A Novel BBO Algorithm Based on Local Search and Nonuniform Variation for Iris Classification

Lisheng Wei, Ning Wang, and Huacai Lu

1Anhui Key Laboratory of Electric Drive and Control, Anhui Polytechnic University, Wuhu 241000, China
2Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment of Ministry of Education, AnHui Polytechnic University, Wuhu 241000, China
3Anhui Key Laboratory of Detection Technology and Energy Saving Devices, Anhui Polytechnic University, Wuhu 241000, China

Correspondence should be addressed to Lisheng Wei; lshwei_11@163.com and Ning Wang; 18375318758@163.com

Received 6 November 2020; Revised 8 January 2021; Accepted 27 March 2021; Published 15 April 2021

1. Introduction

Iris classification is especially suitable for recognition with uniqueness, stability, inviolability, and reliability characteristics and has been one of the hottest biological characteristic recognition research spot recently [1]. It has been widely used technology in national defence, financial industry, and entrance guard system [2, 3]. Alfred Wallace and Charles Darwin proposed the theory of biogeography to study the distribution, migration, and extinction of habitats of biological species in the 19th century. Inspired by biogeography, Simon proposed the BBO algorithm in 2008 [4]. BBO is a new swarm intelligence optimization algorithm and solves the optimization problem by simulating the mathematical model of species migration in biogeography. In general, BBO can find a good initial candidate solution and generate an acceptable optimization solution, which has been widely used in image processing [5], scheduling optimization [6, 7], parameter estimation [8], power flow calculation [9], and load analysis [10]. However, BBO has some shortcomings, such as weak search ability and easy to fall into local optimum in the later period of operation. How to use BBO global search ability to improve the iris classification rate has important theoretical and practical value.

In recent years, many researchers have worked on the improvement of the BBO performance. For instance, Wang et al. [11] combined the chaotic mapping strategy with the BBO optimal migration model and proposed a biogeography optimization algorithm based on adaptive population migration mechanism. Feng et al. [12] proposed an improved BBO with random ring topology and Powell’s method, in which the local loop topology was used instead of the global topology. The adaptive Powell method was modified to adapt the evolutionary process to improve the accuracy of the solution. Li et al. [13] designed a hybrid algorithm by combining the artificial bee colony algorithm (ABC) with BBO to obtain ABC’s exploring ability and BBO’s developing ability. Chen et al. [14] combined Cuckoo Search (CS) and bBBO using two search strategies: heterogeneous rhododendron search and biogeography-based discovery. Zhao et al. [15] proposed an optimization method based on two-stage differential biogeography to solve the
In order to improve the iris classification rate, the NBBO based on local search and nonuniform variation was proposed in this paper. Firstly, the hyperbolic cotangent migration model was used to replace the original linear migration model to improve the adaptability of the algorithm. Then, the local search strategy was added to the BBO migration operation in order to enhance the algorithm search ability and improve the convergence speed of the algorithm. The nonuniform variation was used in the operation to increase the search ability of the algorithm and prevent the local optimum, which could achieve a stronger iris classifier by lifting weaker similarity classifiers during the training stage.

The rest of the paper was organized as follows. Section 2 presented the improved biogeography-based optimization algorithm, including migration model, migration operations, and nonuniform variation operation. Section 3 presented NBBO test on different test functions and iris classification samples. Finally, a brief conclusion and future work are given in Section 4.

2. Methods

In nature, biological species live in different areas with obvious boundaries, which are called habitats. Habitat suitability index (HSI) is used to describe the survival degree of species suitable for habitat. HSI is affected by humidity, temperature, habitat area, and vegetation diversity. These factors are called suitability variables (SIV). Habitats with higher HSI can accommodate more biological species, whereas habitats with lower HSI can only accommodate fewer biological species. BBO is a population-based algorithm, which regards each solution as a habitat, fitness as HSI, and each component of solution as a SIV. The optimization problem can be solved by simulating the migration and mutation process in biogeography. The two main operations of BBO are migration operation and mutation operation [16, 17].

Like other intelligent algorithms, BBO is prone to oscillation in the iterative process, resulting in slow and premature convergence, poor local search ability, and inaccuracy. This section presents the NBBO algorithm in terms of migration model and operation. The flow chart of the improved algorithm is shown in Figure 1.

2.1. Migration Model. There are different migration models in biogeography. As shown in formula (1), the original BBO adopts the simplest linear migration model. But, the ecosystem is nonlinear in nature; small changes in a certain part of the system may have complex effects on the whole system. For the actual species, the migration process is more complicated than what is described by the linear model. Among the related improved algorithms, the cosine mobility model proposed by Ma performs best under most of the functions tested [18], and the mobility model close to the natural law is better than the simple linear model. In order to better adapt to the nonlinear migration problem, this paper proposes a hyperbolic cotangent nonlinear migration model to improve the performance of the basic BBO algorithm.

