Gradient Eigen-decomposition Invariance Biogeography-based Optimization for Mobile Robot Path Planning

Jiaqian Wang
Dalian University of Technology Faculty of Electronic Information and Electrical Engineering
Xiaodong Na
Dalian University of Technology Faculty of Electronic Information and Electrical Engineering
Min Han (minhan@dlut.edu.cn)
Dalian University of Technology https://orcid.org/0000-0002-2964-4884
Deicai Li
Shenyang Institute of Automation Chinese Academy of Sciences

Research Article

Keywords: Mobile robot path planning, Biogeography-based optimization, Eigen-decomposition, Gradient decent strategy, System search strategy

Posted Date: September 27th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-885939/v1

Gradient Eigen-decomposition Invariance Biogeography-based Optimization for Mobile Robot Path Planning

Jiaqian Wang[1], Xiaodong Na[1], Min Han[2,3*] and Decai Li[4]

[1]Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, 116024, China.
[2*]Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian, 116024, China.
[3*]Professional Technology Innovation Center of Distributed Control for Industrial Equipment of Liaoning Province, Dalian University of Technology, Dalian, 116024, China.
[4]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China.

*Corresponding author(s). E-mail(s): minhan@dlut.edu.cn; Contributing authors: wangjiaqian@mail.dlut.edu.cn; naxiaodong@mail.dlut.edu.cn; lidecai@sia.cn;

Abstract

The path planning for mobile robots has attracted extensive attention, and evolutionary algorithms have been applied to this problem increasingly. In this paper, we propose a novel gradient eigen-decomposition invariance biogeography-based optimization (GEI-BBO) for mobile robot path planning, which has the merits of high rotation invariance and excellent search performance. In GEI-BBO, we design an eigen-decomposition mechanism for migration operation, which can reduce the dependence of biogeography-based optimization (BBO) on the coordinate system, improve the rotation invariance and share the information between eigen solutions more effectively. Meanwhile, to find the local optimal solution better, gradient descent is added, and the system search strategy can reduce the occurrence of local trapping phenomenon.

In addition, combining the GEI-BBO with cubic spline interpolation will solve the problem of mobile robot path planning through a defined coding method and fitness function. A series of experiments are implemented on benchmark functions, whose results indicated that the optimization performance of GEI-BBO is superior to other algorithms. And the successful application of GEI-BBO for path planning in different environments confirms its effectiveness and practicability.

Keywords: Mobile robot path planning, Biogeography-based optimization, Eigen-decomposition, Gradient decent strategy, System search strategy

1 Introduction

In recent decades, mobile robot path planning (MRPP) has become an indispensable aspect of artificial intelligence in robotics, and has been widely studied and discussed [1]. The goal of path planning is to search out an optimal or nearly optimal collision free path from the initial state to the target state according to a certain performance index (such as time, distance, etc.) [2]. According to different environmental information, path planning can be divided into global path planning and local path planning [3]. Global path planning is carried out in the known environment. By modeling the current environmental information, mobile robots can use the existing mature path planning algorithm or improved algorithm to plan an optimal path from the start to the end in the established environmental model [4]. In the local path planning problem, the environmental information is usually unknown, focusing on considering the current local environmental information of the mobile robot to make the robot have good obstacle avoidance ability. The working environment of the mobile robot is detected by sensors to obtain information such as the location and several properties of obstacles [5]. However, without complete environmental information, the results of local path planning may not be best or even incorrect. Therefore, global path planning, which can be regarded as an optimization problem, has attracted more attention and become a hot topic in research of path planning [6].

Researchers have developed many methods for global MRPP. Traditional path planning methods include grid method [4], artificial potential field method [7], visual graph method [3], neural network method [8], etc. However, these methods partly have some problems such as complex calculation, low efficiency, easy to fall into local optimization, which usually leads to certain inconveniences in solving the problem of global MRPP. Meanwhile, evolutionary algorithms for MRPP show a growing number of advantages and attract increasing attention [9].

Using the evolutionary algorithm for global MRPP, which can put the problem of path planning down to the problem of finding the optimal path with the minimum cost [10]. In addition, many evolutionary algorithms, such as genetic algorithm (GA) [11], particle swarm optimization algorithm (PSO) [12] and ant colony algorithm (ACO) [13], have strong computing capability and robustness, which can achieve optimization results well. At present, many evolutionary algorithms have been applied to the problem of MRPP [14]. For example, considering the slow convergence speed of ACO, Liu et al. proposed an improved ACO to improve its convergence speed and applied it to the MRPP in the grid environment [15]. Marco et al. combined the artificial bee colony algorithm with the evolutionary planning algorithm to solve the problem of MRPP, through a set of local processes to refine the feasible path [16]. The comparison results of the method on a set of benchmark problems and the experimental results on a real mobile robot show that the method has good performance. Hong et al. proposed a co-evolutionary improved GA for global MRPP, which puts forward an effective fitness function and modifies the genetic operator of traditional GA [17]. For the problem of global smooth MRPP, song et al. advanced a new multimodal delayed PSO (MDPSO) [18]. The test results based on benchmark function show that the performance of MDPSO is better than other five famous PSO algorithms. Finally, the application of MDPSO in the global smooth MRPP further proves that it has better performance than the global smooth path generated by GA in previous studies. These evolutionary algorithms have been used to solve the problem of MRPP, and have achieved remarkable results, but in general, these methods still have some shortcomings in computational complexity, local optimization and adaptability.

Biogeography-based optimization (BBO) [19] is an evolutionary algorithm based on the concept of biogeography. Based on a mathematical model, the algorithm describes the migration of species between habitats, that is, from unsuitable habitats to suitable habitats [20]. BBO is an efficient bionic search algorithm, which has the advantages of relatively simple principle, less adjustment parameters, easier implementation of the algorithm, and high operation efficiency. Many researchers have improved BBO and applied it to many fields [21], including MRPP. For example, Zhu et al. proposed a method based on chaotic predator-prey BBO (CPPBBO) to solve the path planning problem of unmanned aerial vehicle [22], and the simulation results show that CPPBBO is more effective than other algorithms. Mo et al. advanced a biogeography PSO algorithm to plan the path of mobile robot by combining the BBO and PSO, then using the BPSO algorithm, the optimal path based on the approved voronoi boundary network is found out [23]. Yang et al. put forward an improved optimization algorithm based on biogeography to solve the problem of global MRPP in the static environment [24]. It can be seen from these articles that BBO has been effectively applied to MRPP. However, due to the strong dependence of BBO on the coordinate system, the performance of the algorithm is inadequate when solving high-dimensional problems, and the ability of BBO to mine the global optimal solution is also slightly insufficient.

In order to solve these problems, we proposed a novel gradient eigen-decomposition invariance biogeography-based optimization algorithm (GEI-BBO). The innovation of this paper is as follows:

1. A novel biogeography-based optimization migration eigen-decomposition based migration strategy which shares information between eigen solutions more effectively is proposed based on eigen-decomposition. The egien-decomposition based migration reduces the dependence on the coordinate and improves the rotation invariance of BBO.
2. On the basis of the egien-decomposition based migration strategy, the gradient descent strategy is added to search the neighborhood of the best individual, which can more effectively mine and not only search the optimal solution, so as to improve the local search ability of BBO.
3. The system search strategy is put forward to ensure that the algorithm with gradient descent strategy finds the global optimal solution, which can carry out a range search for each dimension, so as to cover the whole search range well and avoid falling into local optimum.
4. GEI-BBO is combined with the cubic spline interpolation method to solve the problem of MRPP. This paper defines the coding method based on the path node and constructs fitness function which aims at avoiding the obstacle and finding the shortest path.

