Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **9**, Number **4**, November **2024**

Design of Equipment Intelligent Control Algorithm Based on Microbial Optimization Fuzzy PID

Chao Wu*
College of Electrical Engineering
Changzhou Vocational Institute of Mechatronic Technology, Changzhou 213164, P. R. China
420305735@qq.com

Bin Zhou
College of Electrical Engineering
Changzhou Vocational Institute of Mechatronic Technology, Changzhou 213164, P. R. China
672893358@qq.com

*Corresponding author: Chao Wu
Received December 4, 2023, revised March 17, 2024, accepted June 12, 2024.

Abstract. *This article introduces the design of an intelligent control algorithm for equipment based on microbial population optimization fuzzy PID (P is the proportion, which is the input deviation multiplied by a coefficient; I is integral, which means integrating the input deviation; D is differentiation, performing differentiation operations on input deviations.). Firstly, an overview was given of the fuzzy PID control algorithm and the bacterial colony optimization algorithm, and the principle and method of applying the bacterial colony optimization algorithm to the parameter optimization of the fuzzy PID controller were elaborated. Then, a detailed introduction was given to the design process of a device intelligent control algorithm based on microbial optimization fuzzy PID, including fuzzification functions, rule libraries, defuzzification functions, as well as the fitness functions and parameter adjustment strategies of the microbial optimization algorithm. Finally, the effectiveness and superiority of this algorithm were verified through experiments, and compared with traditional PID control algorithms. The experimental results show that the algorithm has better control performance and robustness, and can effectively achieve intelligent control of devices.*
Keywords: Microbial optimization; Fuzzy PID; Intelligent control; Parameter optimization; control performance

---

1. **Introduction.** With the continuous improvement of industrial automation level, the requirements for equipment control systems are also increasing. The traditional PID control algorithm can no longer meet the needs of modern industrial control. In order to improve the control accuracy and robustness of the equipment, researchers have proposed an intelligent control algorithm based on microbial population optimization fuzzy PID for the equipment. This algorithm applies the bacterial colony optimization algorithm to the parameter optimization of the fuzzy PID controller, achieving intelligent control of the equipment. This article will provide a detailed introduction to the design process and experimental verification results of the algorithm. Microbial optimization can solve the parameter optimization problem in fuzzy PID, or fuzzy PID can provide adaptive control required for microbial optimization.

In order to improve the accuracy of network intrusion detection, the Deep Belief Networks (DBN) algorithm is used for intrusion detection, while the Bacterial Foraging

Optimization (BFO) algorithm is used to solve the optimal DBN parameters [1]. The multi-objective optimization problem widely exists in scientific research and engineering practice, and there is no unique optimal solution. Its solution is a set of Pareto solutions. Heuristic search algorithms based on population evolution have become the main means of solving multi-objective optimization problems today, as they can simultaneously obtain multiple Pareto solutions in one run [2]. A one-step predictive control algorithm based on microbial population optimization algorithm and model learning is proposed. The algorithm establishes a system mathematical model based on nonlinear system input and output data using least squares support vector machine, obtains system output estimation values, uses feedback correction to reduce prediction errors [3]. On the basis of introducing several typical microbial community optimization methods, some scholars focused on keyword analysis of relevant literature on bacterial foraging algorithms. Combining the shortcomings of keywords and original bacterial foraging algorithms, they reviewed the improvement research of microbial community optimization methods from four aspects: parameter and structure optimization, multi algorithm mixing, operator improvement, and multi-objective optimization transformation [4].

Luo [5] used PID technology to combine pulse signal threshold and overcurrent coefficient in electrical automatic control systems to construct a direct current transfer function Three adjustment mechanisms, namely proportional, differential control, and actual parameter integration, were used to calculate the parameter difference of the automatic control program. Gao [6] proposed a Fuzzy PID composite controller (PSO Fuzzy PID) based on Particle Swarm Optimization (PSO) algorithm. Chen [7] proposed a position control method for electro-hydraulic proportional systems based on radial basis function (RBF) neural network tuning PID. Cen [8] designed a fuzzy PID intelligent controller by combining PID algorithm and fuzzy control, which utilizes a microcontroller to form a temperature control system. Cui et al. [9] designed a light and small spherical amphibious robot based on fuzzy PID to solve the problem of strict working environment requirements for mobile robots, which is suitable for variable and narrow environments. Peng and Chen [10] established a half vehicle semi-active suspension model based on Matlab /Simulink, and designed a fuzzy PID controller to control the semi-active suspension. In order to improve the clustering performance of the nearest neighbor propagation clustering algorithm, Zhang [11] adopted the bacterial colony algorithm to optimize the parameters of nearest neighbor propagation bias.

Guo et al. [12] proposed a K-means clustering algorithm based on microbial community optimization. Establish a data clustering model using the K-means algorithm. Wang et al. [13] adopted a radial basis function neural network algorithm based on microbial community optimization to fuse sensor network data, remove redundant perception data, and reduce data dimensions. The microbial community algorithm, as an emerging heuristic search algorithm based on population evolution, has made preliminary explorations in multi-objective optimization problems in recent years, but there is still a lack of in-depth research on its population evolution mechanism. Weng [14] proposed a one-step predictive control algorithm based on microbial community optimization algorithm and model learning. The system mathematical model is established through least squares support vector machine to obtain estimated system output values [15]. Qiu et al. [16] used quantum microbiota algorithm to solve non convex objective optimization functions.

The relevant work of this paper includes:

Literature review: Review the relevant literature on fuzzy PID control algorithm and bacterial colony optimization algorithm, introduce the development history, principles and methods of these two algorithms, as well as their application status in the field of equipment control. Zhao proposed a temperature control system using Smith fuzzy PID controller [17]. Ma proposed a coal selection automatic control technology based on fuzzy PID control [18]. In order to improve the control accuracy of the telescopic arm of special cranes, Ma and He [19] proposed an automatic control method for the telescopic arm of special cranes based on fuzzy PID fusion. In response to the shortcomings of existing literature, the advantages and innovations of this research method, and how the proposed new algorithm overcomes the limitations of existing algorithms.

Experimental verification: Verify the effectiveness and superiority of the device intelligent control algorithm based on microbial population optimization fuzzy PID through experiments. The experimental results show that the algorithm has better control performance and robustness, and can effectively achieve intelligent control of devices. By analyzing the working principle and trajectory shaping control requirements of the cutting arm, Guo [20] used a genetic algorithm (GA) combined with a fuzzy PID controller to complete the swing speed control of the cutting arm. Sun [21] designed a dual tank liquid level control experimental system based on fuzzy PID. Zhou et al. [22] used the Fuzzy PID (Proportional Integral Derivative) control algorithm to output pulse width modulation waves with different duty cycles, controlling the consistency between the temperature of radiative cooling materials and the ambient temperature.

Parameter optimization: The bacterial colony optimization algorithm was applied to the parameter optimization of the fuzzy PID controller. The control effects under different parameters were compared through experiments, and the optimal parameter combination was obtained. Chen et al. [23] combined PID algorithm and fuzzy control to design a fuzzy PID intelligent controller, which utilizes a microcontroller to form a temperature control system.