\[
\begin{align*}
\lambda_k &= \frac{1}{k} \cdot \left(1 - \frac{k}{n}\right), \\
\mu_k &= E \cdot \frac{k}{n}, \\
\lambda_k &= \frac{1}{2} \cdot \left(\frac{e^{k/n} - e^{-k/n}}{e^{k/n} + e^{-k/n}} + 1\right), \\
\mu_k &= E \cdot \frac{e^{k/n} - e^{-k/n}}{e^{k/n} + e^{-k/n} + 1},
\end{align*}
\]

where \(\lambda_k\) and \(\mu_k\) are related to the number of species \(k\) and \(n\) is the maximum number of species. \(\lambda_k\) represents the immigration rate which refers to the probability of species moving into the habitat, \(\mu_k\) represents the emigration rate which refers to the probability of species moving out of the habitat, \(I\) represents the maximum immigration rate, and \(E\) represents the maximum emigration rate, all of which are functions of the number of habitat populations \(k\). When the number of habitat species is 0, the immigration rate is the highest, that is, \(\lambda_k = I\), the migration rate is the smallest, that is, \(\mu_k = 0\). When the number of species reaches the

![Figure 1: NBBO algorithm flowchart.](image-url)
maximum number of species \( n \) that can be accommodated in the habitat, the immigration rate \( \lambda_k \) drops to 0, and the migration rate reaches the maximum, that is, \( \mu_k = E \). When the number of species is \( k_0 \), the immigration rate and the emigration rate of the species are equal, that is, \( \lambda_k = \mu_k \), and the dynamic equilibrium state is reached at this time.

In the hyperbolic tangential mobility model, the trend of mobility changing with species number is like that of cosine model, but the amplitude is more moderate than that of cosine model.

### 2.2. Local Search Migration Operations

In the original BBO, a generalized search was carried out based on the immigration rate and the emigration rate, a habitat with a higher immigration rate was randomly selected, and the characteristics of the corresponding position (SIV) were moved to the habitat with low HSI. However, in some habitats with high migration rates, the characteristics of the corresponding locations SIV are not necessarily the best for the migrated habitats. If migrated directly, it may result in a decrease in the suitability index of the move into habitat.

Therefore, in order to enhance the global search ability of the algorithm and improve the convergence speed of the algorithm, the selected migration characteristics can be modified by using equation (6), which can reduce the poor migration characteristics from the habitat with higher migration rate to the low migration rate. Similarly, if you move out of the same habitat with a high migration rate each time, as the number of evolutions increases, the population diversity decreases, and the convergence speed will slow down:

\[
C = \text{MinVal} + \text{rand} \ast (\text{MaxVal} - \text{MinVal}), \tag{4}
\]

\[
\text{randTemp} = \text{rand}(1, 2), \tag{5}
\]

\[
X'(k, j) = X(k, j) + C \ast \text{randTemp} \ast X(k, j), \tag{6}
\]

where \( C \) represents a space allocated to each individual in the population for storing algorithm parameters. \( X(k, j) \) represents the current individual. \( X'(k, j) \) represents the new individual generated by the local search. MinVal = -1 and MaxVal = 1.

### 2.3. Nonuniform Variation Operation

In the early stage of iteration, the population position is usually far from the optimal solution and needs to be iterated many times to find a feasible search space when the search radius is too small. At the end of iteration, the population position is close to or near the optimal solution. A small range of searches is used as a fine-tuning of the solution vector. It is shown by equation (7), as the traditional mutation operator cannot search the optimal solution efficiently, the nonuniform mutation operator is introduced. The whole population is searched at a large range during the early stage of search. As the number of iterations increases, the search range is gradually narrowed, and the algorithm is effectively balanced to escape from local minimum.

\[
m_s = m_{\text{max}} \cdot \left(1 - \frac{P}{P_{\text{max}}}ight), \tag{7}
\]

where \( P_s \) denotes the probability for the \( S \) number of species; \( S \) denotes the species number; and \( m_{\text{max}} \) denotes the maximum variation rate given; \( P_{\text{max}} = \text{max}(P_s) \).

The specific operation of nonuniform variation is like uniform variation, but it focuses on searching for small areas near the original individual. Assuming that the variation of the \( k_0 \) component \( X_k \) of the \( t \) generation (\( t \) is the current number of iterations) of Habitat \( X_t = [X_1, X_2, \ldots, X_k, \ldots, X_n] \) is performed, the variation component is as follows:

\[
X'_k = X_k + \Delta(t, U_{\text{max}} - X_k), \quad \text{if rand} < 0.5, \tag{8}
\]

\[
X'_k = X_k - \Delta(t, X_k - U_{\text{min}}), \quad \text{if rand} \geq 0.5, \tag{9}
\]

where \( U_{\text{max}} \) and \( U_{\text{min}} \) are the upper and lower limits of component \( X_k \), respectively. \( U_{\text{max}} - X_k \) and \( X_k - U_{\text{min}} \) represent random numbers in the range of \([0, 1]\) and meet the nonuniform distribution:

\[
\lim_{t \to \infty} \Delta(t, y) = 0. \tag{10}
\]

This feature allows the mutation operator to search the entire space evenly in the initial stage of the algorithm (when \( t \) is very small), and the latter part of the algorithm focuses on several local range exact searches. \( \Delta(t, y) \) can be defined as follows:

\[
\Delta(t, y) = y \cdot \left(1 - r(1-1/(Tt))b\right), \tag{11}
\]

where \( r \) is a random number that fits evenly within the \([0, 1]\) range, \( T \) is the maximum number of iterations, and \( b \) is the system parameter that determines the degree of dependency on the number of iterations, select here \( b = 2 \).