The rest of the paper are organized as follows. The second section introduces the basic BBO. The third section introduces gradient eigen-decomposition invariance biogeography-based optimization. The fourth section is the method for path planning. The fifth section is the simulation results and analysis. The sixth section is the conclusion. The seventh section is the acknowledgement. The eighth section is references.

2 Biogeography-based optimization

BBO is a new population-based optimization algorithm to solve global optimization problems, based on the concept of biogeography. In biogeography, each habitat is regarded as an individual, and the index to measure its quality of life is called habitat suitability index (HSI) [23]. Good habitats for species have high HSI and bad habitats have low HSI. A habitat has many characteristics, such as area, temperature, rainfall and so on. These characteristics will affect the habitat, so they are referred to as the suitability index variables (SIVs). In BBO, a candidate solution to the problem is considered as a habitat. Each candidate solution is associated with a fitness value that is similar to the HSI of the habitat. High HSI habitats represent better solutions, and low HSI habitats represent worse solutions. Immigration rates and emigration rates for each candidate solution are used to share features probabilistically between habitats.

In BBO, the rate of immigration λ and the rate of emigration μ determine the dynamic movement between habitats, depending on the number of species

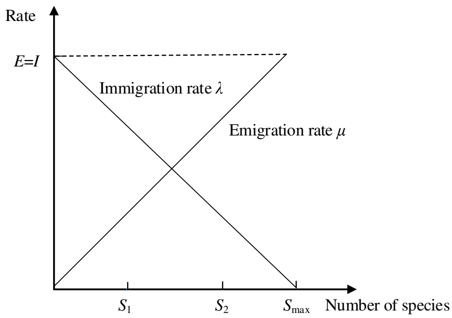


Fig. 1: Species model of the habitat

in the habitat [25]. The equation for the rate of immigration λ_k and the rate of emigration μ_k of k species can be written as follows:

$$\lambda_k = I(1 - \frac{k}{n}) \tag{1}$$

$$\mu_k = E(\frac{k}{n}) \tag{2}$$

where I and E are the maximum immigration rate and the maximum emigration rate, respectively. $n = s_{\max}$ is the maximum number of species in the habitat. If $I = E$, then $\lambda_{\mathrm{k}} + \mu_k = I$.

BBO is different from other evolutionary algorithms because of its migration strategy. The migration strategy is based on the migration model, that is, the immigration rate and the emigration rate equation. The migration model generally adopts the linear model. When a habitat X_i needs to be relocated according to the migration rate, some methods, such as roulette, will be used to select a source habitat X_j probabilistically according to the migration rate, and then a SIV will be selected randomly in the source habitat to replace directly or modify X_i. BBO can improve the solution through the migration operation.

The mutation operation of BBO is to randomly mutate the habitat according to the mutation rate $m(s)$, that is, changing the SIVs of the habitat. $m(s)$ is determined by the following equation:

$$m(s) = m_{max}(\frac{1 - P_s}{P_{max}}) \tag{3}$$

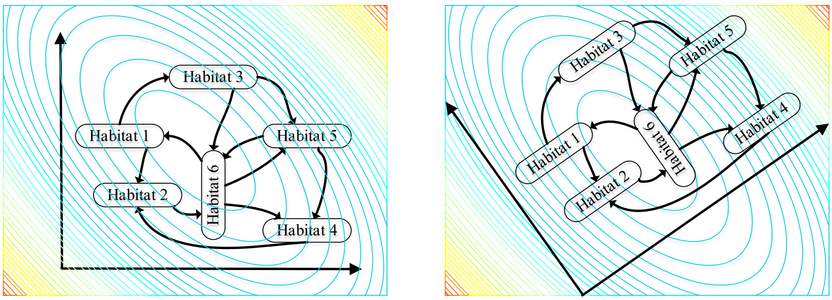
where $m(s)$ is the mutation rate of habitat, m_{max} is the maximum mutation rate, P_{max} is the maximum probability. BBO can increase the diversity of species through mutation operation [26].

3 Gradient eigen-decomposition invariance biogeography-based optimization

In this section, we introduce a method to solve the problem of two-dimensional static global MRPP. We introduce three innovative points of the proposed algorithm, namely, the eigen-decomposition based migration, gradient descent and system search strategy. Finally, the algorithm is described.

3.1 Eigen-decomposition based migration

In order to improve the rotation invariance of BBO algorithm, the eigen-decomposition based migration is proposed in this paper. The core of the eigen-decomposition based migration is to rotate the original coordinate system into the eigenvector-based coordinate system, in which habitants can share their information more effectively. The proposed method illustrates that the migration of BBO can be carried out more effectively in the eigenvector-based coordinate system. The eigen-decomposition based migration is to carry out the eigen-decomposition of population, and migrate the decomposed population, which will make BBO run more effectively.



(a) Original migration (b) Eigen-decomposition based migration

Fig. 2: The core of the eigen-decomposition based migration

Firstly, suppose a population H

$$\left.\begin{array}{l} H = (H_1^G, ..., H_{\rm i}^G, ..., H_{\rm n}^G)^T \\ H_{\rm i}^G = (H_{\rm (i.1)}^G, ..., H_{\rm (i.j)}^G, ..., H_{\rm (i.D)}^G) \end{array}\right\} i = 1,...,n;\ j = 1,...,D \tag{4}$$

where n is the population size, D is the number of independent variables, G is the number of iterations, H_i^G is the ith population in the Gth iteration.

H is not a square matrix of order n, it cannot be eigen-decomposed directly, so we introduce the concept of covariance matrix.

Make:

$$H^G_{\mathrm{n}\times D} = \begin{bmatrix} H^G_{(1,1)} & H^G_{(1,2)} & \cdots & H^G_{(1,D)} \\ H^G_{(2,1)} & H^G_{(2,2)} & \cdots & H^G_{(2,D)} \\ \vdots & \vdots & \ddots & \vdots \\ H^G_{(\mathrm{n},1)} & H^G_{(n,1)} & \cdots & H^G_{(n,D)} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \cdots & c_D \end{bmatrix} \tag{5}$$

Then the covariance matrix is:

$$cov(H) = \frac{1}{n{-}1} \begin{bmatrix} cov(c_1,c_1) & cov(c_1,c_2) & \cdots & cov(c_1,c_D) \\ cov(c_2,c_1) & cov(c_2,c_2) & \cdots & cov(c_2,c_D) \\ \vdots & \vdots & \ddots & \vdots \\ cov(c_D,c_1) & cov(c_D,c_2) & \cdots & cov(c_D,c_D) \end{bmatrix} \tag{6}$$

where $cov(c_i, c_j)$ is the covariance between the ith independent variable and the jth independent variable in Gth generation, which is defined as follows:

$$cov(c_i, c_j) = E[(c_i - \xi_i)(c_j - \xi_j)] \tag{7}$$

where ξ_i and ξ_j represents the mean values of the ith and jth independent variables, respectively.

