scientific reports

Check for updates

OPEN

PID control algorithm based on multistrategy enhanced dung beetle optimizer and back propagation neural network for DC motor control

Weibin Kong[1], Haonan Zhang[1], Xiaofang Yang[1], Zijian Yao[1]✉, Rugang Wang[1], Wenwen Yang[2] & Jiachen Zhang[1]

Traditional Proportional-Integral-Derivative (PID) control systems often encounter challenges related to nonlinearity and time-variability. Original dung beetle optimizer (DBO) offers fast convergence and strong local exploitation capabilities. However, they are limited by poor exploration capabilities, imbalance between exploration and exploitation phases, and insufficient precision in global search. This paper proposes a novel adaptive PID control algorithm based on enhanced dung beetle optimizer (EDBO) and back propagation neural network (BPNN). Firstly, the diversity of exploration is increased by incorporating a merit-oriented mechanism into the rolling behavior. Then, a sine learning factor is introduced to balance the global exploration and local exploitation capabilities. Additionally, a dynamic spiral search strategy and adaptive t-distribution disturbance are presented to enhance search precision and global search capability. The BPNN is employed to fine-tune both PID and network parameters, leveraging its powerful generalization and learning ability to model nonlinear system dynamics. In the simplified motor experiments, the proposed controller achieved the lowest overshoot (0.5%) and the shortest response time (0.012 s), with a settling time of 0.02 s and a steady-state error of just 0.0010. In another set of experiments, the proposed controller recorded an overshoot and response time of 0.7% and 0.0010 s, across five DC motor tests. These results demonstrate the proposed adaptive PID control algorithm has superior performance in optimizing control system parameters, as well as improving system robustness and stability.

Keywords Heuristic algorithms, Neural networks, Optimization methods, Proportional control, Parameter estimation

Literature review

The PID control algorithm has been used in industrial process control applications for many years[1]. Although the PID control algorithm has existed for a long time[2], it is still the most popular control algorithm in the process and manufacturing industry today[3]. While effective in various industrial applications[4], the traditional PID algorithm faces challenges with complex and nonlinear systems[5]. Parameter tuning often relies on trial and error[6]. Additionally, it lacks stability and robustness against disturbances and parameter changes[7]. In the context of DC motor control, traditional PI controllers have been widely applied and studied. However, traditional PI controllers also have limitations in managing nonlinearity and dynamic uncertainty, such as overshoot, response time delay, and difficulty in adjusting control parameters. These issues become more prominent in complex systems with different loads and speeds[8].

Several studies have focused on improving the structure of the traditional PID controller by integrating neural networks and other advanced methodologies. Zhu et al. developed a hybrid-optimized BP neural network PID controller for agricultural applications, improving control performance by enhancing the efficiency of initial weights[9]. Maraba and Kuzucuoglu introduced a PID neural network controller for speed control of

[1]School of Information Engineering, Research Center of Photoelectric and Information Technology, Yancheng Institute of Technology, Yancheng 224000, Jiangsu, China. [2]School of Information Science and Technology, Nantong University, Nantong 226000, Jiangsu, China. ✉email: yzjtz@ycit.edu.cn

asynchronous motors, combining the benefits of artificial neural networks with the strengths of the classic PID controller[10]. Cong and Liang developed a nonlinear adaptive PID-like neural network controller using a mix of locally recurrent neural networks, enhancing the controller's ability to adapt to nonlinear and time-variant system dynamics[11]. Ambroziak and Chojecki designed a PID controller optimized for air handling units (AHUs) by combining nonlinear autoregressive models, fuzzy logic, and FST-PSO metaheuristics, achieving superior performance in HVAC systems[12]. Aygun et al. presented a PSO-PID controller for regulating the bed temperature in a circulating fluidized bed boiler, leveraging the strengths of particle swarm optimization (PSO) to improve control precision[13]. Dahiya et al. proposed a hybrid gravitational search algorithm optimized PID and fractional-order PID (FOPID) controller to address the automatic generation control problem, demonstrating enhanced system stability and performance[14]. In recent years, several advanced PID variants have also been developed to handle nonlinear and complex systems. Suid and Ahmad designed a sigmoid-based PID (SPID) controller for the Automatic Voltage Regulator (AVR) system, employing a Nonlinear Sine Cosine Algorithm (NSCA) to optimize its parameters, which significantly improved the transient response and steady-state errors of the AVR system[15]. Sahin et al. proposed a sigmoid-based fractional-order PID (SFOPID) controller for the Automatic Voltage Regulator (AVR) system[16]. Ghazali et al. proposed a multiple-node hormone regulation neuroendocrine-PID (MnHR-NEPID) controller for nonlinear MIMO systems, improving control accuracy by introducing interactions between hormone regulation nodes based on adaptive safe experimentation dynamics (ASED)[17]. Kumar and Hote proposed a PIDA controller design using an improved coefficient diagram method (CDM) for load frequency control (LFC) of an isolated microgrid, enhancing control stability through maximum sensitivity constraints[18].