Then, the specific steps of NBBO algorithm are as follows: (Algorithm 1)

### 2.4. Convergence Analysis

By using Markov chain strategy, we can analyze the convergence of the proposed NBBO algorithms [19, 20]. Let \( X(t) = [x_1(t), x_2(t), \ldots, x_n(t)] \) be a population with improved algorithm algebra of \( t \). We divide \( X(t) \) into \( n \) subsets based on the fitness of the population. \( x_i(t) (i = 1, 2, \ldots, n) \) is a subset of them; next, we use Markov chain to analyze the global convergence of the improved BBO algorithm.

The evolution process of the NBBO algorithm is mainly composed of selection, migration and mutation operations, which are independent of evolution algebra. Considering the number of individuals of BBO algorithm, the evolution process of the NBBO algorithm satisfies the finite homogeneous Markov chain. Population subset \( x_i(t) (i = 1, 2, \ldots, n) \) can be equivalent to a state on a finite homogeneous Markov chain, \( p_i \) means the probability of being in state \( x_i(t) \), and then \( p_i > 0 \) and \( \sum_{i=1}^{n} p_i = 1 \).

The transition probability \( p_{ij} = P(x_i \longrightarrow x_j) \) of state \( x_i(t) \) is as follows:
Then, the state transition matrix \( P \) of the Markov chain is

\[
P = \begin{bmatrix}
P_{11} & 0 & \cdots & 0 \\
P_{21} & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots \\
0 & \cdots & \cdots & \cdots \\
\end{bmatrix} = \begin{bmatrix}
P_{11} & 0 \\
Q_2 & Q_1 \\
\end{bmatrix},
\]

where

\[
Q_1 = \begin{bmatrix}
P_{22} & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \cdots \\
\end{bmatrix},
Q_2 = \begin{bmatrix}
P_{21} & P_{31} & \cdots & P_{nl} \\
\end{bmatrix}^T.
\]

It can be found by formulas (12) and (13) \( Q_2 \neq 0, Q_1 \neq 0 \). Since the sum of the probabilities of each row in the Markov transition matrix is 1, \( P_{11} = 1 \neq 0 \). It can be concluded that the state transition matrix \( P \) of the Markov chain satisfies the reductive random matrix. According to [21], the definition of a reductive random matrix is as follows:

Let matrix \( Z \) of order \( N \) be a reductive random matrix which can be obtained by the same row transformation and column transformation, namely,

\[
Z = \begin{bmatrix}
C & 0 \\
R & T
\end{bmatrix},
\]

where \( C \) is the \( m \) (\( m \leq N \)) order primitive matrix and \( R \) and \( T \) are nonzero matrices.

Then, we have

\[
Z^\infty = \lim_{k \to \infty} Z^k = \lim_{k \to \infty} \left[ \begin{array}{c}
C^k & 0 \\
R^k & T^k
\end{array} \right] = \left[ \begin{array}{c}
C^\infty & 0 \\
R^\infty & T^\infty
\end{array} \right].
\]

The above matrix is a stable random matrix, and \( Z^\infty \) satisfies the following conditions:

\[
Z^\infty = \left[ \begin{array}{c}
Z_{ij} \\
N \times N
\end{array} \right] \begin{cases}
Z_{ij} > 0, & 1 \leq i \leq N, 1 \leq j \leq m, \\
Z_{ij} = 0, & 1 \leq i \leq N, m < j \leq N.
\end{cases}
\]

Therefore, state transition matrix \( P \) is a reductive random matrix, then

\[
P^\infty = \lim_{t \to \infty} \left[ \begin{array}{c}
P_{11} & 0 \\
Q_2 & Q_1
\end{array} \right]^t = \lim_{t \to \infty} \left[ \begin{array}{c}
\prod_{i=1}^{t-1} P_{11} & 0 \\
\prod_{i=1}^{t-1} Q_2 & \prod_{i=1}^{t-1} Q_1
\end{array} \right]
\]

where \( Y_t \) is the optimal fitness value of population \( X(t) \), \( f^* \) represents the global optimal fitness value, and \( p(Y_t = f^*) \) indicates the probability that the optimal individual of the \( t \) generation is the global optimal value. It is proved that the NBBO algorithm has global convergence.

3. Simulation Experiments

3.1. Test Function Simulation. In order to test the optimization performance of the proposed IMBBO algorithm, 13 standard test functions were selected for simulation experiments [22, 23]. Table 1 shows the test function expression, the range of values for each independent variable, and the theoretical optimal value. We select the optimal value, average value, and standard deviation of the test results as the evaluation criteria to test the algorithm performance.

The experimental function of Table 1 is numerically tested using the proposed NBBO algorithm proposed in this paper, and the basic BBO algorithm, grey wolf optimization algorithm [24] (DE), elephant herding optimization [25] (EHO), salp swarm algorithm [26] (DEBBO), and particle swarm optimization biogeography optimization algorithm [27] (BBOPSO) for comparison of results. The experimental parameter settings of the algorithm are shown in Table 2.

To study the performance of the algorithm in different dimensions, the dimensions of test function \( f_1 - f_3 \) and \( f_5 - f_7 \) are \( D = 10, 30, 50 \), the dimension of test functions \( f_4 \) is \( D = 2 \), and the dimension of test functions \( f_8 \) is \( D = 4 \). In order to ensure that the comparison between each method is fair, we ensure that the experimental environment of each method is consistent during the experiment: operating system: window 10, memory: 8 GB, programming language: MATLAB 2016, and ensure that the setting of common parameters of the algorithm is consistent, and the parameters of different algorithms are set according to the parameters of specific references. As the algorithm is running randomly, the six algorithms independently run each test function 30 times to avoid unnecessary errors caused by the random operation of the algorithm.