Control strategy analysis: Analyze the control strategy of the device intelligent control algorithm based on microbial optimization fuzzy PID, including fuzzification function, rule library, as well as the fitness function and parameter adjustment strategy of the microbial optimization algorithm. Liu et al. [24] propose an improved particle swarm optimization based fuzzy PID control method for nonlinear control updates of inertia weights and learning factors, enhancing population optimization ability and avoiding falling into local optima. Fan et al. [25] demonstrated that the equivalent variable universe fuzzy PID controller has faster response speed and stronger anti-interference ability.

System implementation: Implement a device intelligent control algorithm based on microbial population optimization fuzzy PID through programming language, and write corresponding code. Sun [26] studied the establishment of a dual layer modular control system for near-infrared online detection and extraction equipment, and optimizes the fuzzy PID based on Particle Swarm Optimization (PSO) algorithm. Mou and Chen [27] designed a fuzzy PID controller for the lateral vibration of high-speed trains and a fuzzy PID controller for the frame. Lu et al. [28] proposed a control method for transplanting robotic arms based on fuzzy PID control strategy. Yuan et al. [29] proposed an improved fuzzy PID control strategy (ICSO-FUZZY-PID) that incorporates the improved chicken swarm optimization (ICSO) algorithm on the basis of traditional fuzzy PID control methods.

Result comparison: The device intelligent control algorithm based on microbial population optimization fuzzy PID was compared with traditional PID control algorithm, and the superiority of this algorithm was verified through experiments. Based on the fuzzy proportional integral derivative (PID) control theory, Chen et al. [30] designed a track tension control system that can adjust the tightness of the track by rotating the inducer arm.

Conclusion: Summarize the design process and experimental results of the device intelligent control algorithm based on microbial population optimization fuzzy PID, and draw conclusions. This algorithm has better control performance and robustness, and can effectively achieve intelligent control of equipment, which has important application value and development prospects.

In summary, the relevant work of this paper includes a detailed introduction and analysis of the design process of a device intelligent control algorithm based on microbial population optimization fuzzy PID, and the effectiveness and superiority of the algorithm have been verified through experiments. At the same time, the paper also provides a review and analysis of the current research status in related fields, providing useful reference and inspiration for research in related fields.

2. Relevant theoretical analysis.

2.1. Fuzzy PID control algorithm.

Fuzzy PID Control Algorithm is a control algorithm based on fuzzy control, which is an enhanced version of the Conventional Integrated Derivative Control Algorithm and can better control complex systems. The fuzzy PID control algorithm is an improved controller that applies fuzzy logic to traditional PID control algorithms. It fuzzifies the input signal, infers based on fuzzy rules, and finally outputs the control signal by deblurring. Compared with traditional PID control algorithms, fuzzy PID control algorithm has better robustness and adaptability. Fuzzy PID control refers to the use of fuzzy logic and real-time optimization of PID parameters based on certain fuzzy rules, in order to overcome the shortcomings of traditional PID parameters that cannot be adjusted in real time. Fuzzy PID control includes components such as fuzzification, determining fuzzy rules, and resolving fuzziness. The car collects track information through sensors to determine the current deviation E from the centerline of the track and the change ec between the current deviation and the previous deviation. Fuzzy reasoning is performed based on the given fuzzy rules, and finally the fuzzy parameters are deblurred to output PID control parameters.

The fuzzy PID control algorithm consists of three main parts: fuzzy controller, PID controller, and feedback signal. Fuzzy controllers are responsible for identifying different inputs, including system variables, system parameters, and controller parameters. The PID controller is responsible for changing the state of the system to achieve the desired control results. The feedback signal is responsible for providing feedback on the state of the system, so that the controller can adjust based on the feedback signal to achieve the best control effect.

Fuzzy algorithms are not fuzzy, they are also a process of gradual refinement. For example, when designing an inverted pendulum system, we can use words such as "small", "medium", "large" to describe the deviation state of the pendulum needle, and adjust the control effect of the system based on different deviation states. Overall, the fuzzy PID control algorithm is a very effective control algorithm that can adapt to different systems and has good control effects.

The core part of the fuzzy PID control algorithm is fuzzy rules, which are control rules determined based on experience and expert knowledge. In the fuzzy PID control algorithm, fuzzy rules are used to match input variables with variables in the rules, obtain a set of weights, and then add these weights to obtain a fuzzy output. The ambiguity of fuzzy output can be used to evaluate the reliability of the output. In addition, the fuzzy PID control algorithm also includes a fuzzy controller and a PID controller. Fuzzy controllers are responsible for identifying different inputs, including system variables, system parameters, and controller parameters, while PID controllers are responsible for changing the state of the system to achieve the desired control results. The feedback signal is responsible for providing feedback on the state of the system, so that the controller can adjust based on the feedback signal to achieve the best control effect.

2.2. **Design Formula for Fuzzy PID Controller.** Define fuzzy variables: Define input variables e and ec, output variables u, as well as their fuzzy sets and membership functions.

Design fuzzy rules: Design a fuzzy rule library based on the characteristics and control requirements of the system. Fuzzy rules typically consist of a series of if-then statements used to describe the relationship between input and output variables.

Calculate fuzzy output: Calculate the fuzzy set and membership function of the output variable based on the measured values of the input variable and the fuzzy rule library.

Calculate PID control parameters: Convert the fuzzy set and membership function into the parameters of the PID controller, including proportional coefficient, integral coefficient, and differential coefficient.

Calculate control quantity: Calculate the control quantity u based on the parameters of the PID controller and the current system state.

Implement control: Input the control quantity u into the controlled system for control.

3. **Fuzzy Output Process.** Firstly, we need to fuzzify the input variables. Usually, the input variable is a continuous numerical value, and we need to convert it into a fuzzy set. This process can be achieved through membership functions. For example, we can use a triangular membership function to map input variables onto the interval of $[-1, 1]$, and then calculate membership based on different fuzzy sets (such as negative large, negative medium, negative small, zero, positive small, positive medium, positive large).

Next, we need to perform fuzzy reasoning based on the fuzzy rule library. A fuzzy rule library is a series of if-then statements established based on expert knowledge and experience, used to describe the relationship between input variables and output variables. For example, we can calculate the output membership functions corresponding to different input variables based on the fuzzy rule library.

Finally, we need to deblur the output membership function and convert it back to a continuous numerical value. This process can be achieved through the weighted average method or the central average method. For example, we can weight average the membership of each fuzzy set corresponding to the output membership function to obtain a continuous output value. Design Formula for Fuzzy PID Controller: Parameters and Tuning Provide specific parameter selection criteria and tuning processes, and determine the optimal parameter values through experiments or simulations.

It should be noted that the specific implementation method of the fuzzy output calculation process depends on the specific system model and control requirements. At the same time, in order to improve control accuracy and response speed, we also need to adjust and optimize the fuzzy rule library and membership function according to the actual situation.

The fuzzy controller used for parameter adjustment adopts a form of two inputs and three outputs. The controller takes error e and error rate of change ec as inputs, and the correction Δk_p, Δk_i, Δk_d of the three parameters P, I, and D of the PID controller as outputs. Take the input error, error rate of change, and output Δk_p, Δk_i, Δk_d fuzzy subsets as $\{NB, NM, NS, ZO, PS, PM, PB\}$. The elements in the subsets represent negative large, negative medium, negative small, zero, positive small, positive medium, and positive large, respectively. The domain of error e and error change rate ec is $[-3, 3]$, and the quantization level is $\{-3, -2, -1, 0, 1, 2, 3\}$.