Thus it can be seen that covariance matrix $cov(H)$ is a symmetric matrix of order n , and the following standard form can be obtained by eigen-decomposition:

$$cov(H) = Q_H \Lambda_H {Q_H}^T \tag{8}$$

where Q_H is a $D \times D$ eigen matrix composed of feature vector of $cov(H)$, Λ_H is a diagonal matrix composed of all the eigenvalues of $cov(H)$. After eigen-decomposition, the eigenvector is obtained. We also need to rotate the matrix H with the eigenvectors. That is:

$$\left.\begin{aligned} eigH^G_i &= H^G_i \times Q_H \\ eigH^G_i &= (eigH^G_{(i,1)}, ..., eigH^G_{(i,\mathrm{j})}, ..., eigH^G_{(i,D)}) \end{aligned}\right\} i = 1, ..., n;\ j = 1, ...D \tag{9}$$

Migration of the $eigH^G_i$ obtained by the rotation operation results in the generation of $eigH^{G+1}_i$. Because of the eigen-decomposition and rotation operation, the migration can run more efficiently. After the migration operation, H should be rotated back and used as the population of the new generation. The following operation is enough.

$$H^{G+1}_i{=}eigH^{G+1}_i \times Q_H \tag{10}$$

In summary, the steps of the migration algorithm based on eigen-decomposition are as follows:

Algorithm 1 Eigen-decomposition based migration

Require: H_i^G
Ensure: H_i^{G+1}
for $i = 1$ to n **do**
  Compute the matrix $cov(H)$
  Apply Eigen-decomposition according to Equation 8
  Rotate according to Equation 9
  for $k = 1$ to D **do**
    if $rand < \lambda_i$ **then**
      select $eigH_j^G$ with probability μ_j
      $eigH_{(i,k)}^G \leftarrow eigH_{(i,k)}^G + \alpha \cdot eigH_{(j,k)}^G$
      $eigH_{(i,k)}^{G+1} \leftarrow eigH_{(i,k)}^G$
    end if
  end for
  Get H_i^{G+1} according to Equation 10
end for

3.2 Gradient decent strategy

BBO algorithm is effective in searching the global optimal region, but it is difficult in mining the global optimal region. The local search performance of BBO can be improved by adding gradient descent into BBO to search the best individual domain.

The following conditions are required to activate the local search of gradient descent:

1) N_G is a positive integer.

2) gradient flag G_{flag} is equal to G_A, G_A is a predefined value.

If both of these conditions are met, then we apply gradient descent strategy to N_G optimal individuals. G_{flag} starts at zero and increases by 1 when the following conditions are met.

$$R_I = \frac{f_{\min}(it) - f_{\min}(it+1)}{f_{\min}(it)} \leq \varepsilon_1 \tag{11}$$

where R_I is the improvement ratio, which is the relative improvement of the minimum generation value from the $i(t)$ iteration to the $i(t+1)$ iteration. ε_1 is a predefined threshold. Moreover, by adding G_A through G_{INC}, the incremental G_A delays the next call to gradient descent local search. The reason why the gradient is called infrequently is that when the algorithm falls into a local optimal state, multiple local searches may not be helpful, and the function evaluation times will be wasted.

In short, gradient descent is activated when R_I is continuously smaller than ε_1. We use the $fmincon$ function in MATLAB to achieve the gradient descent of the best individual. The gradient descent algorithm is shown as follows:

Algorithm 2 Gradient descent

Require: $H_i^G, i = 1 : N_G$
Ensure: $H_i^{G+1}, i = 1 : N_G$
Compute improvement ratio $R_I < (f_{\min}(it) - f_{\min}(it+1))/f_{\min}(it)$
if $R_I < \varepsilon_1$ **then**
    $G_{flag} = G_{flag} + 1$
else
    $G_{flag} = 0$
end if
if $N_G > 0$ and $G_{flag} = G_A$ **then**
    Apply gradient descent to the N_G best individuals in the population
    $G_{flag} = 0$
    $G_A = G_{flag} + G_I$
end if

3.3 System search strategy

In order to ensure that the algorithm with gradient descent strategy can find the global optimal solution, we add the system search strategy, which can cover the whole search range well and avoid falling into the local optimal.

The conditions for activating the system search strategy are similar to those for gradient descent in 3.2, and the improved ratio R_I is also used. When the following formula is satisfied, the system search strategy will be activated.

$$R_I = \frac{f_{\min}(it) - f_{\min}(it+1)}{f_{\min}(it)} \leq \varepsilon \tag{12}$$

where ε is the computer precision, that is, when there is no improvement in the minimum cost value from the $i(t)$ iteration to the $i(t+1)$ iteration, a global search will be conducted for Ns optimal individuals to execute the system search strategy. The steps of the system search strategy are as algorithm 3.

As shown in the table, we increment or decrement the independent variables in accordance with Δ in the search space, where α_0 is a specific score that can be evaluated as $\alpha_0 = 0.1$. The system search strategy decreases the value of the given dimension $pop_{(i,k)}$ by an increment Δ, equal to 10% of the size of the search space, one increment at a time, until the value reaches the lower bound of the search space. Then, the system search strategy increases the value of the given dimension, one increment at a time, until it reaches the upper limit of the search space. By performing this process for each dimension, the system search strategy realizes the global search, and replaces the original individual with the best one found by it to avoid falling into the local optimization.

3.4 Description of the GEI-BBO algorithm

The GEI-BBO algorithm proposed in this paper includes the contents mentioned in section 3.1, 3.2 and 3.3 above. Before the number of iterations reaches

Algorithm 3 System search strategy

Require: $H_i^G, i = 1 : N_S$
Ensure: $H_i^{G+1}, i = 1 : N_S$

if $R_I < \varepsilon$ **then**
  $\Delta = \alpha_0(UB - LB)$
  for $i = 1$ to N_s **do**
    for $k = 1$ to D **do**
      $temp_{(i,k)} = pop_{(i,k)}$
      while $temp(i,k) \geq LB$ **do**
        $temp(i,k) \leftarrow temp(i,k) - \Delta$
        if $f(temp_{(i,k)}) < f(pop_{(i,k)})$ **then**
          $pop_{(i,k)} = temp_{(i,k)}$
        end if
      end while
      $temp_{(i,k)} = pop_{(i,k)}$
      while $temp(i,k) \leq UB$ **do**
        $temp(i,k) \leftarrow temp(i,k) + \Delta$
        if $f(temp_{(i,k)}) < f(pop_{(i,k)})$ **then**
          $pop_{(i,k)} = temp_{(i,k)}$
        end if
      end while
    end for
  end for
end if

the maximum number of iterations m, n habitats are selected to migrate according to the proportional parameter P in each iteration. If the random number is less than P, the migration is based on the eigen-decomposition, and if the random number is greater than P, the migration is based on the standard BBO. After migration, the mutation was carried out. Then the gradient descent strategy is implemented on the N_G best individuals, and the system search strategy is implemented on the N_s best individuals. N_G and N_s are preset parameters. The specific algorithm flow is as follows:

4 Method for path planning

Combining the improved algorithm with cubic spline interpolation method, the coding method based on the path node is defined, and the method and fitness function aiming at solving the obstacle avoidance and shortest path of mobile robot are constructed to solve the problem of MRPP. Fig.3 shows a conceptual map of habitat migration in MRPP using GEI-BBO to understand how the proposed approach works.

In the figure, habitat 1 has the highest HSI, representing the most suitable habitat, followed by habitat 2, habitat 3 and habitat 4. The higher the HSI, the more suitable for species growth, the less need to be changed, so the lower the

Algorithm 4 GEI-BBO

Initialize a population of N habitats
for $it = 1$ to M **do**
  for $t = 1$ to N **do**
    if **then**$rand < P$
      Apply Eigen-decomposition based migration (3.1)
    else
      Apply standard BBO migration
      Apply mutation operation according to mutation rate $m(s)$
    end if
    Apply gradient descent to the N_G optimal individuals in the population (3.2)
    Apply system search strategy to the N_S optimal individuals in the population (3.3)
  end for
end for

immigration rate, the higher the emigration rate. It can be seen that habitat 1 has the highest emigration rate, while habitat 4 has the highest emigration rate. Residents represent the path nodes. So the fourth habitat accepts many residents (path nodes) from other habitats, as shown in different colors. Purple nodes and links also describe mutations that occur in all habitats, regardless of their HSI values. This example shows how path planning evolved using the proposed approach. The specific coding method and fitness function are shown below.