Tables 3 and 4 show the recorded optimal function value, average value, and standard deviation obtained by the
algorithm. In the iterative process of the algorithm, we record the best fitness of each evaluation, find the average optimal value of each evaluation and draw a curve, which reflects the convergence trend of the algorithm. The convergence curves of some test functions are shown in Figures 2 and 3.

In order to compare the optimization performance of each algorithm more easily and clearly, the algorithm with the best optimization performance of the optimal value, average value and standard deviation of each test function is boldly marked in Tables 3 and 4. From Tables 3 and 4 and the simulation diagram, we find that the convergence speed of the analytical algorithm can be qualitatively and quantitatively analyzed. For the simple multidimensional unimodal function in Figure 2, the NBBO algorithm has faster convergence speed and solution precision in the later iteration although its advantage is not obvious in the initial stage of iteration. For the more complex multidimensional multimodal function in Figure 3, the other five algorithms have the phenomenon of search stagnation or slow search, i.e.,

```
Begin
Algorithmic parameters setting:
Population size = 50,
Maximum number of iterations G = 500,
Maximum immigration rate I = 1,
Maximum emigration rate E = 1,
Mutation probability Pm = 0.05,
Number of elites retained Keep = 2.
/* Initialization */
Random generation of a set of initial habitats constitutes the initial population.
Calculate the fitness value of each habitat in the population.
While not T /* T is the condition for the end of the iteration */
Sort the habitat in descending order according to the fitness value.
The immigration rate $\lambda_i$ and emigration rate $\mu_i$ of each habitat were calculated according to formula (1).
/* End of initialization */
/* Migration */
P_i
Select $H_j$ according to the immigration probability $\lambda_i$
for $j = 1: D$ /* D is dimension */
if rand < $\lambda_i$
  Select $H_j$ according to the migration probability $\mu_i$
  Randomly select a characteristic variable SIV from $H_j$
  Replace a random characteristic variable SIV in $H_j$ with SIV
else
  Perform a local search on the characteristic variable SIV
  according to formula (6) can get SIV'.
  Replace random characteristic variable SIV in $H_j$ with SIV'.
end if
end for
/* End of migration */
/* Mutation */
for $i = 1$: Size
  Calculate the probability of mutation $P_s$ according to $\lambda_i$ and $\mu_i$
  Selection of unmutated habitat $H_i$ based on $P_s$
  if rand(0,1) < $m_i$ then
    if rand < 0.5
      Replace $H_i$(SIV) with a randomly generated SIV by formula (8)
    else
      Replace $H_i$(SIV) with a randomly generated SIV by (9)
    end if
  end if
end for
/* End of mutation */
Recalculate the habitat fitness values
end while
end
```

Algorithm 1: The main procedure of NBBO.
they fall into local optimum and are difficult to jump out. Then, by comparing the optimal value of each test function and the average value of the optimization results, we can see that IMBBO algorithm has achieved a smaller value. Although the optimization results of DE, DEBBO, and BBOPSO algorithms also reach the global optimal value in the optimization of some test functions with different dimensions, it can be seen from the optimization results of single peak function and multi peak function that NBBO algorithm has a great advantage in the optimization ability. Figures 2 and 3 show that the proposed NBBO algorithm can quickly and effectively find the optimal solution under a short number of iterations, and its optimal solution has higher precision and stability.

