100| PSO       | 555.5021| 255.4887| 392.3255| 5.7516e+03 |
|    | BA        | 0.2636  | 0.0296  | 0.0830  | 0.0035   |
|    | NBA       | 1.4235e-62| 5.2865e-95| 6.8143e-64| 7.4661e-126 |
|    | OPBA      | 6.7700e-116| 1.4027e-123| 3.2352e-117| 1.5530e-232 |

---

**Table 5.** Four algorithms simulating results in function $f_5(x)$.

| D  | Algorithm | Worst   | Best    | Mean    | Variance |
|----|-----------|---------|---------|---------|----------|
| 50 | PSO       | 3.8710  | 1.5310  | 2.6801  | 0.3453   |
|    | BA        | 0.0093  | 0.0037  | 0.0058  | 3.1068e-06 |
|    | NBA       | 0.0238  | 1.3297e-71| 0.0082  | 4.0451e-05 |
|    | OPBA      | 1.1031e-66| 5.4473e-72| 7.2591e-68| 4.2594e-134 |
| 100| PSO       | 10.9957 | 6.2427  | 8.3261  | 1.8508   |
|    | BA        | 0.0244  | 0.0143  | 0.0185  | 9.4809e-06 |
|    | NBA       | 0.0595  | 1.6091e-05| 0.0319  | 2.5990e-04 |
|    | OPBA      | 4.4168e-66| 3.7968e-70| 4.3433e-67| 9.7949e-133 |

---

**Table 6.** Four algorithms simulating results in function $f_6(x)$.

| D  | Algorithm | Worst   | Best    | Mean    | Variance |
|----|-----------|---------|---------|---------|----------|
| 50 | PSO       | 101.1651| 30.7429 | 65.6426 | 319.4510 |
|    | BA        | 0.0051  | 0.0013  | 0.0031  | 1.2803e-06 |
|    | NBA       | 7.2126e-80| 3.4296e-125| 2.4042e-81| 1.7340e-160 |
|    | OPBA      | 1.3737e-118| 8.6293e-126| 8.2894e-120| 7.9117e-238 |
| 100| PSO       | 612.7605| 301.9635| 431.0787| 5.5980e+03 |
|    | BA        | 0.1570  | 0.0204  | 0.0635  | 0.0011   |
|    | NBA       | 8.0752e-66| 2.3505e-93| 5.4356e-67| 3.3595e-132 |
|    | OPBA      | 2.4272e-115| 6.3277e-124| 1.4824e-116| 2.7901e-231 |

---

**Table 7.** Four algorithms simulating results in function $f_7(x)$.

| D  | Algorithm | Worst   | Best    | Mean    | Variance |
|----|-----------|---------|---------|---------|----------|
| 2  | PSO       | 0.3979  | 0.3979  | 0.3979  | 0        |
|    | BA        | 0.3979  | 0.3979  | 0.3979  | 7.1782e-20 |
|    | NBA       | 0.3979  | 0.3979  | 0.3979  | 0        |
|    | OPBA      | 0.3979  | 0.3979  | 0.3979  | 4.0956e-12 |
algorithm. Figures 1 to 10 is a comparison chart of the convergence curves of the four algorithms.

It can be seen from Figures 1, 4, 5, 6, 8, and 10, the convergence rate of OPBA is much faster than that of PSO and BA. Although the convergence curves of OPBA and NBA are almost coincident, the convergence rate of OPBA is slightly faster than NBA. It can be seen from Figures 2 and 3, although the early convergence speed of OPBA algorithm is not as fast as that of NBA, near the 10th generation, OPBA convergence speeds up. After that, the convergence speed is always higher than that of NBA, indicating that OPBA easily jumps from local optimum. From Figures 7 and 9, we can see that OPBA, NBA, and PSO algorithms achieve the final convergence in about 10 generations, and the convergence speed of these three algorithms is obviously better than