Various optimization tools have been developed to dynamically adjust PID parameters for improved control performance. Shi et al. introduced the RBF-NPID algorithm, utilizing radial basis function (RBF) neural networks to dynamically adjust PID parameters, leading to more effective contrsol in complex systems[19]. Hanna et al. developed an adaptive PID algorithm (APIDC-QNN) that uses quantum neural networks combined with Lyapunov stability criteria for stable parameter optimization, enhancing system robustness[20]. Zhao and Gu proposed an adaptive PID method for car suspensions, where a radial basis function neural network is used to fine-tune the PID parameters, improving ride quality and suspension control[21]. Kebari et al. optimized PID parameter values based on real-time task demand and the cumulative sum of previous demands, providing a more responsive control system[22]. Similarly, Gupta et al. employed a hybrid swarm intelligence algorithm to adjust PID gains for stabilizing the active magnetic bearing (AMB) system under unstable conditions[23]. Faria et al. found that a PSO-based PID tuning strategy offers a practical solution for enhancing the effectiveness of radiofrequency ablation (RFA) techniques, demonstrating the utility of swarm intelligence in medical applications[24]. Nanyan et al. proposed an improved Sine Cosine Algorithm (ISCA) to optimize PID controllers for DC-DC buck converters, demonstrating enhanced transient response and robustness compared to traditional algorithms[25]. Mourtas et al. utilized the beetle antennae search (BAS) algorithm for robust tuning of PID controllers, achieving superior performance in stabilizing feedback control systems with significantly reduced computational time[26]. Ghith and Tolba introduced a hybrid Arithmetic Optimization Algorithm (AOA) and Artificial Gorilla Troop Optimization (GTO) for tuning PID controllers in micro-robotics systems[27].

Recent swarm intelligence algorithms[28–33] have shown superior performance compared to traditional genetic algorithms[34] and particle swarm optimization[35]. These novel algorithms effectively balance global exploration and local exploitation, featuring fast convergence and high solution accuracy, making them suitable for optimizing PID algorithm parameters driven by neural network models[36–39]. While significant advancements have been made in PID variants like the sigmoid-based fractional-order PID[16] and multiple-node hormone regulation neuroendocrine-PID (MnHR-NEPID)[17], which excel in specific application scenarios, the proposed control algorithm extends these capabilities by providing dynamic, real-time adjustments in complex control systems. This combination of enhanced search strategies and neural network integration allows for superior performance in scenarios that demand rapid responses and robust stability under parameter variations. The enhanced DBO incorporates a novel search state adjustment mechanism oriented towards preferred positions to enhance exploration diversity. It also introduces a sinusoidal learning factor to balance global exploration and local exploitation and adopts a dynamic spiral search method to improve search efficiency. To enhance population diversity and search accuracy, an adaptive t-distribution disturbance is used for more effective global search capabilities. In comparison to the conventional PI controllers, which often rely on fixed parameter settings and exhibit sensitivity to parameter uncertainties[8], the proposed PID control algorithm offers dynamic adjustment capabilities and improved robustness. Through the incorporation of adaptive tuning mechanisms such as the merit-oriented search and sine learning factor, our approach addresses the overshoot, long response times, and tuning difficulties associated with traditional PI control methods. Additionally, the hybrid integration of the enhanced Dung Beetle Optimizer (EDBO) with Back Propagation Neural Network (BPNN) allows the system to handle nonlinear system dynamics more effectively, achieving faster response and lower steady-state errors in complex control systems such as DC motors.