Considering the fixed dimension test function has few dimensions to be adjusted, the accuracy and stability of each algorithm are relatively high. For the multidimensional test function, when its dimension \(D\) is increased from 10 to 30 or 50, the accuracy and stability of the algorithm have decreased. This is because the objective function is more complicated and the algorithm needs to make more adjustments. However, no matter how the dimension changes, the proposed NBBO algorithm still maintains a stable and high-precision optimization performance for high-dimensional complex functions. In reference [28], derrick proposed that for the evaluation of the performance of the improved evolutionary algorithm, only based on the average value and the standard deviation value to compare the advantages and disadvantages of the algorithm is not rigorous enough, so statistical test should be carried out. The statistical test shows that the improved algorithm has significant improvement advantages over other existing algorithms. Therefore, in order to verify the significant difference between the IMBBO algorithm and other algorithms, a nonparametric estimation method, Wilcoxon rank sum test, is used for statistical analysis. For each test function, the 30 times solution results of IMBBO algorithm are tested with those of BBO, DE, EHO, DEBBO, and BBOPSO algorithms respectively, and H0 is assumed: the solution results of the two algorithms are generally the same; H1: the solution
Table 3: Unimodal function test results.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x)$ Quartic</td>
<td>D = 10</td>
<td>Best</td>
<td>$3.40 \times 10^{-3}$</td>
<td>$1.70 \times 10^{-3}$</td>
<td>$2.10 \times 10^{-3}$</td>
<td>$1.81 \times 10^{-4}$</td>
<td>$7.92 \times 10^{-5}$</td>
<td>$5.78 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>Ave</td>
<td>$1.37 \times 10^{-2}$</td>
<td>$4.10 \times 10^{-3}$</td>
<td>$6.60 \times 10^{-3}$</td>
<td>$9.45 \times 10^{-4}$</td>
<td>$1.20 \times 10^{-3}$</td>
<td>$4.51 \times 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td>$9.80 \times 10^{-3}$</td>
<td>$1.70 \times 10^{-3}$</td>
<td>$3.70 \times 10^{-3}$</td>
<td>$6.91 \times 10^{-4}$</td>
<td>$8.62 \times 10^{-4}$</td>
<td>$3.25 \times 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td>$7.58 \times 10^{-2}$</td>
<td>$3.50 \times 10^{-2}$</td>
<td>$8.44 \times 10^{-2}$</td>
<td>$1.05 \times 10^{-2}$</td>
<td>$6.20 \times 10^{-3}$</td>
<td>$1.10 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ave</td>
<td>$1.47 \times 10^{-1}$</td>
<td>$5.46 \times 10^{-2}$</td>
<td>$1.33 \times 10^{-1}$</td>
<td>$1.46 \times 10^{-2}$</td>
<td>$2.23 \times 10^{-2}$</td>
<td>$2.70 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td>$4.41 \times 10^{-2}$</td>
<td>$1.03 \times 10^{-2}$</td>
<td>$4.13 \times 10^{-2}$</td>
<td>$4.20 \times 10^{-3}$</td>
<td>$9.10 \times 10^{-3}$</td>
<td>$2.00 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td>$2.51 \times 10^{-1}$</td>
<td>$1.21 \times 10^{-1}$</td>
<td>$1.98 \times 10^{-1}$</td>
<td>$3.47 \times 10^{-2}$</td>
<td>$4.63 \times 10^{-2}$</td>
<td>$3.00 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>D = 50</td>
<td>Ave</td>
<td>$5.67 \times 10^{-1}$</td>
<td>$2.29 \times 10^{-1}$</td>
<td>$5.37 \times 10^{-1}$</td>
<td>$5.61 \times 10^{-2}$</td>
<td>$7.82 \times 10^{-2}$</td>
<td>$8.50 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td>$2.08 \times 10^{-1}$</td>
<td>$5.16 \times 10^{-2}$</td>
<td>$1.38 \times 10^{-1}$</td>
<td>$1.56 \times 10^{-2}$</td>
<td>$2.22 \times 10^{-2}$</td>
<td>$4.20 \times 10^{-3}$</td>
<td></td>
</tr>
</tbody>
</table>

| $f_2(x)$ Sphere | D = 10 | Best | $1.94 \times 10^{0}$ | $1.18 \times 10^{-2}$ | $2.35 \times 10^{-4}$ | $6.72 \times 10^{-3}$ | $2.02 \times 10^{-2}$ | $3.66 \times 10^{-4}$ |
| | Ave | $8.38 \times 10^{0}$ | $3.52 \times 10^{-2}$ | $3.74 \times 10^{-4}$ | $1.69 \times 10^{-3}$ | $1.37 \times 10^{-2}$ | $5.79 \times 10^{-4}$ |
| | Std | $6.12 \times 10^{0}$ | $2.77 \times 10^{-2}$ | $6.88 \times 10^{-5}$ | $2.06 \times 10^{-3}$ | $1.49 \times 10^{-2}$ | $9.78 \times 10^{-4}$ |
| | Best | $7.13 \times 10^{0}$ | $1.20 \times 10^{-2}$ | $3.00 \times 10^{-3}$ | $1.40 \times 10^{-3}$ | $6.73 \times 10^{-4}$ | $2.09 \times 10^{-3}$ |
| | Ave | $1.43 \times 10^{0}$ | $2.60 \times 10^{-3}$ | $3.60 \times 10^{-3}$ | $5.83 \times 10^{-3}$ | $1.24 \times 10^{-3}$ | $1.43 \times 10^{-3}$ |
| | Std | $4.64 \times 10^{0}$ | $2.20 \times 10^{-3}$ | $3.01 \times 10^{-4}$ | $4.90 \times 10^{-4}$ | $3.35 \times 10^{-5}$ | $3.91 \times 10^{-5}$ |
| | Best | $5.93 \times 10^{0}$ | $1.58 \times 10^{0}$ | $7.90 \times 10^{-3}$ | $2.38 \times 10^{-5}$ | $4.11 \times 10^{-2}$ | $4.66 \times 10^{-3}$ |
| | Ave | $7.27 \times 10^{0}$ | $2.30 \times 10^{0}$ | $9.00 \times 10^{-3}$ | $1.62 \times 10^{-4}$ | $1.64 \times 10^{-1}$ | $6.83 \times 10^{-3}$ |
| | Std | $1.34 \times 10^{0}$ | $4.63 \times 10^{0}$ | $1.20 \times 10^{-3}$ | $1.54 \times 10^{-4}$ | $9.32 \times 10^{-2}$ | $8.48 \times 10^{-4}$ |