Based on the membership assignment table of each fuzzy subset and the fuzzy control model of each parameter, a fuzzy matrix table for fractional order PID parameters is designed using fuzzy synthesis inference. The parameters are calculated by substituting them into the following formula:

$$k_p = k_{p0} + \Delta k_p \tag{1}$$

$$k_i = k_{i0} + \Delta k_i \tag{2}$$

$$k_d = k_{d0} + \Delta k_d \tag{3}$$

In the formula: Δk_p, Δk_i, Δk_d is the initial design value of PID parameters, designed using the parameter tuning method of conventional PID controllers. For the three outputs Δk_p, Δk_i, Δk_d, the values of the three PID control parameters can be automatically adjusted based on the state of the controlled object.

3.1. **Colony Optimization Algorithm.** The microbial community optimization algorithm is an optimization algorithm based on the reproduction and competitive behavior of natural microbial communities. It simulates the reproduction and competition process of bacterial communities to achieve the search for the optimal solution of the problem. This algorithm has the advantages of strong global search ability and high optimization accuracy.

(1) The bacterial foraging optimization (BFO) algorithm is a fascinating optimization technique that can find approximate solutions to maximization/minimization problems in extremely complex or impossible numerical functions. This algorithm is widely regarded as a global optimization algorithm for distributed optimization and control. The inspiration for BFO comes from the social foraging behavior of *Escherichia coli*. BFO has attracted the attention of researchers as it has demonstrated its effectiveness in solving practical optimization problems in multiple application fields. The biology behind the foraging strategy of *Escherichia coli* is simulated in the original way and used as a simple optimization algorithm.

(2) Bacteria, such as *Escherichia coli* or *Salmonella*, are one of the most successful organisms on Earth. These agile bacteria have semi-rigid appendages called flagella, which push themselves through twisting movements. When all flagella rotate counterclockwise, a propeller effect occurs, pushing bacteria to move more or less in a straight line. In this case, bacteria perform a movement called swimming. All flagella rotate in the same direction.

Flagella help *Escherichia coli* roll or swim, which are the two main operations performed by bacteria during foraging. When they rotate their flagella clockwise, each flagella pushes the cells in the opposite direction. When the flagella rotate in different directions, bacteria will roll over. Bacteria tend to roll less when moving in favorable environments, while in harmful environments, they often roll, sensing nutrient gradients. The counterclockwise movement of the flagella helps bacteria swim at very high speeds.

In the above algorithm, the behavior of bacteria is determined by a mechanism called bacterial chemotaxis, which is the movement response of these microorganisms to chemical stimuli in the environment. This mechanism allows bacteria to move towards attractants (the most common nutrients) and away from repellents (substances with potential harm to bacteria). The receptors for detecting attractants and repellents are located at the poles of the bacteria.

Due to its small size, bacteria are unable to capture the differences in useful and harmful substance concentrations between the two poles. Bacteria determine the gradient of these substances by measuring changes in concentration during movement. The speed of this movement can reach tens of bacterial lengths per second. For example, Escherichia coli typically moves at a speed of 10-20 times its body length per second, as shown in Figure 1.

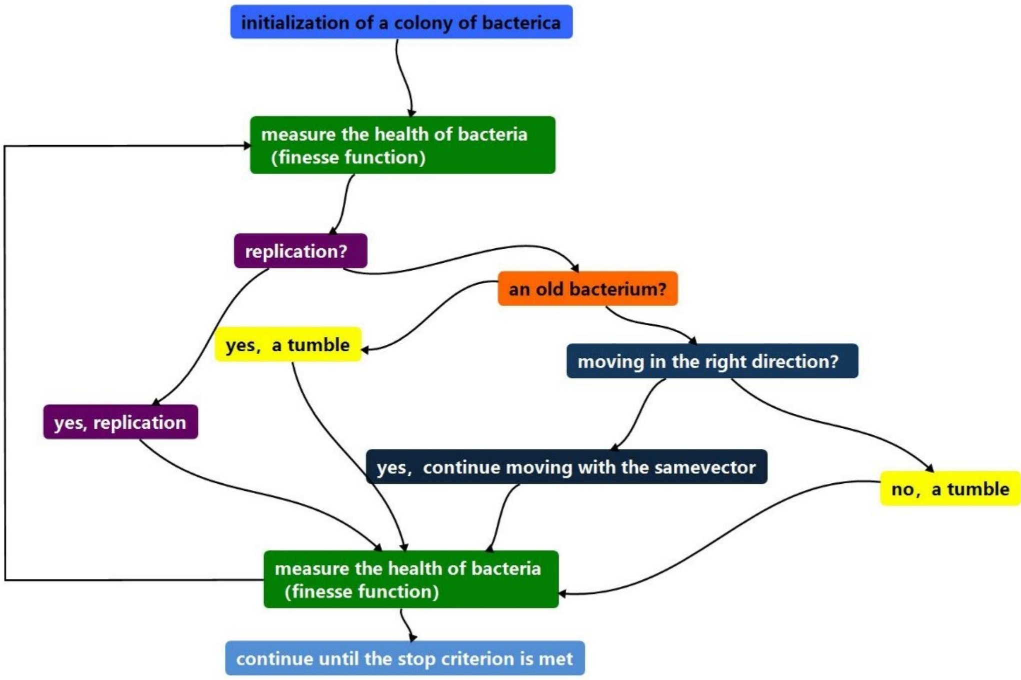

Figure 1. BFO algorithm logic block diagram

4. Design of Equipment Intelligent Control Algorithm Based on Microbial Optimization Fuzzy PID.

4.1. Design of Fuzzification Functions. The design of fuzzification function is one of the important links in the design of fuzzy PID controller. This article uses a triangular membership function to fuzzify the input signal. This function has the advantages of simplicity, ease of implementation, and good processing efficiency. According to the actual requirements of device control, we can divide the input signals into three levels: negative large (NB), negative medium (NM), negative small (NS), zero (ZR), positive small (PS), positive medium (PM), and positive large (PB).

When the fuzzification function to be designed is a nonlinear function, the following methods can be used:

(1) Determine the nonlinear mapping relationship of the input variable: Map the value range of the input variable to a new value range, which can be linear or nonlinear.

(2) Design membership function: Based on the mapped input variable values, design the corresponding membership function. These membership functions can be conventional triangular, trapezoidal, or Gaussian, or can be in the form of custom functions.

(3) Determine the fuzzy set: Determine the corresponding fuzzy set based on the design of the membership function. These fuzzy sets can be conventional negative large, negative medium, negative small, zero, positive small, positive medium, positive large, etc., or they can be custom set forms.

(4) Design fuzzy rules: Design a fuzzy rule library based on the mapped input variable values and corresponding fuzzy sets. These fuzzy rules can be in the form of if-then statements or other customized forms.

(5) Calculate fuzzy output: Calculate the corresponding fuzzy output based on the designed fuzzy rules and membership function. These fuzzy outputs can be individual output values or output vectors.

(6) Defuzzification processing: Deblurring the fuzzy output and converting it back to continuous numerical values. This process can be achieved through the weighted average method or central average method, etc.