**Table 8.** Four algorithms simulating results in function $f_8(x)$.

| D  | Algorithm | Worst | Best  | Mean   | Variance |
|----|-----------|-------|-------|--------|----------|
| 50 | PSO       | 2.6986| 2.0033| 2.3260 | 0.0361   |
|    | BA        | 2.2007| 0.0034| 1.1704 | 0.4598   |
|    | NBA       | 1.5099e-14 | 4.4409e-15 | 8.8226e-15 | 6.6881e-30 |
|    | OPBA      | 0     | 0     | 0      | 0        |
| 100| PSO       | 2.9874| 2.3157| 2.6516 | 0.0262   |
|    | BA        | 2.1098| 0.0057| 1.1001 | 0.3775   |
|    | NBA       | 2.2204e-14 | 7.9936e-15 | 1.191e-14  | 1.5973e-29 |
|    | OPBA      | 0     | 0     | 0      | 0        |

**Table 9.** Four algorithms simulating results in function $f_9(x)$.

| D  | Algorithm | Worst | Best  | Mean   | Variance |
|----|-----------|-------|-------|--------|----------|
| 2  | PSO       | -1.0000 | -1.0000 | -1.0000 | 9.7435e-21 |
|    | BA        | -1.0000 | -1.0000 | -1.0000 | 9.7435e-21 |
|    | NBA       | -1.0000 | -1.0000 | -1.0000 | 9.7435e-21 |
|    | OPBA      | -1.0000 | -1.0000 | -1.0000 | 9.7435e-21 |

**Table 10.** Four algorithms simulating results in function $f_{10}(x)$.

| D  | Algorithm | Worst | Best  | Mean   | Variance |
|----|-----------|-------|-------|--------|----------|
| 50 | PSO       | 80.8390 | 14.1991 | 43.5017 | 246.3672 |
|    | BA        | 0.0321 | 0.0110 | 0.0206 | 1.7735e-05 |
|    | NBA       | 6.4558e-05 | 6.7063e-10 | 5.396e-06 | 1.8727e-10 |
|    | OPBA      | 2.1905e-37 | 3.0823e-108 | 7.3019e-39 | 1.5994e-75 |
| 100| PSO       | 336.8756 | 101.0367 | 209.0595 | 3.2154e+03 |
|    | BA        | 0.2646 | 0.0704 | 0.1285 | 0.0016   |
|    | NBA       | 6.2789e-05 | 2.2813e-08 | 9.5989e-06 | 2.6274e-10 |
|    | OPBA      | 9.0590e-105 | 5.7167e-120 | 3.1105e-106 | 2.7321e-210 |

Figure 1. The convergence curve of Sphere function.
BA algorithm, but the convergence speed of OPBA algorithm is slightly faster than that of NBA and PSO algorithms. The above experimental results show that the convergence rate of the improved algorithm in this paper is obviously higher than that of the other three algorithms, and oscillating phenomenon rarely occurs in the convergence process. This is because the mutation factor is introduced into the speed update formula of global search phase in OPBA algorithm, which expands the search range and diversity of bat population. And in the local search stage, a Modestly trend towards optimal
value strategy is adopted, and an adaptive random perturbation mechanism is added in each generation of evolution, which reduces the probability of getting trapped in local extremes and speeds up the convergence of the algorithm.

Conclusion
Aiming at the disadvantages that bat algorithm is easy to fall into local extremum, the precision of searching is not high, and the convergence speed of the algorithm is slow in the later period and so on. This paper proposes a bat optimization algorithm with moderate orientation and perturbation of trend. The algorithm introduces a nonlinear variation factor in the global search stage, which controls the step size of bat individual and expands the bat population search range. In the local search stage, a strategy of modestly trend towards optimal value was adopted to improve the local search ability of the algorithm. At the same time, introducing a random perturbation strategy that decreases with the evolutionary trend makes it easier for the algorithm to jump out of the local extremum, which well balances the breadth of the global search in the early stage and the depth of the local mining in the later period. The simulating results show that the improved algorithm in this paper has faster convergence speed and higher search accuracy.