4.1 Cubic spline interpolation

Cubic spline interpolation is a kind of piecewise interpolation method to form a smooth curve through a train of interpolation point intervals based on cubic polynomials. The curve of moving path of mobile robot fitted by cubic spline interpolation method is smoother than the curve fitted by straight line and circular arc. In this paper, the cubic spline interpolation method is integrated into the improved GEI-BBO algorithm to solve the problem of static global MRPP.

On the interval $[a, b]$, taking $n + 1$ nodes $a = x_0 < x_1 < ... < x_n = b$, if $s(x)$ satisfies following conditions:

1) $s(x) \in C^2[a, b]$.

2) On each small interval $[x_i, x_{i+1}]$, $s(x)$ is a cubic polynomial.

3) On node x_i, given the function value $f_i = f(x_i), i = 0, 1, ..., n$ and $s(x_i) = f_i, i = 0, 1, ..., n$. Then $s(x)$ is a cubic spline interpolation function.

The cubic spline interpolation function is a piecewise cubic polynomial, on each small interval $[x_i, x_{i+1}]$, which can be written as: $s(x) = a_i x^3 + b_i x^2 + c_i x + d_i, i = 0, 1, ..., n - 1$, where a_i, b_i, c_i, d_i is the undetermined coefficient, so s(x) has $4n$ undetermined coefficients. To solve for s of x, we need $4n$ conditions.

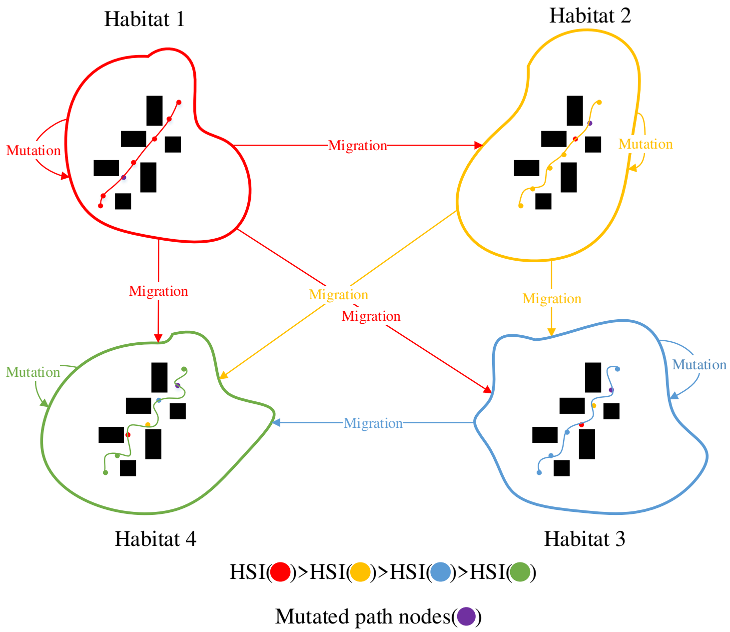


Fig. 3: Conceptual model of migration between the habitats for MRPP

From $s(x_i) = f_i, i = 0, 1, ..., n$, $n + 1$ interpolation conditions can be obtained. And from $s(x) \in C^2[a, b]$, $s(x)$ is second differentiable on the interval $[a, b]$, so the first order is differentiable and $s(x)$ is continuous. Therefore, the following conditions can be obtained

$$s_-^{''}(x_i) = s_+^{''}(x_i), i = 0, 1, ..., n - 1 \tag{13}$$

$$s_-^{'}(x_i) = s_+^{'}(x_i), i = 0, 1, ..., n - 1 \tag{14}$$

$$s_-(x_i) = s_+(x_i), i = 0, 1, ..., n - 1 \tag{15}$$

So far, there are $4n - 2$ conditions. In the actual calculation, two boundary conditions need to be introduced to calculate $s(x)$. Commonly used boundary conditions are:

1) The value of the first derivative at the two endpoints is given.
2) The value of the second derivative at the two endpoints is given.
3) $s(x)$ is a function of period $b - a$.

4.2 Encoding

It can be seen from above that cubic spline interpolation is a piecewise interpolation method, and the junction between segments is called the path node. The splines between segments are different, and the whole spline curve is continuous in the first order, and is continuous in the second order at the path node. Therefore, the directions between segmented splines may be different, and the path node represents the maximum turning times of the entire path. According to this, this paper takes all nodes on a path as the code of a habitat individual, that is, a habitat individual represents all nodes on the corresponding path. The x-coordinate set of all m path nodes on a path constitutes the m-dimensional x-coordinate of habitat individuals, and correspondingly, the y-coordinate set of all m path nodes on a path constitutes the m-dimensional y-coordinate of habitat individuals.

Suppose we know the coordinates of m path node $(x_{m1}, y_{m1}), (x_{m2}, y_{m2}), ..., (x_{mm}, y_{mm})$, starting point coordinates (x_s, y_s) and terminal coordinates (x_t, y_t), on the interval $(x_s, x_{m1}, x_{m2}, ..., x_{mm}, x_t)$ and $(y_s, y_{m1}, y_{m2}, ..., y_{mm}, y_t)$, the abscissa and ordinate of n interpolation points are obtained by cubic spline interpolation. In this way, we get n interpolation points, and the line between the path node and the interpolation point and the starting point and the ending point is the path of the mobile robot that we want.

4.3 Fitness function

There are two conditions for global MRPP: 1) Do not collide with obstacles; 2) The path length should be as short as possible. In this paper, the evaluation criterion of fitness function is the shortest path length satisfying the above conditions. In this paper, the constructed fitness function is:

$$f = L \cdot (1 + \beta \cdot v) \tag{16}$$

where β is a very large number, and it's used to impose a very large cost value to exclude paths that have collisions, and it can take on a value of 1000. L is the distance from the starting point to the terminal point, the calculation formula is:

$$L = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{17}$$

where v is a flag variable. $v = 0$ means the path is non-collision path; otherwise, there is a collision path. The initial value of v is set to 0. In the rectangular obstacle environment, the following equation is used to determine whether the obstacle is encountered:

$$\begin{aligned} d = (xobs_k - space) < xx_j \&\& xx_j < (xobs_k + l\_obs_k + space) \&\& \\ (yobs_k - space) < yy_j \&\& yy_j < (yobs_k + w\_obs_k + space) \end{aligned} \tag{18}$$

where j is the number of nodes, including path nodes and interpolation points. xx_j and yy_j represent the abscissa and ordinate of the node. $(xobs_k, yobs_k)$ is the starting point of the kth rectangle obstacle, that is the bottom left corner of the rectangle. l_obs_k , w_obs_k is the length and width of the rectangular obstacle. *space* is a small positive number to avoid hitting the edge of the rectangle.

At each node, judge whether it collides with the obstacle according to formula 18. When it encounters the obstacle, d will generate a non-zero change value. When d changes, v will be added with one. If the path does not pass through any obstacle, then $v = 0$, otherwise, v is equal to a number that is not zero.

4.4 Path planning process

We combine GEI-BBO with cubic spline interpolation to design the path planning steps, the flow chart of path planning is shown in Figure 6.

Step 1: Determine the number of path nodes m and initialize the algorithm.

Step 2: The coordinates of n interpolation points are obtained by cubic spline interpolation.

Step 3: The path length L and the flag variable v are calculated respectively to obtain the value of fitness function.