Innovative contributions

Many controllers have been developed for industrial control processes in the existing literature and this study introduces new aspects that set it apart from the rest of the literature. The main contributions of this paper are as follows:

- Hybrid optimization algorithm framework. This study proposes a unique combination of the enhanced Dung Beetle Optimizer (EDBO) and Back Propagation Neural Network (BPNN) to optimize PID control parameters. This hybrid approach leverages the strengths of both heuristic algorithms and neural networks to enhance the optimization process.

- Development of an enhanced Dung Beetle Optimizer. This study develops an enhanced DBO with a novel multi-strategy combined location update mechanism that addresses the limitations of traditional Dung Beetle Optimizer (DBO) by improving global search capability, local exploitation, and search precision.
- Multistrategy innovations. EDBO incorporates a merit-oriented mechanism, a sine learning factor, a dynamic spiral search strategy, and adaptive t-distribution disturbance. These improvements ensure that EDBO is efficient and reliable when addressing complex optimization problems.

BPNN PID control algorithm

PID control

As a linear control method, PID control has the advantages of simple structure and easy implementation. The PID control law consists of three links: proportional control, integral control, and differential control, which is expressed as follows:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right] \tag{1}$$

where K_p is the proportional coefficient, T_i is the integration time constant, and T_d is the differential time constant. $e(t)$ is the deviation signal of the system. $u(t)$ is the control quantity of the system.

In actual operation, the control law is usually implemented using the incremental PID control algorithm. The basic structure of the PID control system is shown in Fig. 1.

The incremental PID control algorithm is expressed as follows:

$$\begin{aligned} \Delta u(k) = {} & K_p \left[e(k) - e(k-1) \right] + K_i e(k) + \\ & K_d \left[e(k) - 2e(k-1) + e(k-2) \right] \end{aligned} \tag{2}$$

where $e(k)$ is the deviation value of the control system at the k-th sampling time. $e(k-1)$is the deviation value of the control system at the $k-1$st sampling time. K_p is the proportional coefficient. K_i is the integral coefficient, and K_d is the differential coefficient. $\Delta u(k)$ is the difference between the control quantity at the k -th sampling time and the $k-1$st sampling time. The PID control law's performance heavily depends on the tuning of its parameters. Traditional tuning methods, while effective, often require trade-offs between stability, responsiveness, and robustness.

According to the basic principles of PID control, proportional control accelerates the system's response by reflecting the magnitude of the current error, enabling real-time adjustments that bring the system closer to the desired value. Integral control reduces accumulated past errors, thereby eliminating steady-state errors. The derivative term predicts future error trends, accelerating response times and minimizing overshoot. These three control actions combine to form a closed-loop PID control, which stabilizes the system. Accurate tuning of these parameters is key to ensuring controller stability. Therefore, this study focuses on optimizing these parameters to enhance system robustness and stability.

BPNN PID control

Back propagation neural network (BPNN) is one of the most utilized neural network models currently. BPNN deals with the arbitrary nonlinear relation between input and output variables by simulating the human brain's intelligence. It shows incomparable advantages in complex model fitting and distribution approximation over the traditional statistical methods. The underlying reason may be that BPNN has self-learning and generalization ability, which constitute its high prediction accuracy, simple structure, self-organization, and self-adaptive capabilities. BP neural network can learn the system performance according to the operating status of the controlled system to achieve optimal PID control. The block diagram of the PID control structure based on BPNN is shown in Fig. 2.

The input signal $r(k)$, deviation signal $e(k)$, and actual output signal $y(k)$ of the control system are used as the input of the BPNN. After being trained by the BPNN, the three parameters K_p, K_i and K_d are sent to the control system. Then, the $u(k)$ is sent to the controlled object to realize the real-time online adjustment of three parameters. The BPNN structure is shown in Fig. 3.

