

Research on Improvement of Bacterial Foraging Algorithm

Weihua Pan^{1, a}, Zhenkui Zhao^{1, b}

¹School of Control and Computer Engineering, North China Electric Power University, Baoding, 071003, Hebei province, China

^a23680118@qq.com, ^b1807751927@qq.com

Abstract

This article mainly studies the bacterial foraging optimization algorithm, explores the basic principles of the algorithm, deeply understands the chemotaxis operation, copy operation, and migration operation in the algorithm process. Through the study of the algorithm process, the algorithm is deeply understood, and the algorithm's chemotaxis process is improved. The optimal bacteria and the number of iterations are affected by the improvement of the genetic algorithm for the copy operation, the genetic algorithm hybridization and mutation are used to replace the copy only for the excellent bacteria, and for the migration operation, the combined influence of health and fitness is used to change the migration probability, and finally through experiments Verify the effectiveness of the algorithm.

Keywords

BFO; IBFO; Combinatorial optimization.

1. Introduction

In the human body, there is a kind of bacteria-Escherichia longa, which is usually called Escherichia coli. This kind of bacteria coexists harmoniously with people. In order to survive better, bacteria can obtain food in nutrient-rich areas. After countless generations of evolution, bacteria have acquired the ability to move and obtain food through their own evolution. Through this ability, bacteria can compete with other types of bacteria for food. And it can also compete for food with the same bacteria in the same area. This type of bacteria is usually a group of bacteria as a whole. After mutual feedback and exchange of information, the bacterial population gradually moves into the food-rich area.

Cilia and flagella grow on the surface of bacteria. These two kinds of hair have their own functions. The function of cilia is to transmit a certain gene between bacteria, and the function of flagella is to control the movement of the body. Moreover, the behavior of bacteria in the movement of looking for food is guided by their own control system. At the same time, this can guide bacteria to always move toward the food-rich area, and it can avoid damage to itself when looking for food. substance. For example, in the process of looking for food, it will avoid the acid and alkaline places that are harmful to it. At the same time, it will summarize the effect of its own movement and provide a reference for its next movement. The moving direction of the bacteria is also affected by the hair, which determines whether the bacteria move forward or backward. Bacteria are rejected by bacteria in these poor areas during their lifetime. Eventually the bacterial population moves to food-rich areas. When bacteria reach the food-rich area, the bacteria will split from one to two, and the bacteria that have not entered the food-rich area will die.

Therefore, Professor Passino has established a bacterial foraging optimization algorithm through modeling through studying the process of the bacteria looking for food. The algorithm

simulates a series of activities in the search for food by bacteria, and summarizes them into three movements: trend, copy, and migration.

2. Literature References

The BFO algorithm finds the optimum through the random search of multiple individuals in the population in the optimization space, combined with the mutual cooperation between individual bacteria. According to a series of activities of individual bacteria in the entire life cycle, the algorithm summarizes all its own behaviors into tropism operation, copy operation and migration operation, and finds the optimal solution of the problem through the continuous repetition of the above process. However, the algorithm has the problem of weak global optimization ability in solving optimization problems, some improvements to the algorithm are needed.

Swarm intelligence (as an emerging intelligent computing technology has become the focus of artificial intelligence researchers. In 2002, Professor Passino proposed a bacterial foraging optimization algorithm (BFO) to solve the optimization problem based on the foraging and chemotaxis behavior of bacteria.), adding a new member to the swarm intelligence family. BFO guides optimized search through swarm intelligence generated by the cooperation and competition of various bacteria in the swarm. It has better global search capabilities. In recent years, BFO has gradually attracted domestic and foreign experts The attention of scholars and the corresponding research on the theory and application of this algorithm. A research team composed of Abraham, Biswas, Kim and others have conducted a series of studies on BFO and demonstrated that the copy operation in BFO is beneficial to improve the convergence speed of the algorithm. , Using genetic algorithm and differential evolution algorithm to improve the ability of BFOA to search for the global optimum, and BFO is used in PID controller parameter optimization; Indian scholars Sambarta Dasgupta and Swagatam Das theoretically analyzed the use of adaptive mechanism step size algorithm Convergence and stability are affected, but their theoretical analysis is based on certain conditional assumptions, only considering the tropism operation of a single particle in one-dimensional space; Chu Ying et al. discussed the quorum sensing mechanism in the colony algorithm Improved, proposed a fast bacterial swarm algorithm and applied it to image compression processing; Li Ming et al. improved the mechanism of the bacterial colony algorithm, proposed a bacterial colony optimization algorithm, and verified the effectiveness of the colony optimization algorithm Sex.