Step 4: Using GEI-BBO to update the ordered habitat individuals.

Step 5: Determine whether the maximum number of iterations is reached. If yes, output the optimal result directly. If not, return to the step 2 for circulation.

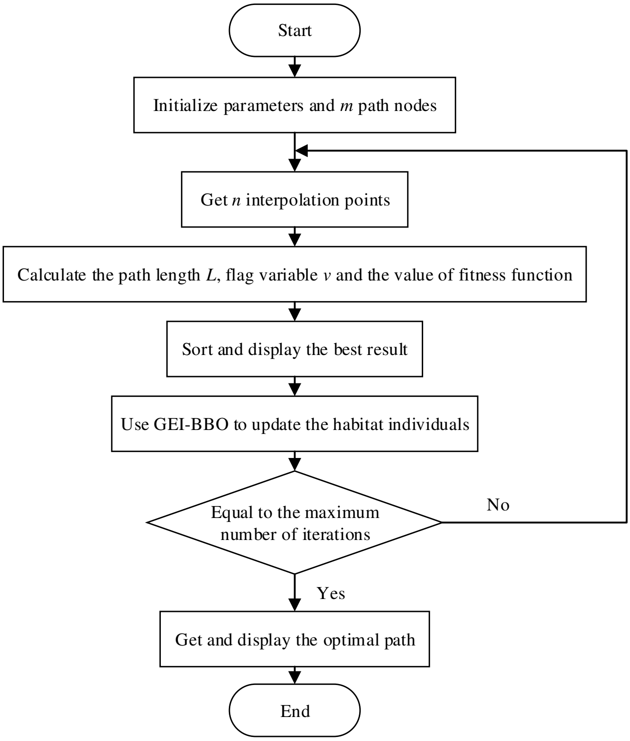


Fig. 4: Flow chart of path planning

5 Simulation results and analysis

In order to prove the effectiveness and practicability of the proposed method, we first use 23 benchmark test functions to compare and test the GEI-BBO, and then use it in the simulation experiment of path planning, and conduct in-depth analysis and summary of the experimental results.

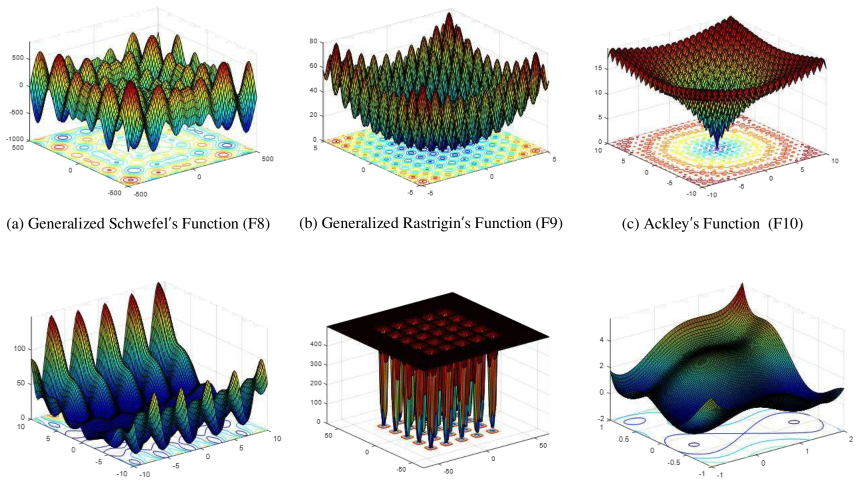
5.1 Experiments on benchmark test functions

We use the benchmark test function proposed by Yao et al. [27] for comparative test to verify the effectiveness of the algorithm proposed in this paper. There are 23 test functions in total, and the basic information is shown in Table 1.

Table 1: benchmark test functions

ID	Function name	Separability	Dimension	Search space	Optima
F1	Sphere model	Separable	30	$[-100, 100]^D$	0
F2	Schwefels problem 2.22	Non-separable	30	$[-10, 10]^D$	0
F3	Schwefels problem 1.2	Non-separable	30	$[-100, 100]^D$	0
F4	Schwefels problem 2.21	Non-separable	30	$[-100, 100]^D$	0
F5	Generalized Rosenbrocks functions	Non-separable	30	$[-30, 30]^D$	0
F6	Step function	Separable	30	$[-100, 100]^D$	0
F7	Quartic function	Separable	30	$[-1.28, 1.28]^D$	0
F8	Generalized Schwefels problem 2.26	Separable	30	$[-500, 500]^D$	-12569.5
F9	Generalized Rastrigins function	Separable	30	$[-5.12, 5.12]^D$	0
F10	Ackleys function	Separable	30	$[-32, 32]^D$	0
F11	Generalized Griewank function	Separable	30	$[-600, 600]^D$	0
F12	Generalized Penalized function 1	Non-separable	30	$[-50, 50]^D$	0
F13	Generalized Penalized function 2	Non-separable	30	$[-50, 50]^D$	0
F14	Shekels Foxholes function	Non-separable	2	$[-65.536, 65.536]^D$	0.9980
F15	Kowaliks function	Non-separable	4	$[-5, 5]^D$	0.00031
F16	Six-Hump Camel-Back function	Non-separable	2	$[-5, 5]^D$	-1.03162
F17	Branin Function	Non-separable	2	[-5,10] $\times$ [0,15]	0.39788
F18	Glodstein-Price function	Non-separable	2	$[-2, 2]^D$	2.99999
F19	Hartmans function 1	Non-separable	3	$[0, 1]^D$	-3.86278
F20	Hartmans function 2	Non-separable	6	$[0, 1]^D$	-3.32199
F21	Shekels Function 1	Non-separable	4	$[0, 10]^D$	-10.1531
F22	Shekels Function 2	Non-separable	4	$[0, 10]^D$	-10.4029
F23	Shekels Function 3	Non-separable	4	$[0, 10]^D$	-10.5364

It can be seen from Table 1 that function F1-F13 is a high-dimensional problem, function F1-F5 is unimodal, function F6 is a step function, function F7 is a quartic function with noise, function F8-F13 is a multimodal function, the number of local minimum values increases exponentially with the problem dimension, function F14-F23 is a low-dimensional function with only a few local minimum values [27]. This group of test functions includes not only low-dimensional and single-mode functions, but also many high-dimensional and multi-mode functions. It can detect the convergence speed of the algorithm well, and also reflect the ability of the algorithm to get rid of the poor local optimization and find the better near global optimization. We selected some of the more typical test functions, whose three-dimensional diagram is shown in Figure 5.



(a) Generalized Schwefel's Function (F8) (b) Generalized Rastrigin's Function (F9) (c) Ackley's Function (F10)

(d) Generalized Penalized Function (F12) (e) Shekel's Foxholes Function (F14) (f) Six-Hump Camel-Back Function (F16)

Fig. 5: Three dimensional graph of partial test function

In order to carry out the comparative test, we selected seven other evolutionary algorithms, namely GA [11], PSO [10], ACO [28], BBO [25], covariance matrix adaptation evolution strategy (CMA-ES) [29], whale optimization algorithm [30], sine cosine algorithm [31], salp swarm algorithm [32], hybrid invasive weed/biogeography-based optimization (IWO/BBO) [33] and linearized biogeography-based optimization (LBBO) [34], and Tables 2 and 3 shows the test results of comparative algorithms.