In the figure, j represents the input layer, i represents the hidden layer, and l represents the output layer. The input layer neurons are $r(k)$, $e(k)$, and $y(k)$. The output layer neurons are the three parameters of PID control K_p , K_i and K_d. The output of each neuron in the input layer of the neural network $O_j^{(1)}(k)$ is expressed as follows:

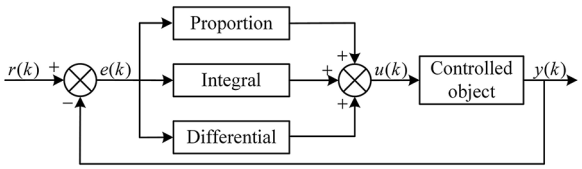

Fig. 1. Basic structure of PID control system.

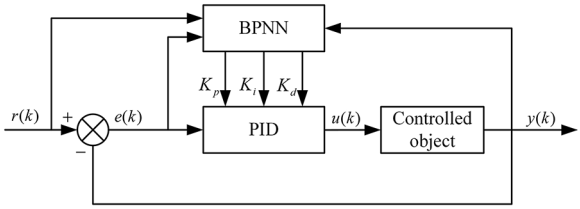

Fig. 2. PID control structure based on BPNN.

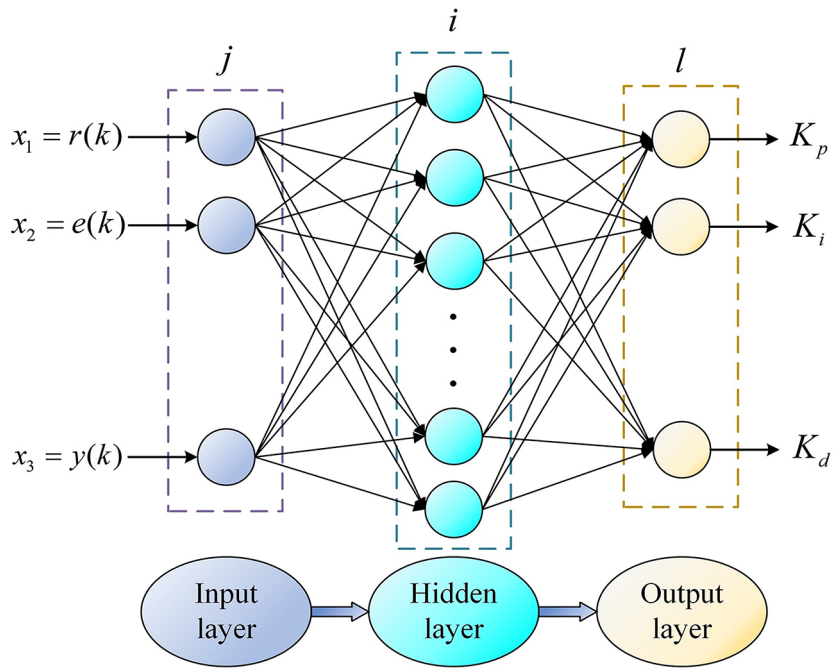

Fig. 3. Design of BPNN structure.

$$O_j^{(1)}(k) = x(j)\ j = 1, 2, 3 \tag{3}$$

The input $net_i^{(2)}(k)$ and output $O_i^{(2)}(k)$ of each neuron in the hidden layer are represented as follows:

$$\begin{cases} net_i^{(2)}(k) = \sum_{j=1}^{3} w_{ij}{}^{(2)} O_j^{(1)}(k) \\ O_i^{(2)}(k) = f(net_i^{(2)}(k))\ i = 1, 2, \cdots, 8 \end{cases} \tag{4}$$

where $w_{ij}{}^{(2)}$ is the weight connecting the input layer neurons and the hidden layer neurons. $f(x)$ is the activation function in the hidden layer, which is expressed as follows:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5}$$

The input $net_l^{(3)}(k)$ and output $O_l^{(3)}(k)$ of each neuron in the output layer are represented as follows:

$$\begin{cases} net_l^{(3)}(k) = \sum_{i=1}^{8} w_{li}^{(3)} O_i^{(2)}(k) \\ O_l^{(3)}(k) = g(net_l^{(3)}(k))\ l = 1, 2, 3 \end{cases} \tag{6}$$

$$\begin{cases} O_1^{(3)}(k) = K_p \\ O_2^{(3)}(k) = K_i \\ O_3^{(3)}(k) = K_d \end{cases} \tag{7}$$

where $w_{li}^{(3)}$ is the weight connecting the hidden layer neurons and the output layer neurons. $g(x)$ is the activation function in the output layer.