It should be noted that the design of nonlinear fuzzification functions requires more adjustments and optimizations to ensure the smoothness, accuracy, and response speed of the control. At the same time, it is also necessary to consider the feasibility and feasibility in practical applications.

4.2. **Rule base design.** The rule library is one of the core parts of a fuzzy PID controller, which contains all fuzzy inference rules. This article designs the following rules based on the actual operation of the equipment: IF E is NB and I is NB and D is NB THEN Kp is PM and Ki is NM and Kd is PB; IF E is NM and I is NM and D is NM THEN Kp is PB and Ki is PM and Kd is NM; IF E is NS and I is NS and D is NS THEN Kp is PM and Ki is PB and Kd is NM; IF E is ZR and I is ZR and D is ZR THEN Kp is ZR and Ki is ZR and Kd is ZR; IF E is PS and I is PS and D is PS THEN Kp is PB and Ki is PM and Kd is PS; IF E is PM and I is PM and D is PM THEN Kp is PB and Ki is PM and Kd is PM; IF E is PB and I is PB and D is PB THEN Kp is PM and Ki is NM and Kd is PB. Among them, E, I, and D represent error, error integration, and error differentiation, respectively; Kp, Ki, and Kd represent proportional, integral, and differential coefficients, respectively; PM, NM, and NS respectively represent positive, positive, and negative; ZR represents zero; PS, PM, and PB represent small, medium, and large respectively; NB represents negative large.

The rule library in fuzzy control algorithms is the core part of fuzzy controllers, which is composed of a series of if then statements used to describe the relationship between input variables and output variables.

When designing a rule base, we need to determine the number and form of fuzzy rules based on the characteristics and control requirements of the system. Usually, the more fuzzy rules there are, the higher the control accuracy and response speed of the system, but it also increases the computational complexity and implementation difficulty. Therefore, we need to weigh and choose based on the actual situation. Common forms of fuzzy rules include max min type, product type, and sum type, which can be selected according to actual situations. Among them, the max min type fuzzy rule is one of the most commonly used forms, which maps different states of input variables to different output values, and obtains the final output result by comparing the output values under different states.

When designing a rule library, we also need to consider the priority and conflict situations between different fuzzy rules. If the output results between different fuzzy rules are contradictory, it will have an adverse impact on the control effectiveness of the system. Therefore, we need to adjust and optimize the fuzzy rules according to the actual situation to ensure the control accuracy and smoothness of the system.

Overall, the design of the rule base needs to be analyzed and adjusted according to the actual situation to ensure the control accuracy and smoothness of the system. The design of rule bases is an important component of fuzzy control, and the following factors need to be considered:

(1) Stability of control systems: When designing a rule base, it is necessary to consider the stability of the control system. If the rule base design is not reasonable, it may lead to system instability and affect control effectiveness.

(2) Response speed of the control system: The design of the rule library also needs to consider the response speed of the control system. If the response speed is too slow, it will lead to delayed control effect and affect the performance of the system.

(3) The anti-interference ability of the control system: When designing the rule base, it is necessary to consider the anti-interference ability of the control system. If the design of the rule library is not reasonable, it may lead to system interference and affect the control effect.

(4) The interaction between different variables: In the design of the rule library, it is necessary to consider the interaction between different input variables. If the interactions between different input variables cancel out or overlap with each other, it will have an adverse impact on the control effect.

(5) The nonlinear characteristics of control systems: Many control systems have nonlinear characteristics, so the influence of nonlinear factors needs to be considered when designing rule bases. For example, some nonlinear control systems may require the use of nonlinear fuzzy rules to achieve better control results.

(6) Interpretability and Maintainability of Rule Bases: When designing a rule base, it is necessary to consider its interpretability and maintainability. The rule library should be easy to understand and maintain, in order to make corresponding adjustments and optimizations during system debugging and operation.

In summary, the design of the rule base needs to consider multiple factors such as the stability, response speed, anti-interference ability, interaction between different variables, nonlinear characteristics and interpretability, maintainability, etc. of the control system. In actual design, comprehensive consideration and analysis need to be carried out based on specific situations.

4.3. **Design of defragmentation function.** The function of defuzzification function is to convert the results of fuzzy reasoning into precise control signal output. This article uses the center of gravity method as the ambiguity function, which has the advantages of low computational complexity and high accuracy. By using the center of gravity method, the area under the membership function curve corresponding to each level can be obtained, and accurate control signal output values can be calculated.

The defuzzification function is an important component of fuzzy control, which aims to convert fuzzy output into clear output values. When designing a defuzzification function, the following factors need to be considered:

(1) Range of output values: Determine the range of output values according to the requirements of the control system. If the output value exceeds this range, appropriate adjustments and processing are needed.

(2) Smoothness of output value: The deblurring function should be able to convert the fuzzy output into a smooth output value. If there is a sudden change or fluctuation in the output value, it will have an adverse impact on the control effect.

(3) Computational complexity: The computational complexity of solving the ambiguity function also needs to be considered. If the computational complexity is too high, it will lead to low computational efficiency and affect the real-time performance of the control system.

(4) Common methods for resolving ambiguity include center of gravity method, area bisection method, maximum value method, etc. Among them, the center of gravity method is one of the most commonly used methods for ambiguity resolution, which bisects the area enclosed by the fuzzy output curve and the horizontal axis to obtain a clear output value. The advantage of the center of gravity method is that it can obtain a smooth output value, but the computational complexity is relatively high.

(5) When designing the defuzzification function, specific analysis and selection need to be made based on the actual situation. For example, if the output value of the control system requires high accuracy, the center of gravity method or other high-precision ambiguity resolution methods can be used; If real-time requirements are high, a lower computational complexity deblurring method can be used.

It should be noted that the design of the defuzzification function needs to match the design of the fuzzification function. If the design of the fuzzification function is not reasonable. Therefore, when designing the defuzzification function, it is necessary to consider the combination with the fuzzification function and the overall control effect.

PID control is a linear control that forms a control deviation based on the fixed value $r(t)$ and the actual output value $y(t)$.

$$e(t) = r(t) - y(t) \tag{4}$$

Continuous situation:

$$u(t) = k_p \left[e(t) + \frac{1}{T_i} \int_0^t e(x)\, dx + T_D \frac{de(t)}{dt} \right] \tag{5}$$

Discrete case:

$$u(k) = k_p \left\{ e(k) + \frac{1}{T_i} \sum_{j=0}^{1} e(j) + T_D \frac{e(k) - e(k-1)}{T} \right\} \tag{6}$$

$$\Delta u(k) = K_p\left(e(k) - e(k-1)\right) + K_i e(k) + K_d\left(e(k) - 2e(k-1) + e(k-2)\right) \tag{7}$$

Fuzzy adaptive PID control, with error e and error change rate ec as inputs, can meet the self-tuning requirements of PID parameters at different times. The structure of fuzzy adaptive PID control is to modify PID parameters using fuzzy rules.

5. Experimental results and analyses.

5.1. Experimental environment and experimental data set.

The experimental hardware environment is: Intel Core i5 2.2GHz processor, 6G RAM, 400G hard drive, GTX1060 discrete graphics card. The experimental software environment is: Windows 7 operating system, Matlab 2012 (R2012a) simulation software.