Table 2: Algorithm comparison results 1

ID	Fitness	GA	PSO	ACO	CMA-ES	WOA	SCA
F1	BEST	5.14E+01	9.29E-07	7.11E-15	3.53E-25	2.81E-83	1.73E-02
	MEAN	3.71E+02	2.75E-04	3.96E-13	8.04E-24	5.40E-74	1.71E+01
F2	BEST	3.03E-03	2.50E-02	1.09E-09	1.90E-12	**1.48E-58**	7.63E-11
	MEAN	5.69E-03	3.29E-01	1.02E-08	3.20E-12	**2.65E-53**	1.47E-08
F3	BEST	1.16E+03	2.17E+02	1.82E+04	1.64E-01	5.33E+00	9.00E-06
	MEAN	2.63E+03	1.05E+03	4.90E+04	2.59E+00	1.63E+02	1.77E-01
F4	BEST	1.18E+01	2.89E+00	9.32E+00	6.01E-10	1.95E-02	2.27E-05
	MEAN	1.70E+01	6.41E+00	3.87E+01	1.10E-09	2.32E+00	1.80E-03
F5	BEST	8.61E+01	2.69E+01	1.89E+01	1.75E+00	6.66E+00	7.02E+00
	MEAN	8.63E+02	7.10E+01	9.93E+01	2.28E+00	7.17E+00	7.54E+00
F6	BEST	5.59E+02	5.00E+00	**0.00E+00**	**0.00E+00**	9.15E-05	1.67E-01
	MEAN	1.03E+03	1.41E+01	**0.00E+00**	**0.00E+00**	1.87E-03	4.22E-01
F7	BEST	1.48E-02	2.35E-02	2.29E-02	1.40E-03	1.47E-04	1.25E-04
	MEAN	2.30E-02	5.88E-02	3.24E-02	2.93E-03	3.22E-03	4.04E-03
F8	BEST	-6.75E+03	-9.01E+03	**-1.09E+78**	-5.39E+03	-3.65E+03	-2.49E+03
	MEAN	-4.82E+03	-7.45E+03	-2.01E+78	-4.60E+03	-3.10E+03	-2.11E+03
F9	BEST	1.39E+01	2.29E+01	1.52E+02	1.34E+02	**0.00E+00**	**0.00E+00**
	MEAN	2.12E+01	4.12E+01	1.89E+02	1.53E+02	9.30E+00	1.15E-01
F10	BEST	4.08E+00	1.34E+00	2.00E-08	1.81E-13	**8.88E-16**	6.86E-09
	MEAN	6.42E+00	2.46E+00	1.34E-07	2.66E-13	**3.73E-15**	4.54E-06
F11	BEST	3.64E+00	4.78E-04	2.44E-15	**0.00E+00**	**0.00E+00**	2.96E-13
	MEAN	1.17E+01	5.97E-02	9.11E-03	**0.00E+00**	1.34E-01	5.48E-02
F12	BEST	1.13E+00	1.48E-03	6.27E-13	1.81E-24	2.36E-04	3.90E-02
	MEAN	4.86E+00	9.12E-01	2.07E-02	7.99E-24	2.51E-03	1.12E-01
F13	BEST	2.08E+01	1.11E-01	6.63E-12	**3.52E-24**	8.24E-03	2.01E-01
	MEAN	3.75E+01	6.43E+00	4.39E-03	**4.16E-23**	4.46E-02	2.93E-01
F14	BEST	1.33E+00	**9.98E-01**	**9.98E-01**	1.00E+00	9.98E-01	9.98E-01
	MEAN	5.03E+00	1.95E+00	3.51E+00	5.18E+00	2.86E+00	2.38E+00
F15	BEST	3.60E-04	**3.08E-04**	6.00E-04	1.29E-03	3.16E-04	6.06E-04
	MEAN	8.20E-04	1.88E-03	9.16E-04	2.69E-03	6.34E-04	1.17E-03
F16	BEST	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	**-1.03E+00**	-1.03E+00
	MEAN	-1.03E+00	-1.03E+00	**-1.00E+01**	-1.03E+00	-1.03E+00	-1.03E+00
F17	BEST	3.98E-01	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**	3.98E-01
	MEAN	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	4.00E-01
F18	BEST	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	3.00E+00
	MEAN	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	3.00E+00	3.00E+00
F19	BEST	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	-3.86E+00	-3.86E+00
	MEAN	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	-3.86E+00	-3.85E+00
F20	BEST	**-3.32E+00**	**-3.32E+00**	**-3.32E+00**	**-3.32E+00**	-3.32E+00	-3.16E+00
	MEAN	**-3.32E+00**	-3.24E+00	-3.27E+00	-3.26E+00	-3.09E+00	-3.01E+00
F21	BEST	-1.02E+01	-1.02E+01	**-1.02E+05**	-1.02E+01	-1.01E+01	-4.58E+00
	MEAN	-7.17E+00	-7.15E+00	-5.67E+00	-8.65E+00	-8.09E+00	-1.50E+00
F22	BEST	**-1.04E+01**	**-1.04E+01**	**-1.04E+01**	**-1.04E+01**	-1.04E+01	-6.48E+00
	MEAN	-7.44E+00	-7.15E+00	-4.48E+00	**-1.04E+01**	-7.90E+00	-2.87E+00
F23	BEST	**-1.05E+01**	**-1.05E+01**	**-1.05E+01**	**-1.05E+01**	-1.05E+01	-5.14E+00
	MEAN	-9.00E+00	-8.38E+00	-7.57E+00	**-1.05E+01**	-8.09E+00	-3.09E+00