The three output nodes of the output layer respectively correspond to K_p, K_i and K_d. The activation function $g(x)$ is expressed as follows:

$$g(x) = \frac{1}{2}(1 + \tanh(x)) = \frac{e^x}{e^x + e^{-x}} \tag{8}$$

The performance index function of the control system is defined as follows:

$$E(k) = \frac{1}{2}(r(k) - y(k))^2 \tag{9}$$

The weights of the neural network are adjusted using the gradient descent method, which iteratively updates the weights to minimize the error between the predicted and actual outputs.

Additionally, an inertia term is included in the weight adjustment process to accelerate convergence. The weight adjustment from the hidden layer to the output layer can be expressed as follows:

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \alpha \Delta w_{li}^{(3)}(k-1) \tag{10}$$

where η is the learning rate and α is the inertia coefficient. According to the chain rule of derivatives, the gradient descent can be expressed as follows:

$$\begin{aligned} \frac{\partial E(k)}{\partial w_{li}^{(3)}} &= \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_l^{(3)}(k)} \\ &\cdot \frac{\partial O_l^{(3)}(k)}{\partial net_l^{(3)}(k)} \cdot \frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}} \end{aligned} \tag{11}$$

$$\begin{cases} \dfrac{\partial u(k)}{\partial O_1^{(3)}(k)} = e(k) - e(k-1) \\ \dfrac{\partial u(k)}{\partial O_2^{(3)}(k)} = e(k) \\ \dfrac{\partial u(k)}{\partial O_3^{(3)}(k)} = e(k) - 2e(k-1) + e(k-2) \end{cases} \tag{12}$$

Finally, the learning algorithm for the weights of the output layer and hidden layer is obtained as follows:

$$\begin{cases} \Delta w_{li}^{(3)}(k) = \alpha \Delta w_{li}^{(3)}(k-1) + \eta \delta_l^{(3)} O_i^{(2)}(k) \\ \delta_l^{(3)} = e(k)\,\mathrm{sgn}\left(\dfrac{\partial y(k)}{\partial u(k)}\right) \dfrac{\partial u(k)}{\partial O_l^{(3)}(k)} g'(net_l^{(3)}(k)) \\ \Delta w_{ij}^{(2)}(k) = \alpha \Delta w_{ij}^{(2)}(k-1) + \eta \delta_l^{(2)} O_j^{(1)}(k) \\ \delta_l^{(2)} = f'(net_i^{(2)}(k)) \sum_{l=1}^{3} \delta_l^{(3)} w_{li}^{(3)}(k) \end{cases} \tag{13}$$

where $g'(x) = g(x)(1 - g(x))$, $f'(x) = (1 - f^2(x))/2$, η is the learning rate of neural network and α is the inertia coefficient of neural network.

Dung beetle optimizer

Basic dung beetle optimizer

The dung beetle optimizer (DBO) is a novel population based technique. It is inspired by the ball-rolling, dancing, foraging, stealing, and reproduction behaviors of dung beetles. The DBO algorithm considers both global exploration and local exploitation, thereby having the characteristics of a fast convergence rate and satisfactory solution accuracy in complex optimization problems across various research domains.

(1) Rolling behavior.

During the rolling process, the position of the ball-rolling dung beetle is updated and can be expressed as follows:

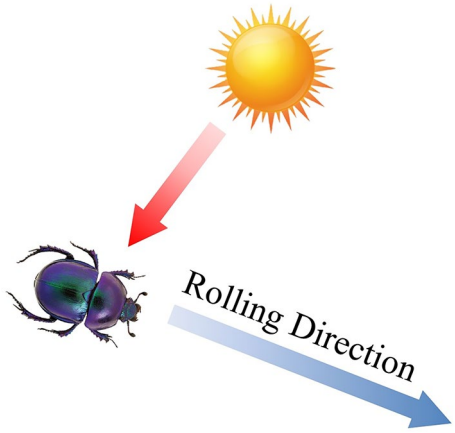

Fig. 4. Conceptual model of rolling behavior.