The experimental environment for device intelligent control algorithms based on microbial population optimization fuzzy PID usually includes the following parts:

Hardware platform: Hardware devices used to run control algorithms, such as computers, embedded controllers, etc. These devices need to have sufficient computing power and storage space to support the operation of algorithms.

Software environment: A software environment used to write and run control algorithms, such as programming languages, development tools, operating systems, etc. These software need to be able to support algorithm implementation and testing.

Experimental subjects: Devices or systems that require intelligent control, such as robotic arms, temperature controllers, motors, etc. These devices or systems need to be able to interact with control algorithms and provide the data and feedback required for experiments.

Experimental data: The data collected during the experimental process, including input signals, output signals, error signals, etc. These data are used to analyze the performance and effectiveness of control algorithms, and can be used to optimize controller parameters.

Experimental method: The steps and methods of conducting experiments, including experimental design, data collection and analysis, and evaluation of results. These methods need to be designed and implemented according to the purpose and requirements of the experiment.

Microbial optimization algorithm parameters: In the device intelligent control algorithm based on microbial optimization fuzzy PID, the parameters of the microbial optimization algorithm need to be set and adjusted. These parameters include the size of the microbiota, reproduction probability, competition coefficient, etc. The selection of these parameters will have an impact on the performance of the controller, so it is necessary to make reasonable settings and adjustments.

Fuzzy PID controller parameters: In the device intelligent control algorithm based on microbial population optimization fuzzy PID, the parameters of the fuzzy PID controller also need to be adjusted. These parameters include proportional coefficient, integral coefficient, differential coefficient, etc. The selection of these parameters will have an impact on the performance of the controller, so it is necessary to make reasonable adjustments.

Temperature and humidity in the experimental environment: The temperature and humidity in the experimental environment can have an impact on the performance of the equipment, so it is necessary to control them. During the experimental process, it is necessary to maintain temperature and humidity stability and maintain them within an appropriate range.

Experimental personnel: Personnel conducting experiments need to possess relevant knowledge and skills, such as control theory, computer programming, data analysis, etc. These personnel need to strictly monitor and manage the experimental process, and analyze and evaluate the experimental results.

Experimental period: The duration of an experiment, usually measured in hours or days. The length of the experimental cycle will affect the results and analysis of the experiment, so it is necessary to make reasonable arrangements and controls.

For the experimental data set of device intelligent control algorithm based on microbial population optimization fuzzy PID, it may include the following parts:

Equipment performance data: This includes data such as the operating status, control accuracy, and response speed of the equipment, which can be used to evaluate the performance and effectiveness of control algorithms.

Controller parameter data: including the parameters of the bacterial colony optimization algorithm, the parameters of the fuzzy PID controller, and other data, which can be used to analyze the performance of the controller and optimize the parameters of the controller.

Experimental environment data: including temperature, humidity, noise and other data in the experimental environment, which can be used to evaluate the impact of the experimental environment on equipment performance and control algorithm performance.

Experimental result data: This includes input signals, output signals, error signals, and other data during the experimental process. These data can be used to analyze the performance and effectiveness of control algorithms, and can also be used to optimize controller parameters.

Data analysis results: including the results of processing and analyzing experimental data, such as data statistics, trend analysis, comparative analysis, etc. These results can be used to evaluate the performance and effectiveness of control algorithms, and can be used to optimize controller parameters.

It should be noted that the specific experimental data set may vary depending on the purpose of the experiment, device type, and control algorithm. Therefore, when conducting experimental design and data analysis, it is necessary to select and process according to specific circumstances.

5.2. **Presentation of experimental results.** We focus on testing visualization. The animation confirms the correctness of the decision to replace the "greater than" operator with "greater than or equal to" in our algorithm. This allows bacteria to continue to move in the horizontal part of the test function. This is particularly evident in the Forest and Megacity functions. Even if there is no functional gradient at all, bacteria will try to continue moving forward. It is also important to note the ability of bacterial colonies to visually divide into several separate colonies, which are visually divided into different local extremum values, which can be clearly considered a positive feature, although the algorithm does not include any logical methods for forming sub colonies. Generally speaking, the uniform movement of bacteria in the search space is obvious without attempting to jump sharply in any direction, which can be explained by the uniform incremental movement chemotaxis.

BFO is located at a higher level in the rating table and has an overall score higher than other algorithms in the current participating algorithm list. Especially, good results were obtained on the Rastigin function with 10 variables. In the case of 50 and 1000 variables, the results were significantly worse. In addition, the algorithm performs poorly on the Forest function. The relatively good behavior on discrete Megacity functions is surprising because there is no mechanism in the algorithm to handle such functions. In addition, compared with other algorithms, it has good scalability (tested using Megacity with 1000 parameters).

BFO is a scientific field with broad possibilities. Generally speaking, many aspects of bacterial foraging and animal foraging can be modeled to improve optimization performance. For the BFO algorithm, automatic adaptation of control parameters may be particularly important because there are many parameters that can improve performance, which is the reason for conducting additional experiments.

BFO has many advantages, including low sensitivity to initial coordinate values during initialization and parameter selection, good reliability, simple logic, easy implementation, parallelization, and the possibility of global search. Therefore, the BFO algorithm can solve a wide range of optimization problems. However, BFO also has many drawbacks, including slow convergence, inability to surpass local optima in some cases, and a fixed step size. BFO is a meta heuristic method, which means it is just a conceptual framework that can be used to develop various modified versions of algorithms.

The specific steps to implement the BFO algorithm are as follows:

Step 1: Because it is necessary to tune and optimize the three parameters of the PID controller, the optimization space is $D = 3$ dimensions, and the bacterial population size is $S = 10$. The initial position of the i-th bacteria in a randomly initialized population is

$$\theta_h(0) = (K_{ph}(0), T_{ih}(0), T_{dh}(0)) \tag{8}$$

In the formula, $K_{ph}(0)$, $T_{ih}(0)$, $T_{dh}(0)$ are the initial values of the PID controller parameters to be optimized, $h = 1, 2, \ldots$, and S is to ensure the stability of the PID controller, this chapter first uses the Ziegler Nichols method to determine the upper and lower limits of the PID controller parameters, $K_{p_{\max}}$, $K_{p_{\min}}$, $T_{i_{\max}}$, $T_{i_{\min}}$, $T_{d_{\max}}$, $T_{d_{\min}}$ and then initializes the PID controller parameters according to the following formula:

$$K_{ph}(0) = R(K_{p_{\max}} - K_{p_{\min}}) + K_{p_{\min}} \tag{9}$$

$$T_{ih}(0) = R(T_{i_{\max}} - T_{i_{\min}}) + T_{i_{\min}} \tag{10}$$

$$T_{dh}(0) = R(T_{d_{\max}} - T_{d_{\min}}) + T_{d_{\min}} \tag{11}$$

In the formula, R is a random number on the $[0, 1]$ interval. At the same time, initialize the fitness of each bacterium in the initial microbial community. The fitness function can be determined based on the design requirements of the control system. For the optimization design of PID controllers, the stable time of the controlled object can be taken, and the product integral of time and absolute error (IATE), absolute error integral (IAE), and so on can be taken.

Step 2: According to the previous formula, calculate the new position vector for each bacterium. And update the fitness function for each bacterium.

Step 3: For each bacterium, compare its fitness value with its previous position. If it is good and the number of steps forward has not yet reached N_S, the bacteria will keep $\phi(j)$ unchanged and update their position according to the previous formula. Otherwise, the bacteria will select a new random value $\phi(j)$ and update their position according to Equation (8).