| $f_3(x)$ Powell | D = 10 | Best | $2.47 \times 10^{-2}$ | $2.04 \times 10^{-5}$ | $4.64 \times 10^{-6}$ | $3.89 \times 10^{-4}$ | $2.63 \times 10^{-4}$ | $8.88 \times 10^{-13}$ |
| | Ave | $3.01 \times 10^{-1}$ | $1.52 \times 10^{-4}$ | $7.51 \times 10^{-6}$ | $2.40 \times 10^{-3}$ | $1.45 \times 10^{-3}$ | $5.16 \times 10^{-7}$ |
| | Std | $2.54 \times 10^{-1}$ | $1.04 \times 10^{-4}$ | $2.22 \times 10^{-6}$ | $1.80 \times 10^{-3}$ | $1.24 \times 10^{-3}$ | $7.21 \times 10^{-7}$ |
| | Best | $2.47 \times 10^{0}$ | $6.19 \times 10^{-1}$ | $2.36 \times 10^{-4}$ | $1.21 \times 10^{-1}$ | $5.35 \times 10^{-2}$ | $1.13 \times 10^{-10}$ |
| | Ave | $5.77 \times 10^{-1}$ | $1.36 \times 10^{0}$ | $4.10 \times 10^{-4}$ | $3.92 \times 10^{-1}$ | $2.90 \times 10^{-1}$ | $1.26 \times 10^{-6}$ |
| | Std | $3.75 \times 10^{-1}$ | $4.70 \times 10^{-1}$ | $1.50 \times 10^{-4}$ | $1.61 \times 10^{-1}$ | $1.25 \times 10^{-1}$ | $2.10 \times 10^{-6}$ |
| | Best | $1.73 \times 10^{0}$ | $1.04 \times 10^{0}$ | $2.20 \times 10^{-3}$ | $2.02 \times 10^{0}$ | $1.63 \times 10^{0}$ | $4.07 \times 10^{-9}$ |
| | Ave | $5.16 \times 10^{-1}$ | $1.87 \times 10^{0}$ | $3.30 \times 10^{-3}$ | $4.55 \times 10^{0}$ | $2.55 \times 10^{0}$ | $4.69 \times 10^{-6}$ |
| | Std | $2.44 \times 10^{-1}$ | $4.83 \times 10^{0}$ | $6.48 \times 10^{-4}$ | $2.61 \times 10^{0}$ | $9.34 \times 10^{-1}$ | $6.33 \times 10^{-9}$ |

| $f_4(x)$ Easom | D = 2 | Best | $-9.13 \times 10^{-1}$ | $-1.00 \times 10^{0}$ | $-9.95 \times 10^{-1}$ | $-1.00 \times 10^{0}$ | $-1.00 \times 10^{0}$ | $-1.00 \times 10^{0}$ |
| | Ave | $-5.00 \times 10^{-1}$ | $-1.00 \times 10^{0}$ | $-9.60 \times 10^{-1}$ | $-1.00 \times 10^{0}$ | $-1.00 \times 10^{0}$ | $-1.00 \times 10^{0}$ |
| | Std | $4.45 \times 10^{-1}$ | $0.00 \times 10^{0}$ | $3.55 \times 10^{-2}$ | $0.00 \times 10^{0}$ | $0.00 \times 10^{0}$ | $0.00 \times 10^{0}$ |
|----------|-----------|----------------------|---------------|-----|--------------|----|---------------|-----|---------------|--------|--------------|--------|
| $f_1(x)$ | 2         | Best                 | 2.59E-02      | 1.24E-19 | 3.59E-06      | 4.09E-06 | 6.19E-07      | 1.58E-04 | 3.07E-03      | 4.26E-02 | 3.55E-04     | 1.03E-10 |
|          |           | Ave                  | 5.18E-02      | 7.12E-09 | 7.22E-06      | 6.62E-06 | 9.06E-06      | 8.89E-06 | 3.97E-03      | 4.36E-02 | 3.65E-03     | 1.05E-13 |
|          |           | Std                  | 2.26E-02      | 1.73E-07 | 5.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
| $f_2(x)$ | 3         | Best                 | 5.18E-02      | 1.24E-19 | 3.59E-06      | 4.09E-06 | 6.19E-07      | 1.58E-04 | 3.07E-03      | 4.26E-02 | 3.55E-04     | 1.03E-10 |
|          |           | Ave                  | 7.83E-02      | 1.11E-09 | 7.12E-06      | 6.62E-06 | 9.06E-06      | 8.89E-06 | 3.97E-03      | 4.36E-02 | 3.65E-03     | 1.05E-13 |
|          |           | Std                  | 3.32E-02      | 2.32E-07 | 5.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
| $f_3(x)$ | 10        | Best                 | 2.59E-02      | 1.24E-19 | 3.59E-06      | 4.09E-06 | 6.19E-07      | 1.58E-04 | 3.07E-03      | 4.26E-02 | 3.55E-04     | 1.03E-10 |
|          |           | Ave                  | 5.18E-02      | 1.73E-07 | 7.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
|          |           | Std                  | 2.26E-02      | 1.73E-07 | 5.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
| $f_4(x)$ | 30        | Best                 | 5.18E-02      | 1.24E-19 | 3.59E-06      | 4.09E-06 | 6.19E-07      | 1.58E-04 | 3.07E-03      | 4.26E-02 | 3.55E-04     | 1.03E-10 |
|          |           | Ave                  | 7.83E-02      | 1.11E-09 | 7.12E-06      | 6.62E-06 | 9.06E-06      | 8.89E-06 | 3.97E-03      | 4.36E-02 | 3.65E-03     | 1.05E-13 |
|          |           | Std                  | 3.32E-02      | 2.32E-07 | 5.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
| $f_5(x)$ | 50        | Best                 | 2.59E-02      | 1.24E-19 | 3.59E-06      | 4.09E-06 | 6.19E-07      | 1.58E-04 | 3.07E-03      | 4.26E-02 | 3.55E-04     | 1.03E-10 |
|          |           | Ave                  | 5.18E-02      | 1.73E-07 | 7.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
|          |           | Std                  | 2.26E-02      | 1.73E-07 | 5.22E-06      | 7.77E-06 | 1.75E-05      | 9.64E-09 | 4.43E-02      | 5.90E-02 | 9.64E-08     | 1.27E-16 |
Figure 2: Continued.
Figure 2: Continued.