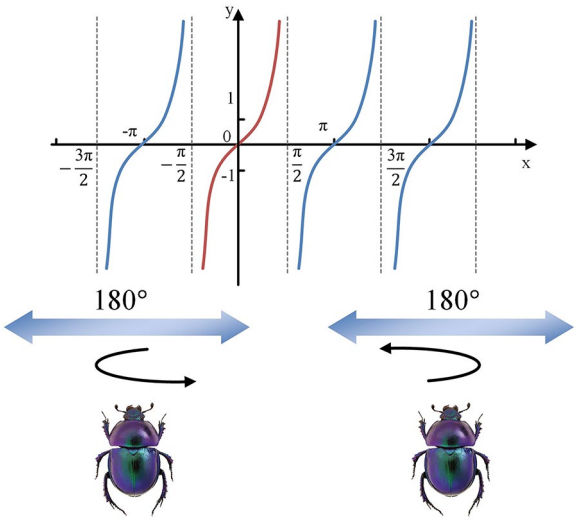

Fig. 5. Conceptual model of the tangent function and the dance behavior.

$$\begin{cases} x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \\ \Delta x = |x_i(t) - X^w| \end{cases} \tag{14}$$

where t represents the current iteration number, $x_i(t)$ denotes the position information of the ith dung beetle at the tth iteration, $k \in (0, 0.2]$ denotes a constant value which indicates the deflection coefficient, $b$indicates a constant value belonging to (0, 1), α is a natural coefficient which is assigned -1 or 1, X^w indicates the global worst position, Δx is used to simulate changes of light intensity. A conceptual model of a dung beetle's rolling behavior is shown in Fig. 4.

(2) Dancing behavior.

To mimic the dance behavior, the tangent function is used to get the new rolling direction. The position of the ball-rolling dung beetle is updated and defined as follows:

$$x_i(t+1) = x_i(t) + \tan(\theta)|x_i(t) - x_i(t-1)| \tag{15}$$

where θ is the deflection angle belonging to $[0, \pi]$. Conceptual model of the tangent function and the dance behavior of a dung beetle is shown in Fig. 5. Once the dung beetle has successfully determined a new orientation, it should continue to roll the ball backward.

(3) Reproduction behavior.

A boundary selection strategy is proposed to simulate the areas where female dung beetles lay their eggs, which is defined by:

$$\begin{cases} Lb^* = \max(X^* \times (1-R), Lb) \\ Ub^* = \min(X^* \times (1+R), Ub) \end{cases} \tag{16}$$

where X^* denotes the current local best position, $Lb^*$and Ub^* mean the lower and upper bounds of the spawning area respectively, $R = 1 - t/T_{\max}$ and $T_{\max}$ indicate the maximum iteration number, Lb and Ub represent the lower and upper bounds of the optimization problem, respectively. As shown in Fig. 6, the current local best position X^* is indicated by using a large circle, while the small circles around X^* indicate the brood balls. In addition, the small red circles represent the upper and lower bounds of the boundary.

The position of the brood ball is also dynamic in the iteration process, which is defined by:

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \tag{17}$$

where $B_i(t)$ is the position information of the ith brood ball at the tth iteration, b_1 and$b_2$represent two independent random vectors by size $1 \times D$, D indicates the dimension of the optimization problem.

(4) Foraging behavior.

The boundary of the optimal foraging area is defined as follows:

The position of the brood ball is also dynamic in the iteration process, which is defined by:

$$\begin{cases} Lb^b = \max(X^b \times (1-R), Lb) \\ Ub^b = \min(X^b \times (1+R), Ub) \end{cases} \tag{18}$$

where X^b denotes the global best position, Lb^b and Ub^b mean the lower and upper bounds of the optimal foraging area respectively. Therefore, the position of the small dung beetle is updated as follows:

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b) \tag{19}$$

where $x_i(t)$ indicates the position information of the ith small dung beetle at the tth iteration, C_1 represents a random number that follows normally distributed, and C_2 denotes a random vector belonging to (0, 1).