Declaration of Conflicting Interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is jointly supported by the Science & Technology Program of Henan Province, China (Grant No. 182102310886), the Postgraduate Education Innovation and Quality Improvement Project of Henan University (Grant No. SYL19050104 and SYL18020105).

ORCID iD
Jingsen Liu https://orcid.org/0000-0002-8054-685X

References
1. Goldberg DE. Genetic algorithm in search optimization and machine learning. Boston, MA: Addison Wesley, 1989, pp.2104–2116.
2. Kennedy J and Eberhart R. Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Perth, 12, 1995, pp.1942–1948. Piscataway, NJ: IEEE.
3. Shi Y and Eberhart R. A modified particle swarm optimizer. In: IEEE international conference on evolutionary computation, Anchorage, 05, 1998, pp.69–73. Piscataway, NJ: IEEE.
4. Dorigo M, Maniezzo V and Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B Cybern 1996; 26: 29–41.
5. Yang XS and Deb S. Cuckoo search via levy flights. In: Proceeding of world congress on nature & biologically inspired computing. Coimbatore, India, 12, 2009, pp. 210–214. Piscataway, NJ: IEEE.
6. Yang XS and Deb S. Engineering optimization by cuckoo search. *IJMIMO* 2010; 1: 330–343.
7. Mirjalili S, Mirjalili SM and Lewis A. Grey wolf optimizer. *Adv Eng Software* 2014; 69: 46–61.
8. Mirjalili S and Lewis A. The whale optimization algorithm. *Adv Eng Software* 2016; 95: 51–67.
9. Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 2016; 27: 1053–1073.
10. Yang XS. A new metahuristic Bat-Inspired algorithm. *Comput Knowled Technol* 2010; 284: 65–74.
11. Alsariera YA, Majid MA and Zamli KZ. Adopting the bat-inspired algorithm for interaction testing. In: *The 8th edition of conference of software testing*, Mandarin Oriental Hotel, 9, 2015.
12. Kashi S, Minuchehr A, Poursalehi N, et al. Bat algorithm for the fuel arrangement optimization of reactor core. *Ann Nucl Energy* 2014; 64: 144–151.
13. Chakri A, Khelif R, Benouaret M, et al. New directional bat algorithm for continuous optimization problems. *Exp Syst Appl* 2017; 69: 159–175.
14. Tharakeshwar TK, Seetharamu KN and Prasad BD. Multi-objective optimization using bat algorithm for shell and tube heat exchangers. *Appl Therm Eng* 2017; 110: 1029–1038.
15. Grewal NS, Rattan M and Patterh MS. A linear antenna array failure correction using improved bat algorithm. *Int J RF Microw Comput Aided Eng* 2017; 27: e21119.
16. Saji Y and Riffi ME. A novel discrete bat algorithm for solving the travelling salesman problem. *Neural Comput Appl* 2016; 27: 1853–1866.
17. Xu B, Qi J, Hu X, et al. Self-adaptive bat algorithm for large scale cloud manufacturing service composition. *Peer to Peer New Appl* 2018; 11: 1115–1128.
18. Ghanem WAHM and Jantan A. An enhanced bat algorithm with mutation operator for numerical optimization problems. *Neural Comput Appl* 2019; 31: 617–651.
19. Li MW, Geng J, Wang S, et al. Hybrid chaotic quantum bat algorithm with SVR in electric load forecasting. *Energies* 2017; 10: 2180.
20. He XS, Ding WJ and Yang XS. Bat algorithm based on simulated annealing and Gaussian perturbations. *Neural Comput Appl* 2014; 25: 459–468.
21. Alihodzic A and Tuba M. Improved hybridized bat algorithm for global numerical optimization. In: *Ukmim-Annms, international conference on computer modeling and simulating*, Cambridge, UK, 2014, 03, pp. 57–62. Piscatway, NJ: IEEE Computer Society.
22. Chen-Hao HU, Yang RF, Guo CX, et al. Improved bat algorithm with particle swarm optimization acceleration factor and individual cognition. *Sci Technol Eng* 2017; 17: 277–282.