- Reference [24]  - Reference [27]
- Reference [25]  - The proposed method

Figure 2: Continued.

- Reference [24]  - Reference [27]
- Reference [25]  - The proposed method

Figure 2: Continued.
results of the two algorithms are generally different. Set the threshold value of significance level of zero hypothesis test $\alpha = 0.05$. When the test result $P < 0.05$, reject the zero hypothesis; the experimental results of the two algorithms have significant difference; when the test result $P > 0.05$, accept the zero hypothesis; the experimental results of the two algorithms have no significant difference. The specific inspection results are shown in Tables 5 and 6.

Tables 5 and 6 show the test results of test function $f_1 - f_8$. Among them, for the probability $p$ value less than the significance level threshold value of 0.05, bold display; “W” is the significance test judgment result, combined with Tables 3 and 4, use “+” to indicate that NBBO is more accurate and significant than other algorithms, use “-” to indicate that NBBO is less accurate and significant than other algorithms, and use “=” to indicate that NBBO algorithm. There is no significant difference in the accuracy of other algorithms; “N/A” means that significance judgment cannot be made. From the test results, most of the test $p$ values are less than 0.05 and “W” is “+”, so zero hypothesis is rejected. Therefore, there are significant differences between the calculation results of IMBBO and BBO, DE, EHO, DEBBO, and BBOPSO, and NBBO is significantly better.

To sum up, the biogeography-based optimization algorithm based on local search and nonuniform variation has good results for most of the benchmark functions, which effectively solves the slow convergence speed of the original BBO algorithm and easy to fall into the local optimization problem in the later stage of the search and greatly improves the convergence speed and search accuracy of the BBO algorithm without adding too many programming steps.

3.2 Iris Classification Based on NBBO Multilayer Perceptron.

In order to test the training performance of NBBO multilayer perceptron for iris classification, the mean square error (MSE) of the actual output value of multilayer perceptron and the expected output value is used as an indicator to measure the performance of multilayer perceptron [29]. MSE of all training samples is calculated as follows:

$$
MSE = \frac{1}{s} \sum_{k=1}^{s} \sum_{i=1}^{m} (o_{ik}^e - o_{ik}^o)^2,
$$

where $s$ is the number of training samples, $m$ is the number of outputs, $o_{ik}^e$ is the expected output of the $i$th input unit under $k$ training samples, and $o_{ik}^o$ is the actual output of the $i$th input unit under $k$ training samples. The MSE value is smaller is equal to the performance of the multilayer perceptron is better, and vice versa. The HIS of the $i$th habitat is calculated by

$$
\text{HSI}(\text{Habitat}_i) = E(\text{Habitat}_i).
$$

Figure 4 shows the schematic diagram of a multilayer perceptron based on NBBO algorithm. NBBO algorithm receives the average MSE of all training samples and all expected output samples as the objective function for iterative training and adjusts the link weights and deviations.
The proposed method

Figure 3: Continued.
Figure 3: Continued.
Figure 3: Multimodal function simulation results diagram. (a) $f_5$: Alpine ($d = 10$), (b) $f_5$: Alpine ($d = 30$), (c) $f_5$: Alpine ($d = 50$), (d) $f_5$: Griewank ($d = 10$), (e) $f_5$: Griewank ($d = 30$), (f) $f_5$: Griewank ($d = 50$), (g) $f_5$: Ackley ($d = 10$), (h) $f_5$: Ackley ($d = 30$), (i) $f_5$: Ackley ($d = 50$), (j) $f_5$: Kowalik ($d = 4$).

Table 5: Wilcoxon rank sum test results of unimodal function.

<table>
<thead>
<tr>
<th>Function</th>
<th>Dimension</th>
<th>NBBO vs BBO</th>
<th>NBBO vs DE</th>
<th>NBBO vs EHO</th>
<th>NBBO vs DEBBO</th>
<th>NBBO vs BBOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x)$</td>
<td>$D = 10$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
</tr>
<tr>
<td></td>
<td>$D = 30$</td>
<td>$2.82e-11$</td>
<td>$2.82e-11$</td>
<td>$2.82e-11$</td>
<td>$2.82e-11$</td>
<td>$2.82e-11$</td>
</tr>
<tr>
<td></td>
<td>$D = 50$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
</tr>
<tr>
<td>$f_2(x)$</td>
<td>$D = 10$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
</tr>
<tr>
<td></td>
<td>$D = 30$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
</tr>
<tr>
<td></td>
<td>$D = 50$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
<td>$2.87e-11$</td>
</tr>
<tr>
<td>$f_3(x)$</td>
<td>$D = 10$</td>
<td>$1.17e-12$</td>
<td>N/A</td>
<td>N/A</td>
<td>$1.17e-12$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$D = 30$</td>
<td>$2.87e-11$</td>
<td>N/A</td>
<td>N/A</td>
<td>$2.87e-11$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$D = 50$</td>
<td>$2.87e-11$</td>
<td>N/A</td>
<td>N/A</td>
<td>$2.87e-11$</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 6: Wilcoxon rank sum test results of multimodal function.