3. Algorithm Research and Improvement

3.1. Algorithm Research

Escherichia coli has two basic actions in the entire search for food: rotating action and swimming action. Rotating action is to change a direction, while swimming action is to keep the direction unchanged. The trend operation of the algorithm is modeled after these two basic actions. Usually, the swimming action of bacteria will be in the place with moderate acidity and alkalinity or a large amount of food, and will leave by rotating action in the place where food is lacking or the environment is harmful to you. Its motion behavior is as follows: the bacteria first swim in any direction; if the fitness value of the new position in this direction is smaller than the fitness value of the previous step, then it will start to rotate and face another random Swim in the direction; if the fitness value of the new position in this direction becomes larger, then continue to swim in this randomly generated direction; if the maximum allowable number of swimming actions is reached, then the tropism operation of this bacteria ends and starts Go to the next bacterium to perform this trending action. Each step of the trend of bacteria is expressed in mathematical expressions as follows:

$$\theta(i, j+1, k, l) = \theta(i, j, k, l) + C(i) * \phi(i) \tag{1}$$

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{2}$$

In the formula, $\theta(i,j,k,l)$ refers to the place where the bacteria i executes the trending operation j times, the copy operation executes k times, and the migration operation executes l times. In the formula, $\Delta(i)$ represents the vector in a random direction, and $C(i)$ represents the unit length of each movement.

Every bacterium can produce and disperse chemical substances at all times during its life cycle. This chemical substance can not only transmit the food information of the place where it is currently staying to other bacteria, and through this affect whether other bacteria swim to that place; but also transmit it. Information repels other individuals, and asks other individuals not to gather too densely with it, and use this behavior to maintain mutual distance. Therefore, in the BFO algorithm, each bacterium is affected by two factors in the behavior of searching for food. It depends on whether the food in front is abundant and whether the individual distance is too dense to affect where to move. This behavior is the quorum sensing mechanism.

In the stage of the bacterial population searching for food, the closeness of each bacteria to each other is caused by mutual attraction and repulsion. Each bacteria attracts and repels each other. Attract the bacteria together. Repulsion makes the location of the bacteria different, ensuring that there is food to get at the location where they are staying, so that they can survive. This process can be realized by the following formula.

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^s j_{cc}^i(\theta, \theta^i(j, k, l)) \tag{3}$$

$$= \sum_{i=1}^s [-d_{attractant} \exp(-w_{attractant} \sum_{m=1}^p (\theta_m - \theta_i^m)^2)] + \sum_{i=1}^s [-h_{repellant} \exp(-w_{repellant} \sum_{m=1}^p (\theta_m - \theta_i^m)^2)]$$

In the formula, $P()$ represents the current position of each individual in the population, $J_{cc}()$ represents the value of the objective function, s represents the total number of bacteria, p is the number of optimized variables, $d_{attractant}$, $w_{attractant}$, $h_{repellant}$, and $w_{repellant}$ are gravitational depth and gravitational force, respectively Width, repulsion height, and repulsion width.

Taking into account the bacteria’s own foraging behavior and the influence of the quorum sensing mechanism on it, after introducing the aggregation operation in the tropism cycle, the calculation formula for the fitness value of the i -th bacteria becomes.

$$j^i(j+1, k, l) = j^i(j+1, k, l) + J_{cc}(\theta^i(j+1, k, l), P(j, k, l)) \tag{4}$$

In the process of evolutionary survival of all living things in nature, those who adapt have survived, and those who have not adapted will die and disappear. After a period of searching for food, the bacteria in the bacterial population will die. However, in order to keep the population size unchanged, other surviving bacteria with strong searching ability will be divided into two, increasing the number of bacteria. Quantity. Replicative operations are used in algorithms to express this behavior.

$$J_{health}^i = \sum_{j=1}^{N_c} J^i(j, k, l) \quad (5)$$

J_{health} is the expression of the degree of health of bacteria i . The N_c value represents the number of tropism operations performed by bacteria i . The current health degree calculation of the individual bacteria i is equal to the sum of all the fitness values of bacteria i performing N_c tropism operations. Here, the health of bacteria can be used to measure the ability of bacteria to obtain food. In the copying behavior, all bacteria in the population are arranged according to J_{health} , the order is from large to small, the first half is copied, and the remaining half is discarded, that is the ones with strong ability to obtain food retain the copy, and the ones with poor ability to find are discarded. This is The process of copying behavior.