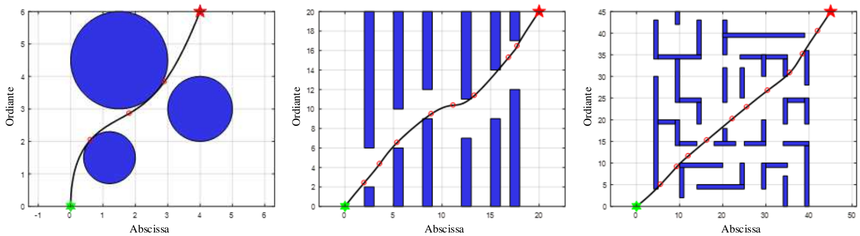
Table 3: Algorithm comparison results 2

ID	Fitness	SSA	BBO	IWO/BBO	LBBO	GEI-BBO
F1	BEST	5.97E-08	3.57E-01	4.42E-17	**0.00E+00**	8.95E-39
	MEAN	2.99E-07	5.88E-01	6.61E-17	**0.00E+00**	2.74E-24
F2	BEST	7.22E-06	1.92E-01	7.10E-09	6.86E-04	1.13E-33
	MEAN	3.14E-03	2.35E-01	2.02E-08	4.56E-03	5.25E-17
F3	BEST	1.92E-08	6.20E+01	1.23E-14	6.65E-12	**6.25E-21**
	MEAN	5.60E-07	8.67E+01	1.71E-14	2.30E-09	**3.93E-19**
F4	BEST	1.49E-05	6.53E-01	2.46E-11	3.40E-08	**5.64E-17**
	MEAN	2.95E-05	7.87E-01	4.54E-08	1.31E-07	**9.50E-17**
F5	BEST	6.41E+00	3.56E+01	8.49E-04	8.50E-03	**0.00E+00**
	MEAN	1.82E+02	1.17E+02	5.67E+01	2.34E+01	**0.00E+00**
F6	BEST	5.66E-10	**0.00E+00**	**0.00E+00**	2.84E+02	**0.00E+00**
	MEAN	8.97E-10	**0.00E+00**	**0.00E+00**	3.74E+02	**0.00E+00**
F7	BEST	5.46E-03	1.79E-03	3.11E-05	2.88E-02	**1.85E-05**
	MEAN	1.29E-02	4.16E-03	**3.19E-05**	3.69E-02	6.75E-05
F8	BEST	-3.14E+03	**-9.80E+03**	-4.38E+03	-6.74E+03	-9.79E+03
	MEAN	-2.79E+03	-9.04E+03	-3.59E+03	-4.79E+03	-8.72E+03
F9	BEST	3.98E+00	2.23E+01	1.78E-14	6.87E-11	**0.00E+00**
	MEAN	1.47E+01	3.40E+01	3.28E-05	1.67E-03	**2.23E-08**
F10	BEST	**1.00E-05**	1.98E-01	2.27E-03	2.32E+00	2.11E+00
	MEAN	8.78E-01	2.39E-01	3.68E-03	2.45E+00	2.12E+00
F11	BEST	5.90E-02	3.71E-01	2.39E-13	1.16E-05	1.11E-16
	MEAN	2.01E-01	5.78E-01	1.06E-02	9.53E-03	5.88E-16
F12	BEST	8.74E-12	8.64E-04	7.35E-17	1.03E-05	**2.73E-32**
	MEAN	8.48E-01	1.37E-03	5.47E-15	7.19E-04	**3.86E-30**
F13	BEST	5.50E-11	1.02E-02	1.87E-18	7.90E-10	1.10E-02
	MEAN	1.10E-03	1.96E-02	3.61E-17	6.08E-04	9.85E-02
F14	BEST	9.98E-01	**9.98E-01**	**9.98E-01**	**9.98E-01**	**9.98E-01**
	MEAN	1.10E+00	1.20E+00	**9.98E-01**	**9.98E-01**	2.08E+00
F15	BEST	6.04E-04	3.61E-04	**3.08E-04**	**3.08E-04**	**3.08E-04**
	MEAN	9.06E-04	5.38E-04	**3.08E-04**	**3.08E-04**	4.11E-04
F16	BEST	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.02E+00	**-1.03E+00**
	MEAN	**-1.03E+00**	-1.03E+00	-9.30E-01	-1.01E+00	**-1.03E+00**
F17	BEST	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**
	MEAN	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**
F18	BEST	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	MEAN	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
F19	BEST	-3.86E+00	**-3.86E+00**	-3.86E+00	**-3.86E+00**	**-3.86E+00**
	MEAN	-3.86E+00	**-3.86E+00**	-3.81E+00	-3.80E+00	**-3.86E+00**
F20	BEST	**-3.32E+00**	**-3.32E+00**	-2.37E+00	**-3.32E+00**	**-3.32E+00**
	MEAN	-3.22E+00	**-3.32E+00**	-2.52E+00	-3.31E+00	**-3.32E+00**
F21	BEST	-1.02E+01	**-1.02E+05**	-2.05E+00	-1.97E+00	-1.02E+01
	MEAN	-8.65E+00	**-1.02E+01**	-1.30E+00	-1.30E+00	**-1.02E+01**
F22	BEST	**-1.04E+01**	**-1.04E+01**	-3.92E+00	-5.10E+00	**-1.04E+01**
	MEAN	**-1.04E+01**	-7.73E+00	-1.96E+00	-5.04E+00	**-1.04E+01**
F23	BEST	**-1.05E+01**	**-1.05E+01**	-2.30E+00	-5.34E+00	**-1.05E+01**
	MEAN	-6.93E+00	-9.33E+00	-1.80E+00	-5.25E+00	**-1.05E+01**

In Tables 2 and 3, MEAN is the average of ten test results, BEST represents the best result of 10 and the bold part is the best one among 11 algorithms. Our proposed GEI-BBO obtains 16 optimal BEST and 14 optimal MEAN solutions in the optimization process of 23 functions, and its performance is better than the other 10 comparison algorithms. The experimental results verify the effectiveness of the optimization process of GEI-BBO on the benchmark function.

5.2 Simulation on path planning

In order to verify the effectiveness of the proposed algorithm in solving the problem of MRPP, we designed a planning environment through environment modeling, and carried out path planning simulation experiments using the proposed algorithm and other evolutionary algorithms. Fig.6 shows the best path planning results in three environments with GEI-BBO.



(a) environment 1 (b) environment 2 (c) environment 3

Fig. 6: Best path in different environments

In environment 1, the starting point coordinate is (0,0) and the ending point coordinate is (4,6). The obstacle consists of three circular obstacles, and the optimal path length is 7.5465. There are 12 rectangular obstacles in environment 2, the starting point coordinate is (0,0), the ending point coordinate is (20,20), and the optimal path length is 28.2721. Environment 3 is composed of more complex rectangular obstacles, starting point coordinate is (0,0), ending point coordinate is (45,45), and the optimal path length is 64.1673. In order to verify the feasibility of the proposed method, we have carried out a large number of simulation experiments. In three environments, each algorithm carries out 10 simulation tests, and takes the best, worst, mean and standard deviation of path length. The results are shown in Table 4. The success rate of each algorithm in the 10 times algorithm is shown in Table 5.

Table 4: Path planning results of different algorithms

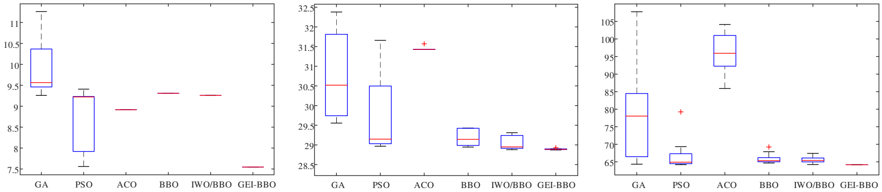
Environment	Algorithm	Best	Worst	Mean	SD
environment 1	GA	9.2584	11.2608	9.8816	0.4425
	PSO	7.5602	9.0484	8.7848	0.5801
	ACO	8.9149	8.9149	8.9149	0
	BBO	9.3092	9.3092	9.3092	0
	IWO/BBO	9.2580	9.2580	9.2580	0
	GEI-BBO	7.5465	7.5465	7.5465	0
environment 2	GA	29.5561	32.3786	30.7575	1.0769
	PSO	28.9670	31.6581	29.7025	1.2418
	ACO	31.4247	31.5697	31.44105	0.0021
	BBO	28.9453	29.4278	29.1776	0.0462
	IWO/BBO	28.8787	29.3112	29.0374	0.0309
	GEI-BBO	28.8721	28.9298	28.8917	0.0009
environment 3	GA	64.3348	107.7602	79.1527	201.8650
	PSO	64.1951	79.2291	66.8617	21.7340
	ACO	85.9004	104.1118	95.5918	35.6040
	BBO	64.6743	69.2475	65.9671	2.1830
	IWO/BBO	64.2394	67.4213	65.57496	0.8040
	GEI-BBO	64.1673	64.1673	64.1673	0

Combining Table 4 and Table 5, we can see that GEI-BBO has good performance in finding the optimal path. In environment 1, each algorithm can find its own optimal path, but compared with GA, PSO, ACO, BBO and IWO/BBO, GEI-BBO has the shortest path length, and the standard deviation is 0. In environment 2, except GEI-BBO, the success rate of other algorithms is not 1. From the optimal path length, the optimal value and mean value of GEI-BBO are optimal. In environment 3, GEI-BBO has the shortest path length of 64.1673, and the standard deviation is 0. Although other algorithms can also find the approximate optimal path with GEI-BBO, they are not very stable, and the success rate is low, among which the success rate of ACO is 0. These data verify the effectiveness of GEI-BBO algorithm in path planning.