<table>
<thead>
<tr>
<th>Function</th>
<th>Dimension</th>
<th>NBBO vs BBO</th>
<th>NBBO vs DE</th>
<th>NBBO vs EHO</th>
<th>NBBO vs DEBBO</th>
<th>NBBO vs BBOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_5(x)$</td>
<td>$D = 10$</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
</tr>
<tr>
<td></td>
<td>$D = 30$</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
</tr>
<tr>
<td></td>
<td>$D = 50$</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
<td>2.87e-11</td>
</tr>
<tr>
<td>$f_6(x)$</td>
<td>$D = 10$</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
</tr>
<tr>
<td></td>
<td>$D = 30$</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
</tr>
<tr>
<td></td>
<td>$D = 50$</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
</tr>
<tr>
<td>$f_7(x)$</td>
<td>$D = 10$</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.03e-13</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$D = 30$</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.03e-13</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$D = 50$</td>
<td>1.17e-12</td>
<td>1.17e-12</td>
<td>1.03e-13</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>$f_8(x)$</td>
<td>$D = 2$</td>
<td>2.87e-11</td>
<td>4.27e-07</td>
<td>1.68e-06</td>
<td>2.31e-08</td>
<td>2.87e-11</td>
</tr>
</tbody>
</table>
through continuous iterative evolution, so as to provide the optimal connection weight and deviation for iris classification samples after iterative training.

The iris dataset has 150 training samples divided into 3 categories: Setosa, Versicolor, and Virginica, with a total of 4 basic features: sepal length, sepal width, petal length, and petal length.

### Table 7: Simulation results of iris classification.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Optimal value</th>
<th>Average value</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference [4]</td>
<td>$3.15 \times 10^{-2}$</td>
<td>$5.34 \times 10^{-2}$</td>
<td>$1.52 \times 10^{-2}$</td>
</tr>
<tr>
<td>Reference [24]</td>
<td>$6.12 \times 10^{-2}$</td>
<td>$1.31 \times 10^{-1}$</td>
<td>$3.48 \times 10^{-2}$</td>
</tr>
<tr>
<td>Reference [25]</td>
<td>$2.37 \times 10^{-1}$</td>
<td>$3.92 \times 10^{-1}$</td>
<td>$7.36 \times 10^{-2}$</td>
</tr>
<tr>
<td>Reference [26]</td>
<td>$2.73 \times 10^{-2}$</td>
<td>$4.72 \times 10^{-2}$</td>
<td>$1.29 \times 10^{-2}$</td>
</tr>
<tr>
<td>Reference [27]</td>
<td>$1.51 \times 10^{-1}$</td>
<td>$2.56 \times 10^{-1}$</td>
<td>$5.34 \times 10^{-2}$</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$1.34 \times 10^{-2}$</td>
<td>$1.93 \times 10^{-2}$</td>
<td>$3.80 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

### Figure 4: NBBO-based Iris classification diagram.

### Figure 5: (a) Convergence curve and (b) classification accuracy.
width. We used the structure of multilayer perceptron as 4-9-3 to classify the problem. Table 7 shows the experimental results of the data set algorithm. Figure 5 shows the average convergence trend and classification accuracy.

Compared with other five iris classification algorithms in Table 7, we can see that the proposed NBBO has the better optimal value, average value, and standard deviation than the other five algorithms. The optimal and average values of the MSE calculated by NBBO are better than other algorithms, which indicates that the optimization performance of the NBBO is the best among the seven algorithms. At the same time, the minimum standard deviation calculated by NBBO also indicates that the NBBO algorithm inherits the strong robustness of the BBO algorithm and is more robust than other algorithms.

Figure 5(a) shows the classification convergence curves of the six algorithms for the iris classification problem, among which the convergence effect of the DE algorithm is the worst of the six algorithms, and the EHO and BBOPSO algorithms fall into the local optimum in the late iteration. BBO algorithm and DEBBO algorithm have similar convergence precision but the convergence speed is not ideal. The proposed NBBO algorithm is optimal in convergence precision and convergence speed. Figure 5(b) shows that DE, EHO, and BBOPSO show significant fluctuations in the classification accuracy of 10 independent operations. The NBBO algorithm has the best stability and average, and its classification accuracy is the highest among the three stable algorithms BBO, DEBBO, and NBBO.

4. Conclusions and Future Work

This paper proposed a novel biogeography-based optimization algorithm based on local migration strategy and non-uniform mutation for iris classification. Firstly, the hyperbolic cotangent transfer model was used to replace the original transfer model to enhance the adaptability of the algorithm. Secondly, the local transfer strategy was used to improve the global search ability of the algorithm. Finally, the nonuniform mutation operation was used to enhance the development ability of the algorithm in the later stage of iteration for the optimal solution in a small range, to improve the ability of the algorithm to exit the local optimum. The simulation results showed that the proposed NBBO has higher stability, better convergence accuracy, and faster convergence speed than other method for iris classification. How to use NBBO algorithm to solve other multiobjective optimization problem compare with multiobjective optimization method [30–33] and realize better applications for engineering practice will be the focus of our next work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Research Programme of Colleges and Universities of Anhui Province under grant KJ2020ZD39, the Foundation for Talented Young People of Anhui Polytechnic University under grant 2016BJRC008, the Open Research Fund of Anhui Key Laboratory of Detection Technology and Energy Saving Devices, and Anhui Polytechnic University under grant DTESD2020A02.

References