In the real situation, the local area where the bacteria live may change slowly (for example, the food inside is eaten) or suddenly change (for example, the temperature in the place becomes higher, etc.). These conditions will cause the bacterial populations in this area to move to new places or be eliminated altogether. Use migratory operations in the algorithm to mimic this situation. Although the migration operation changes the tropism operation of the bacteria, it may also move to a place with high nutrient concentration. Therefore, in the long run, this behavior is helpful for the bacteria to find food. In order to imitate the process of modeling, in the algorithm, the population undergoes multiple replication operations, and the bacterial individuals perform migration operations according to the set probability, and the bacterial individuals begin to randomly allocate to the optimization area again. That is to say: if the migration probability of a certain bacteria in the flora is satisfied, then the bacterial individual is eliminated, and a new individual is randomly generated anywhere in the optimization area. The newly generated individual may have different location information from the original individual. That is, the ability to find food is different. The new individuals randomly generated by the migratory behavior operation may be closer to the global optimal solution, which is helpful for the trend operation to jump out of the local optimal solution, so that the global optimal solution can be searched for.

The migration operation behavior appears according to the set probability P_{ed} . If a certain bacteria in the population meets the probability of migration, then this bacteria is eliminated, and a new individual is randomly generated at any place. This new individual and the eliminated individual may have different positions. The migration operation helps the BFO algorithm to obtain the ability of random search, which is helpful for the BFO algorithm to maintain the diversity of the population, so that the situation of premature convergence can be avoided.

The main calculation steps of the bacterial foraging optimization algorithm are as follows:

1. Initialize each parameter: initialize each parameter, p is the search space dimension, s is the total number of bacteria, N_c : the number of steps in the chemotaxis step, $C(i)$: the swimming length in the chemotaxis, N_{re} : the number of replication steps, N_{ed} : the number of elimination-dispersion events, P_{ed} : elimination-dispersion probability, $d_{attractant}$, $w_{attractant}$, $h_{repellant}$, $w_{repellant}$ gravity depth, gravity width, repulsion height and repulsion width.

2. Initialize the individual bacteria

The number of randomly generated bacteria is S , the position of the bacteria is initialized, and then the initial fitness value of each bacteria is calculated.

3. Initialize loop variables

Set the parameters of the loop variable, where the number of migration cycles is set to $l=1:N_{ed}$, the number of replication cycles is set to $k=1:N_{re}$, and the number of chemotaxis cycles is set to $j=1:N_c$.

4. Trend operation process:

In the chemotaxis process, first flip, and update the position of the bacteria according to the flip formula (1),

Then start the swimming operation. If the fitness function value of the bacteria in the new place becomes larger, then continue swimming in this direction, when the fitness function value does not change or the number of chemotaxis steps reaches the set maximum value.

5. Copy operation process:

And after the chemotaxis cycle ends and the replication operation is started, the fitness function value of each bacteria in the life cycle is added to obtain the bacterial health, which is sorted according to the bacterial health, and the half of the poor health is eliminated, which is good for the health. Half of them are copied.

6. Migration operation process:

When the migration operation is started, a random probability P_i is generated. If $P_i < P_{ed}$, the bacteria will migrate; otherwise, no migration will be performed.

Algorithm termination: Check whether the termination requirements are met (the number of iterations reaches the maximum or the fitness value meets the accuracy requirements), if the termination conditions are met, then the optimal solution is output, and the entire algorithm ends.