Table 5: Success rates of algorithms

Environment	GA	PSO	ACO	BBO	IWO/BBO	GEI-BBO
environment 1	1	0.9	1	1	1	1
environment 2	0.6	0.8	0.2	0.9	0.9	1
environment 3	0.2	0.5	0	0.6	0.8	1

In order to make the experimental results more intuitive, we further add a box diagram, as shown in Fig.7. In environment 1, it can be seen that the variance of GA and PSO data is relatively large. Although the variance of ACO, BBO and IWO\BBO is 0, GEI-BBO has a smaller path length. In environment 2, the variance of GA and PSO is still larger, followed by BBO and IWO\BBO. The variance of ACO is similar to GEI-BBO, but its optimal path length is larger. In environment 3, except for ACO, the optimal path length of the other four algorithms is similar to GEI-BBO, but the variance is greater than GGEI-BBO, especially GA.



(a) environment 1 (b) environment 2 (c) environment 3

Fig. 7: Best path in different environments

6 Conclusion

In this paper, we proposed a new improved algorithm of BBO for MRPP, which combines the eigen-decomposition based migration, gradient descent and system search strategy. This method can effectively reduce the dependence of BBO on coordinate system and improve the local search ability. The comparison of 23 benchmark functions has proven the effectiveness of GEI-BBO. Combining GEI-BBO with cubic spline interpolation function, a method and fitness function for solving the obstacle avoidance and shortest path of mobile robot are constructed to solve the problem of MRPP. The simulation results also show that the proposed method has higher accuracy and success rate, which can prove the feasibility of the algorithm.

In the future, we will study how to use the new approaches to further enhance the performance of BBO, and apply the proposed algorithm to the more complex global path planning for mobile robots.

7 Acknowledgement

This work was supported by the State Key Laboratory of Robotics (2019-O18) and the Fundamental Research Funds for the Central Universities (DUT20LAB114, DUT2018TB06).

References

[1] Manikas, T.W., Ashenayi, K., Wainwright, R.L.: Genetic algorithms for autonomous robot navigation. IEEE Instrumentation & Measurement Magazine **10**(6), 26–31 (2007)

[2] Rasekhipour, Y., Khajepour, A., Chen, S.-K., Litkouhi, B.: A potential field-based model predictive path-planning controller for autonomous road vehicles. IEEE Transactions on Intelligent Transportation Systems **18**(5), 1255–1267 (2017)

[3] Sucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. IEEE Robotics & Automation Magazine **19**(4), 72–82 (2012)

[4] Borenstein, J., Koren, Y.: Histogramic in-motion mapping for mobile robot obstacle avoidance. international conference on robotics and automation **7**(4), 535–539 (1991)

[5] Zeng, N., Zhang, H., Chen, Y., Chen, B., Liu, Y.: Path planning for intelligent robot based on switching local evolutionary pso algorithm. Assembly Automation **36**(2), 120–126 (2016)

[6] Alam, M.S., Rafique, M.U.: Mobile robot path planning in environments cluttered with non-convex obstacles using particle swarm optimization. In: 2015 International Conference on Control, Automation and Robotics, pp. 32–36 (2015)

[7] Chen, Y.-b., Luo, G.-c., Mei, Y.-s., Yu, J.-q., Su, X.-l.: Uav path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science **47**(6), 1407–1420 (2016)

[8] Kim, J.-J., Lee, J.-J.: Trajectory optimization with particle swarm optimization for manipulator motion planning. IEEE Transactions on Industrial Informatics **11**(3), 620–631 (2015)

[9] Lim, K.K., Ong, Y.-S., Lim, M.H., Chen, X., Agarwal, A.: Hybrid ant colony algorithms for path planning in sparse graphs. In: Soft Computing - A Fusion of Foundations, Methodologies and Applications Archive, vol. 12, pp. 981–994 (2008)

[10] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995). IEEE

[11] Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., Bouzouia, B.: Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robotics and Autonomous Systems **89**(89), 95–109 (2017)

[12] Tharwat, A., Elhoseny, M., Hassanien, A.E., Gabel, T., Kumar, A.: Intelligent bzier curve-based path planning model using chaotic particle swarm optimization algorithm. Cluster Computing **22**(2), 4745–4766 (2019)

[13] Kumar, P.B., Sahu, C., Parhi, D.R.: A hybridized regression-adaptive ant colony optimization approach for navigation of humanoids in a cluttered environment. Applied Soft Computing **68**, 565–585 (2018)

[14] Castillo, O., Trujillo, L., Melin, P.: Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. In: Soft Computing - A Fusion of Foundations, Methodologies and Applications - Fuzzy-neural Computation and Robotics Archive, vol. 11, pp. 269–279 (2006)

[15] Liu, J., Yang, J., Liu, H., Tian, X., Gao, M.: An improved ant colony algorithm for robot path planning. In: Soft Computing - A Fusion of Foundations, Methodologies and Applications Archive, vol. 21, pp. 5829–5839 (2017)

[16] Contreras-Cruz, M.A., Ayala-Ramirez, V., Hernandez-Belmonte, U.H.: Mobile robot path planning using artificial bee colony and evolutionary programming. In: Applied Soft Computing Archive, vol. 30, pp. 319–328 (2015)

[17] Qu, H., Xing, K., Alexander, T.: An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing **120**, 509–517 (2013)

[18] Song, B., Wang, Z., Zou, L.: On global smooth path planning for mobile robots using a novel multimodal delayed pso algorithm. Cognitive Computation **9**(1), 5–17 (2017)

[19] Simon, D.: Biogeography-based optimization. IEEE Transactions on Evolutionary Computation **12**(6), 702–713 (2008)

[20] Simon, D., Ergezer, M., Du, D., Rarick, R.: Markov models for biogeography-based optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **41**(1), 299–306 (2010)

[21] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Let a biogeography-based optimizer train your multi-layer perceptron. Information Sciences **269**(269), 188–209 (2014)

[22] Zhu, W., Duan, H.: Chaotic predatorcprey biogeography-based optimization approach for ucav path planning. Aerospace Science and Technology **32**(1), 153–161 (2014)

[23] Mo, H., Xu, L.: Research of biogeography particle swarm optimization for robot path planning. Neurocomputing **148**, 91–99 (2015)

[24] Yang, J., Li, L.: Improved biogeography-based optimization algorithm

for mobile robot path planning. Chinese Intelligent Systems Conference, 219–229 (2017)

[25] Simon, D.: A probabilistic analysis of a simplified biogeography-based optimization algorithm. Evolutionary Computation **19**(2), 167–188 (2011)

[26] Du, D., Simon, D., Ergezer, M.: Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In: 2009 IEEE International Conference on Systems, Man and Cybernetics, pp. 997–1002 (2009)

[27] Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation **3**(2), 82–102 (1999)

[28] Mavrovouniotis, M., Yang, S.: A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. In: Soft Computing - A Fusion of Foundations, Methodologies and Applications Archive, vol. 15, pp. 1405–1425 (2011)

[29] Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2), 159–195 (2001)

[30] Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in Engineering Software **95**(95), 51–67 (2016)

[31] Mirjalili, S.: Sca: A sine cosine algorithm for solving optimization problems. Knowledge Based Systems **96**(96), 120–133 (2016)

[32] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm. Advances in Engineering Software **114**, 163–191 (2017)

[33] Khademi, G., Mohammadi, H., Simon, D.: Hybrid invasive weed/biogeography-based optimization. Engineering Applications of Artificial Intelligence **64**, 213–231 (2017)

[34] Simon, D., Omran, M.G.H., Clerc, M.: Linearized biogeography-based optimization with re-initialization and local search. Information Sciences **267**, 140–157 (2014)