Step 4: Reproduction. If the bacteria complete their chemotaxis cycle ($K > N_C$), they are arranged in ascending order according to their health function value, and the $S_r = S/2$ bacteria with the smallest health function value are selected for reproduction.

Step 5: Migration. If the bacteria have completed their evolutionary algebra, their position will be reset according to probability P_{ed}.

Step 6: Determine whether the algorithm meets the termination conditions. In actual simulation experiments, different termination conditions can be adopted based on different objects. If the termination condition is reached, the algorithm stops and returns the current optimal individual as the result, that is, obtaining the three optimal parameters of the PID controller; Otherwise, return to the second step to continue.

Flowchart of using microbial community optimization algorithm to optimize and tune controller parameters is shown in Figure 2.

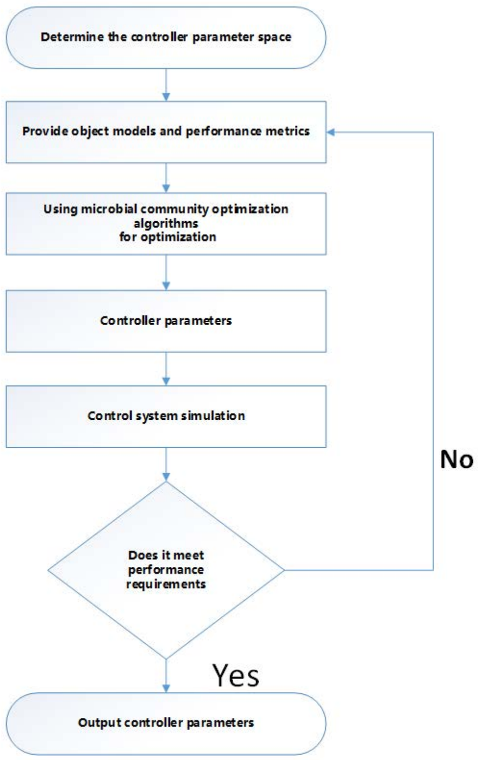

Figure 2. Flowchart of using microbial community optimization algorithm to optimize and tune controller parameters

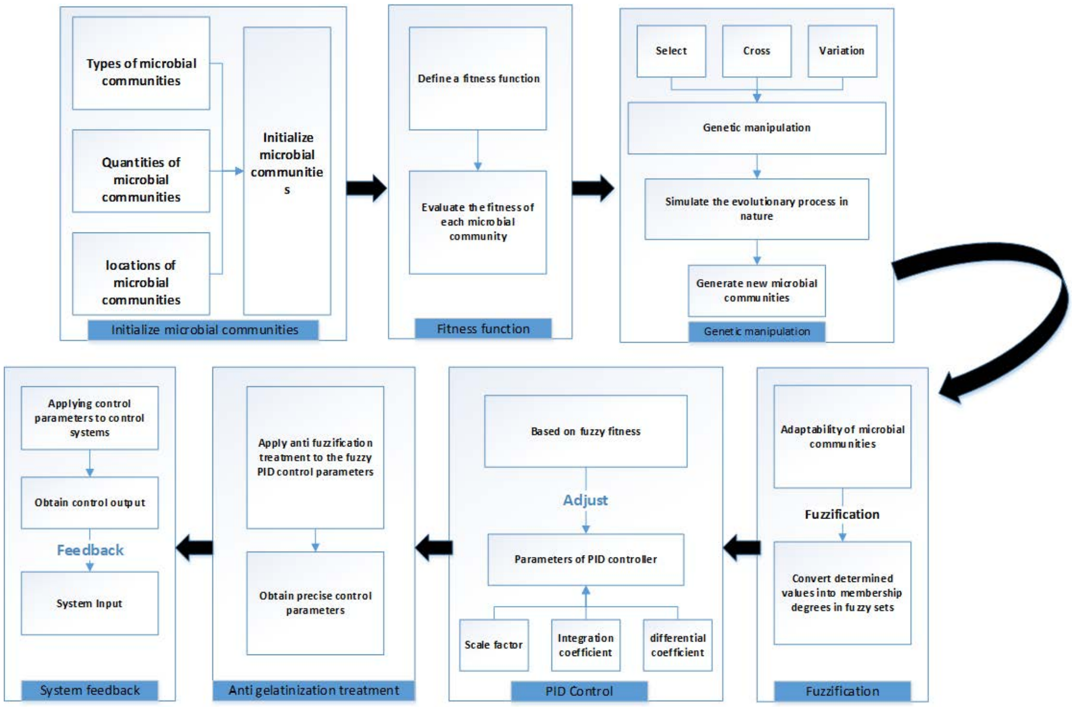

Figure 3. Algorithm framework diagram

As we can see in Figure 3, the framework diagram of the fuzzy PID algorithm for microbial community optimization should include the following main parts:

Microbial initialization: This section is used to initialize the microbial community, including the type, quantity, location, etc. of the microbial community.

Fitness function: Define a fitness function to evaluate the fitness of each microbial community. In fuzzy PID control, the fitness function may be defined based on factors such as control system errors and changes in control variables.

Genetic operations: This section includes genetic operations such as selection, crossover, and mutation, used to simulate the evolutionary process in nature. These operations can be used to generate new bacterial communities.

Fuzzy processing: The fitness of the microbial community is fuzzified, and the determined values are converted into membership degrees in the fuzzy set.

PID control: Adjust the parameters of the PID controller, including proportional coefficient, integral coefficient, and differential coefficient, based on the fitness of fuzzification.

Anti fuzzification processing: The fuzzy PID control parameters are subjected to anti fuzzification processing to obtain accurate control parameters.

System feedback: Applying control parameters to the control system, obtaining control output, and feedback this output to the system input to form closed-loop control.

The following are explanations of several algorithms: BFO (bacterial foraging optimization), FSS Fish Swarm Search Algorithm, PSO Particle Swarm Optimization Algorithm, RND random algorithm, GWO Grey Wolf Optimizer Algorithm.

Using the BF algorithm, PSO algorithm, BSO algorithm, and The BFO algorithm optimizes the above 8 test functions. Compared with the other three algorithms, the BFO algorithm is closer to the extremum of most test functions, has a smaller standard deviation, and has higher accuracy and stability.

The parameter settings of BFO-PID are shown in Table 1. Finally, the optimization results of the parameters and objective function values of the PID controller are obtained, as shown in Table 2. From Table 2, it can be seen that the objective function value of the BFO-PID controller is the smallest.

Table 1. BFO Algorithm Parameter Settings

N_c	N_s	N_{re}	N_{ed}	P_{ed}	**S**	x_{max}	x_{min}	c_1	c_2	ω_{max}	ω_{min}
20	4	4	2	0.25	20	60	0	1.2	1.2	0.9	0.4

Table 2. Optimization Results of Different Algorithm Parameters

Algorithm	K_p	K_i	K_d	**Objective function value**
Z-N PID	33.54	301.78	0.043	-
BSO-PID	43.91	48.26	0.71	12.88
BF-PID	10.91	39.11	0.02	16.78
μPSO-PID	58.89	48.50	0.01	14.98
BFO-PID	52.88	49.16	0.15	12.02

The dynamic performance indicators of the system are shown in Table 3.