3.2. Algorithm Improvement

The cube mapping is used to generate variables, and the cube mapping formula is as follows:

$$y_d(i+1) = 4y_d(i)^3 - 3y_d(i) \text{ and } -1 < y_d(i) < 1$$

$$(i \in \{1, 2, \dots, S-1\}, d \in \{1, 2, \dots, D\}) \quad (6)$$

Assuming that there is a bacterial group in the population, the search space is dimensional, and the position of the bacteria can be represented by a dimensional vector. Therefore, m n -dimensional vectors are generated according to the cube mapping to represent the initial bacterial position.

$$x_d(i+1) = 4x_d(i)^3 - 3x_d(i) \text{ and } -1 < x_d(i) < 1$$

$$(i \in \{1, 2, \dots, m-1\}, d \in \{1, 2, \dots, n\}) \quad (7)$$

Next, the first individual $X(1)$ is generated using a random generation strategy.

$$X(1) = [x_1(1), x_2(1), \dots, x_n(1)] \quad (8)$$

Then generate $m-1$ n -dimensional chaotic variables according to the cube mapping formula, as shown below.

$$X(i) = [x_1(i), x_2(i), \dots, x_n(i)]$$

$$(i \in \{2, 3, \dots, m\}) \quad (9)$$

Next, the chaotic variables generated can be transformed into the optimization space through the following linear transformation formula.

$$P_d^i(0,0,0) = \text{Min} + (\text{Max} - \text{Min}) * x_d(i) \tag{10}$$

$$d \in \{1,2,\dots,n\}$$

$P_d(0,0,0)$ is used to indicate the first place where the i -th bacterium is produced in the d -th dimension. $\text{Min}+(\text{Max}-\text{Min})*x_d(i)$ means to change the variable into the solution area. Through this method of initialization, the bacteria in the population can be more evenly distributed in the optimization area. Using this method to provide help for the subsequent global optimization. Aiming at some shortcomings of current trend operation, the particle swarm algorithm is combined with the chemotaxis process. The improved chemotaxis process mainly takes into account the influence of the optimal bacteria and the number of iterations on the chemotaxis direction and step length of the bacteria during the chemotaxis of the flora. Specifically, the optimal bacteria in the flora are used to guide the trending behavior of the bacteria. Improve the performance of the algorithm.

First, improve the flipping direction so that the bacterial individuals swim in the optimal direction of the group, so the global learning direction is introduced.

$$\Delta_G(i) = (P_{gbest} - P_i) * \|P_{gbest} - P\|_2 \tag{11}$$

Among them, P_{gbest} is the optimal bacterial position in the bacterial population; P_i is the i -th bacterial position. In order to prevent falling into the local optimum, the random flip direction is retained, so the chemotaxis direction becomes the following formula during the chemotaxis:

$$\Delta_m(i) = \alpha * \Delta(i) + (1 - \alpha) * \Delta_G(i) \tag{12}$$

The formula $\Delta_m(i)$ represents the improved flipping direction. For the swimming step length, the traditional one is fixed length, combined with the particle swarm algorithm considering the global optimal particles and the influence of the number of iterations, combined with the global optimal bacteria and the iterative process to affect chemotaxis The step size is as follows:

$$C(i) = \beta * (\nu \exp(-13 * (\frac{g_i}{g_{max}})) + (1 - \nu) * \frac{J_i - J_{min}}{J_{max} - J_i} * rand()) \tag{13}$$

$$\nu = 0.1 + 0.8 * \frac{g_{max} - g_i}{g_{max}}$$

Where g_{max} is the maximum number of iterations, and g_i is the current number of iterations. Using the formula (13) to improve the chemotaxis step length method, the step length is mainly adjusted by the number of iterations in the early stage, and the optimal bacteria influence the step length in the later stage. So the chemotaxis formula is:

$$\theta(i, j + 1, k, l) = \theta(i, j, k, l) + C(i) * \Delta_m(i) \tag{14}$$

In the copying behavior, all bacteria in the population are arranged according to J_{health} , the order is from largest to smallest, the first half is copied, and the remaining half is discarded, that is, the ones with the strong ability to obtain food retain the copy, and the ones with the poor ability to find the food are discarded. The replication operation is to allow rapid convergence of the population, and in this way, the optimization efficiency is improved. However, just using

individuals with good health to replace poor individuals. This method cannot increase the optimal value in the group and reduces the diversity of individuals in the group. This way, by reducing the diversity of the group, the efficiency of population convergence is improved. The method is likely to cause the algorithm to enter the local optimum.

Therefore, combined with the hybridization and mutation theory of genetic algorithm, the hybridization operator is used to make the current poorer half of the bacterial individuals hybridize with the excellent bacterial individuals, so that the poorer half of the bacterial individuals move closer to the best position; use the mutation operator to affect the current health. Half of the excellent bacteria undergo subtle position changes near their own position. This method solves the problem of premature algorithm fall into.

The formula for the crossover operator is:

$$X'(i, k) = \lambda X(i, k) + (1 - \lambda) X(best, k) \quad (15)$$

Among them, $[0, 1]$ is a uniformly distributed random number, and $X(best, k)$ is the optimal position of the current flora.

The mutation operator formula is:

$$X'(i) = X(i) + N(0,1) \quad (16)$$

The improved replication operation can be expressed as: After the bacteria finish the chemotaxis operation, compare the health value $J_{ihealth}$ of all bacteria in the population, and take the bacteria with the best fitness value as the guiding bacteria; for the bacteria with larger fitness value $S_r = S/2$ uses a formula to perform mutation operations; the remaining bacteria hybridize with the guiding bacteria, and this method maintains the diversity of the flora and improves the accuracy of the solution.

The migration operation mimics the phenomenon that bacterial individuals are eliminated because they cannot obtain the food they need or enter a place that is harmful to them, to randomly remove a part of the individuals, and randomly generate the same number of new individuals in the optimization area. The migration operation is to be able to use the method of randomly generating new individuals to expand the diversity of the population and avoid the population from falling into the local optimum. However, setting the migration probability too large will cause more individuals to change their positions, which will prompt the algorithm to jump out of the local extreme. However, if the bacterial individuals that have found the optimal place or are close to the optimal place are migrating, it will make them leave the best place. The greater the probability of migration, the greater the probability that high-quality bacteria individuals will leave the best place, but if the probability of migration is simply reduced, the bacteria will have no way to leave the local optimal place.

In order to solve the problem that bacteria cannot leave under local optimal conditions and prevent high-quality bacteria from migrating away, and to improve the ability of bacteria to search for better positions, some bacteria need to be expelled, and other parts are scattered in the search area according to probability. Considering the two factors of bacterial fitness and health, together to provide guidance on the migration of bacteria, thereby affecting the performance of the algorithm.

Design an adaptive migration probability P_{ged} , all bacteria migrate according to the following probability.

$$P_{ged} = \left(1 - \frac{J_{health}^i - J_{health}^{\min}}{J_{health}^{\max} - J_{health}^{\min}} * \frac{J^i - J^{\min}}{J^{\max} - J^{\min}}\right) * P_{ed} \quad (17)$$

$J_{ihealth}$ refers to the health of the i -th bacterium, and J_i refers to the fitness value of the i -th bacterium.

Therefore, the improved migration operation is to calculate the P_{ged} of the selected individual. If $P_{ged} < P_{ed}$, then the bacteria will migrate, otherwise, no migration will be performed.

The main calculation steps of the bacterial foraging optimization algorithm are as follows:

1. Initialize each parameter: initialize each parameter, p is the search space dimension, s is the total number of bacteria, N_c : the number of steps in the chemotaxis step, $C(i)$: the swimming length in the chemotaxis, N_{re} : the number of replication steps, N_{ed} : the number of elimination-dispersion events, P_{ed} : elimination-dispersion probability, $d_{attractant}$, $w_{attractant}$, $h_{repellant}$, $w_{repellant}$ gravity depth, gravity width, repulsion height and repulsion width.

2. Initialize the individual bacteria

According to formula (7), initialize the location where each bacteria is generated, and calculate the initial fitness value of each bacteria.

3. Initialize loop variables

Set the value of N_{ed} , N_{re} , and N_c , where the number of migration cycles $l=1$: N_{ed} , the number of replication cycles $k=1$: N_{re} , and the number of chemotaxis cycles $j=1$: N_c .

4. Trend operation process:

In the chemotaxis process, first flip, update the position of the bacteria according to the improved flip formula (12), and then perform the swimming operation according to (14). If the fitness function value of the bacteria at the new position is different To improve, continue to swim in this direction until the fitness function value does not change or reaches the predetermined maximum number of chemotaxis steps.

5. Copy operation process:

Calculate the health of all bacteria and sort them in order from largest to smallest, and select the best bacteria currently as the guide bacteria;

Perform mutation operation on the current best half of bacteria according to formula (15)

Cross the current worst half with the guiding bacteria selected from the above bacteria according to formula (16);

6. Migration operation process:

When entering the migration operation, first calculate p_{ged} , and then randomly generated P_{ed} , if $P_{ged} < P_{ed}$, then the bacteria migrate, otherwise the bacteria position will not change.