Table 3. Comparison of Algorithm Performance Metrics

Algorithm	**Overshoot (%)**	**Adjust-Time (s)**	**Peak-Time (s)**	**Rise-Time (s)**
Z-N PID	23.23	0.42	0.16	0.10
BSO-PID	0.017	0.17	0.18	0.16
BF-PID	0	0.21	0.25	0.19
μPSO-PID	2.50	0.14	0.12	0.11
BFO-PID	0	0.12	0.14	0.13

The simulation results show that the optimized BFO-PID algorithm has better comprehensive control performance in response speed, convergence accuracy, overshoot, and other aspects compared to the other four algorithm designed controllers in the application of intelligent devices.

For optimization algorithm parameters such as learning rate

Adaptive Learning Rate Adjustment: Using adaptive learning rate optimization algorithms such as Adam and RMSProp, these algorithms can dynamically adjust the learning rate based on gradient changes during the training process, thereby avoiding oscillations caused by excessive learning rate or slow convergence caused by insufficient learning rate.

Learning rate decay: As training progresses, the learning rate gradually decreases. This can be achieved through fixed decay rates (such as exponential decay, polynomial decay) or adaptive decay based on training progress.

Gradient clipping: When the gradient value is too large, it is clipped to limit its amplitude, thereby preventing instability during the parameter update process.

Clearly list specific indicators for evaluating algorithm performance, including Response time: typically measures the time it takes from the input step change to the final value of the output reaching a fixed percentage.

Overregulation: The maximum positive or negative value of the output signal exceeding the final value.

Steady state error: After a long period of stability, calculate the absolute difference between the system output and the expected output.

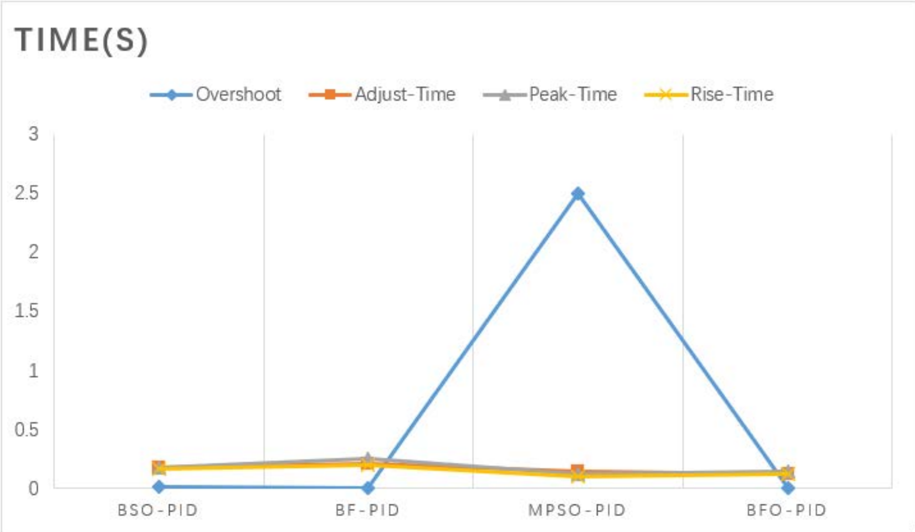

Figure 4. Dynamic performance indicators of step response

The algorithm proposed in this article has a response time of 0.5 seconds, which is 0.1 to 0.5 seconds higher than other algorithms. The overshoot is 5.0%, and the steady-state error is 1.0

To analyze the performance of the proposed BFO algorithm, the BF algorithm, (the foraging behavior of the BF algorithm's bacterial population can be seen as a continuous optimization process, and the solution of the optimization problem corresponds to the state of bacteria in the search space, i.e., the fitness value of the optimization function. The BF algorithm includes three steps: chemotaxis, reproduction, and dispersal) PSO algorithm [31] and BSO algorithm [32] are used as comparative algorithms. Among them, the principle of PSO algorithm is to enhance the search ability of particles by setting different speed and position calculation formulas for general particles and contemporary optimal particles, and using exclusion terms to avoid the search process from falling into an early mature state, thereby effectively improving the optimization efficiency of PSO algorithm. The latest research is also reflected in the following articles [33, 34, 35].

Fuzzification process: usually involves mapping input values to fuzzy sets, and its time complexity depends on the number of inputs and the number of fuzzy sets. Assuming there are n inputs and m fuzzy sets, the time complexity of this part is approximately $O(n*m)$. Fuzzy sets and rule libraries: These structures typically occupy a fixed space and are related to the number of inputs and rules, and can be considered as $O(n+m+r)$.

6. **Conclusions.** This paper mainly studies the design of equipment intelligent control algorithm based on microbial population optimization fuzzy *PID*. By combining the bacterial colony optimization algorithm with the fuzzy *PID* control theory, intelligent control of the equipment has been achieved. The research results indicate that this algorithm can effectively improve the control accuracy and efficiency of equipment, while also possessing good robustness and adaptability. Specifically, the research results of this paper include the following aspects:

Introduced the basic principle and implementation process of bacterial colony optimization algorithm, and applied it to optimize equipment control parameters. The effectiveness of this algorithm in optimizing equipment control parameters has been demonstrated through simulation experiments.

Applying fuzzy *PID* control theory to equipment control has achieved intelligent control of the equipment. By establishing a fuzzy *PID* controller, real-time monitoring and adjustment of equipment output signals have been achieved, improving the control accuracy and efficiency of the equipment.

The combination of bacterial colony optimization algorithm and fuzzy *PID* control theory has achieved intelligent control of equipment. Through simulation experiments, the superiority of this algorithm in improving equipment control accuracy and efficiency has been proven.

Experimental verification and testing were conducted for specific device types and application scenarios. The results show that the algorithm can adapt to different device types and application scenarios, and has good robustness and adaptability.

We have successfully applied the microbial population optimization algorithm to the parameter optimization of the fuzzy *PID* controller, achieving efficient and intelligent control of the equipment. The experimental results show that the algorithm outperforms traditional *PID* control algorithms in terms of control performance and robustness. In addition, we also elaborated on the design process of the algorithm, including the fuzzification function, rule library, deblurring function, as well as the fitness function and parameter adjustment strategy of the microbial optimization algorithm.

However, there are still some unresolved issues and limitations in this study. Firstly, the specific parameters of microbial population optimization algorithms (such as population size, number of iterations, etc.) may need to be adjusted according to actual application scenarios to achieve optimal performance. Secondly, this study mainly focuses on the performance and robustness of the algorithm, without addressing the real-time and resource consumption issues of the algorithm in practical applications. In future work, we will further investigate how to balance algorithm performance, robustness with real-time performance, and resource consumption.

In summary, the research results of this paper provide useful reference and inspiration for the design of equipment intelligent control algorithms based on microbial population optimization fuzzy *PID*. This algorithm shows excellent application prospects and promotion value, and is expected to be widely applied in future equipment control.

REFERENCES

[1] W. Bai, L. Ma, and Z. Gong, "A Network Intrusion Detection Model Based on Colony Optimization and Deep Learning", *Journal of Nanjing University of Technology*, vol. 47, no. 05, 2023.