7. Algorithm termination:

Check whether the termination requirements are met (the number of iterations reaches the maximum or the fitness value meets the accuracy requirements), if the termination conditions are met, then the optimal solution is output, and the entire algorithm is completed.

4. Experimental analysis and verification

In order to verify the performance comparison between the improved bacterial foraging algorithm and the original algorithm, this section uses 5 standard functions to test it, and compare the results with each other.

Table 1. Test function

function	formula	circumscription	Extreme value	type
Sphere	$\sum_{i=1}^D x_i^2$	[-100,100]	0	Single
Rosenbrock	$\sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	0	Single
Rastrigin	$\sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]	0	multiple
Griewank	$\frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0	multiple
Ackley	$-20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^D x_i^2}) - \exp[\frac{1}{n} \sum_{i=1}^D \cos(2\pi x_i)] + 20 + e$	[-32,32]	0	multiple

The test results are as follows:

Table 2. Test result

function	algorithm	best	average
Sphere	BFO	0.432	0.623
	IBFO	0.146	0.078
Rosenbrock	BFO	48.227	71.384
	IBFO	24.759	31.619
Rastrigin	BFO	1.151E2	1.328E2
	IBFO	13.254	31.495
Griewank	BFO	0.856	0.936
	IBFO	1.304E-3	1.861E-2
Ackley	BFO	14.095	15.952
	IBFO	0.096	0.754

The experimental results show that for the five basic test functions, the average value and optimal value obtained by the improved bacterial foraging algorithm are closer to the minimum value of the test function than the original algorithm, which shows that the performance of the improved bacterial foraging algorithm is better when solving the optimization problem, and verifies the effectiveness of the improved strategy.

5. Summary

In this paper, through the improvement of BFOA algorithm, the convergence speed and accuracy of the algorithm are improved. The experimental results show that the improved algorithm has greatly improved the optimization effect.

References

[1] Lin Shijie, Dong Chen, Chen Mingzhi, Zhang Fan, Chen Jinghui. Overview of new swarm intelligence optimization algorithms [J]. Computer engineering and applications, 2018,54 (12): 1-9

- [2] Liu Yan. Swarm intelligence optimization algorithm and its application in optimization design [D]. Nanchang Aeronautical University, 2018
- [3] Yang Cuicui. Bacterial foraging optimization algorithm and its application [D]. Beijing University of technology, 2017
- [4] Li Jun. research and improvement of bacterial foraging optimization algorithm [D]. Lanzhou Jiaotong University, 2018
- [5] Liang Yingxin, Tian Haoshan. Improved hybrid algorithm based on bacterial foraging and particle swarm optimization [J]. Electronic Science and technology, 2017,30 (04): 79-82
- [6] Yan Xiaohui, Zhu Yunlong, Zhang Zhicong, LV Cixing, Li Shuai, Yi Wenjie. Review on optimization methods and application of flora [J]. Journal of Zhengzhou University (Engineering Edition), 2018,39 (05): 1-10 + 21
- [7] Wang Jian, Shi Jing. Prediction model of SVR photosynthetic rate based on chemotaxis improved particle swarm optimization [J]. China Agricultural Science and technology guide, 2018,20 (12): 74-82
- [8] Y.Yao, J. Ren, R. Bi and Q. Liu, "Bacterial Foraging Algorithm Based on Activity of Bacteria for DNA Computing Sequence Design," in IEEE Access, vol. 9, pp. 2110-2124, 2021, doi: 10.1109/ACCESS.2020.3047469.
- [9] Qi Bian, Brett Nener, Xinmin Wang, A modified bacterial-foraging tuning algorithm for multimodal optimization of the flight control system, Aerospace Science and Technology, Volume 93, 2019, 105274, ISSN 1270-9638,
- [10] K.Xiong, W. Wu, N. Li, L. Yang, J. Zhang and Z. Wei, "Risk-Based Multi-Objective Optimization of Distributed Generation Based on GPSO-BFA Algorithm," in IEEE Access, vol. 7, pp. 30563-30572, 2019, doi: 10.1109/ACCESS.2019.2902886.
- [11] J.Zhang et al., "A Coupling Approach With GSO-BFOA for Many-Objective Optimization," in IEEE Access, vol. 7, pp. 120248-120261, 2019, doi: 10.1109/ACCESS.2019.2937538.