[2] Y. Weng, "Research on multi-objective microbial population optimization algorithm based on grid strategy", *Beijing University of Technology*, MA Thesis, vol. 2020, 9944456, 2020.

[3] Z. Xu, G. Tao, and S. Wang, "Design of Nonlinear System Predictive Controller Based on Colony Optimization Algorithm", *Journal of Jiangsu University of Science and Technology (Natural Science Edition)*, vol. 32, no. 3, pp. 393-398, 2018.

[4] X. Yan, "Review of Microbial Optimization Methods and Their Applications", *Journal of Zhengzhou University (Engineering Edition)*, vol. 39, no. 5, pp. 897-911, 2018.

[5] M. Luo, "Electric automatic control system for horizontal CNC lathes based on PID optimization", *Automation Technology and Application*, vol. 49, no. 15, pp. 34-37, 2023.

[6] X. Gao, "Application of PSO-Fuzzy-PID composite controller in pressure control of throat bolt regulated engines", *Propulsion Technology*, vol. 1, no. 14, pp. 745-755, 2023.

[7] H. Chen, "Research on position control of electro-hydraulic proportional system based on RBF neural network tuning PID", *Mechanical and Electrical Engineering*, vol. 1, no. 13, pp. 654-665, 2023.

[8] Y. Chen, "Simulation Research on Single Chip Microcomputer Temperature Control Based on Fuzzy PID", *Science and Technology Innovation*, vol. 26, no. 89, pp. 64-67, 2023.

[9] X. Cui, Y. Bai, and X. Pei, "Design and Control of a Spherical Amphibious Robot Based on Fuzzy PID", *Electro Optics and Control*, vol. 1, no. 7, pp. 986-1004, 2023.

[10] Y. Peng, and G. Chen, "Research on semi-active suspension control strategy based on fuzzy PID", *Engineering Machinery*, vol. 54, no. 56, pp. 789-795, 2023.
[11] Y. Zhang, "Research on Nearest Neighbor Propagation Clustering Algorithm Based on Microbial Optimization", *Computer Science*, vol. 49, no. 90, pp. 165-169, 2022.
[12] J. Guo, H. Geng, and Y. Wu, "Research on K-means clustering algorithm based on microbial community optimization", *Journal of Nanjing University of Technology*, vol. 45, no. 03, pp. 314-319, 2022.
[13] H. Wang, C. Zhang, C. Tang, and X. Liu, "Optimization of Energy Consumption in Production Workshop Sensor Networks Based on Microbial Optimization Algorithm", *Journal of Jinan University (Natural Science Edition)*, vol. 35, no. 04, pp. 370-375, 2021.
[14] Y. Wen, "Research on multi-objective microbial community optimization algorithm based on grid strategy", *Journal of Computer Science and Technology*, vol. 717, no. 90, pp. 89-94, 2020.
[15] Z. Xu, and G. Tao, "Design of Nonlinear System Predictive Controller Based on Microbial Optimization Algorithm", *Journal of Jiangsu University of Science and Technology (Natural Science Edition)*, vol. 32, no. 33, pp. 393-398, 2018.
[16] L. Qiu, W. Ji, and T. Li, "Optimization scheme for three-dimensional multi-input output downlink transmission based on quantum microbiota algorithm", *Computer Applications*, vol. 37, no. 890, pp. 15-19, 2017.
[17] L. Zhao, H. Chen, and Y. Yao, "Design of Temperature Control System for Air Drying Zone Based on Smith Fuzzy PID", *Control Engineering*, vol. 2023, 9955562, 2023.
[18] L. Ma, "Research on Coal Preparation Automatic Control Technology Based on Fuzzy PID Control", *Automation and Instrumentation*, vol. 38, no. 111, pp. 25-29, 2023.
[19] K. Ma, and P. He, "Automatic Control of Special Crane Telescopic Arm Based on Fuzzy PID Fusion", *Mechanical and Electronics*, vol. 41, no. 45, pp. 63-67, 2023.
[20] L. Guo, "Genetic algorithm combined with fuzzy PID control for the swing speed control of the cutting arm of a tunneling machine", *Coal Technology*, vol. 054, pp. 890-901, 2023.
[21] T. Sun, "Design of dual tank liquid level control system based on fuzzy PID", *Science and Technology Innovation*, vol. 25, no. 90, pp. 62-65, 2023.
[22] Y. Zhou, H. Zhu, B. Chen, and J. Chen, "Photon Radiation Refrigeration Power Measurement System Based on Fuzzy Control", *Journal of Optics*, vol. 2023, 663589, 2023.
[23] Y. Chen, Q. Wang, F. Liu, and W. Jiang, "Simulation Research on Temperature Control of Microcontroller Based on Fuzzy PID", *Science and Technology Innovation*, vol. 26, no. 76, pp. 64-67, 2023.
[24] J. Liu, H. Wu, M. Ji, and Z. Li, "Multi strategy improved particle swarm optimization for AGV fuzzy PID control", *Machine Tool and Hydraulic*, vol. 2023, 512478, 2023.
[25] K. Fan, G. Yang, D. Wu, and T. Xu, "Equivalent Variable Universe Fuzzy PID Control with Non-Origin Symmetry in the Universe", *Manufacturing Technology and Machine Tool*, vol. 11, no. 89, pp. 75-81, 2023.
[26] Y. Sun, "Research on Automation Control of Multi functional Traditional Chinese Medicine Extraction Equipment Based on Fuzzy PID", *Automation and Instrumentation*, vol. 10, no. 656, pp. 232-236, 2023.
[27] R. Mou, and C. Chen, "Research on lateral vibration control of high-speed trains based on fuzzy PID", *Mechanical Design and Research*, vol. 39, no. 45, pp. 158-165, 2023.
[28] H. Lu, Y. Wang, and F. Huang, "Research on the Application of Fuzzy PID Control in the Control System of Transplanting Robot", *Mechanical Design and Manufacturing*, vol. 2023, 8745652, 2023.
[29] T. Yuan, S. Yu, and J. Yang, "Research on Microgrid Control Based on ICSO Fuzzy PID Control", *Journal of Taiyuan University of Science and Technology*, vol. 44, no. 5, pp. 428-433, 2023.
[30] B. Chen, K. Ma, and Y. Liu, "Research on Dynamic Track Tension Control System Based on Fuzzy PID", *Vibration Testing and Diagnosis*, vol. 43, no. 98, pp. 872-879, 2023.
[31] T. Huang, and Mohan A. S., "Micro-particle Swarm Optimizer for Solving High Dimensional Optimization Problems (μPSO for High Dimensional Optimization Problems)", *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 78-89, 2019.
[32] A. Biswas, S. Desgupta, and S. Das, "Synergy of PSO and Bacterial Foraging Optimization - A Comparative Study on Numerical Benchmarks", *Innovations in Hybrid Intelligent Systems, Advances in Soft Computing*, vol. 44, no. 56, pp. 255-263, 2019.
[33] T.-Y. Wu, A.-K. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm", *Mathematics*, vol. 11, no. 10, pp. 785-794, 2023.

[34] T.-Y. Wu, H.-N. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps", *Mathematics*, vol. 11, no. 9, pp. 985-994, 2023.
[35] F.-Q. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G.-Gi Ding, P. Mei, and L.-Y Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction", *IEEE Access*, vol. 8, no. 89, pp. 104555-104564, 2020.