(5) Stealing behavior.

During the iteration process, the position information of the thief is updated and can be described as follows:

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*| + |x_i(t) - X^b|) \tag{20}$$

where $x_i(t)$ denotes the position information of the ith thief at the ith iteration, g is a random vector by size $1 \times D$ that follows normally distributed, and S indicates a constant value.

Enhanced dung beetle optimizer

While the Basic Dung Beetle Optimizer (DBO) provides a foundation for global exploration and local exploitation, certain limitations such as susceptibility to local optima require enhancement for complex optimization tasks. To address these limitations, the Enhanced Dung Beetle Optimizer (EDBO) introduces advanced strategies that significantly improve performance and reliability.

The rolling behavior plays an important role in the DBO algorithm, but the dung beetle cannot communicate with other dung beetles when rolling the ball, which makes it easy to fall into a local optimal situation. Based on the original rolling ball behavior, a merit-oriented mechanism is introduced. When the dung beetle rolls the ball, it will select a nearby excellent position based on the current position as a guide for the rolling ball path. The new position of the ball-rolling dung beetle is defined as:

$$x_i(t+1) = x_i(t) + h \times (x_i^s(t) - I \times x_i(t)) \tag{21}$$

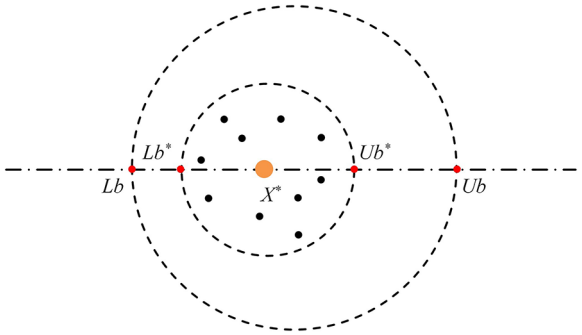

Fig. 6. Conceptual model of the boundary selection strategy.

where h is a uniformly distributed random number with $h \in (0, 1)$. x_i^s denotes the excellent position information of the ith dung beetle at the tth iteration, and i is a random number in the set $\{1, 2\}$. The conceptual model of merit-oriented mechanism is shown in Fig. 7. The blue arrow indicates the final rolling direction.

Dancing behavior is an important exploration mechanism. To balance the ability of global exploration and local development, a sine learning factor r is introduced in the position update of dancing behavior. In the early iteration, the sine factor r can promote global exploration, whereas in the later iteration, it helps to refine local development and optimize search behavior at different stages. The sine learning factor and the new position equation for the dancing behavior are defined as:

$$r = r_{\min} + (r_{\max} - r_{\min}) \times \sin(\frac{\pi t}{T_{\max}}) \tag{22}$$

$$x_i(t+1) = r \times x_i(t) + (1 - r)\tan(\theta)|x_i(t) - x_i(t-1)| \tag{23}$$

where $r_{\max}$ is the maximum value. $r_{\min}$ is the minimum value. t represents the current iteration number, and $T_{\max}$ represents the maximum iteration number.

To ensure the convergence speed of the algorithm and increase the diversity of individuals in the population, a dynamic spiral search method is proposed to improve the original breeding behavior. As the iteration proceeds, the shape of the spiral changes dynamically from large to small, and the dynamic nonlinear search mode helps to improve the population diversity and search accuracy of the entire search process. The proposed dynamic spiral search method is defined as follows:

$$q = e^{\cos(\frac{\pi t}{T_{\max}})} \tag{24}$$

$$\begin{aligned} B_i(t+1) = X^* + e^{ql} \times \cos(2\pi l) \times b_1 \times (B_i(t) - Lb^*) + e^{ql} \\ \times \cos(2\pi l) \times b_2 \times (B_i(t) - Ub^*) \end{aligned} \tag{25}$$

where q is the spiral factor used to adjust the intensity of spiral search, l is a uniformly distributed random number with $l \in (-1, 1)$. The conceptual model of the dynamic spiral search method is shown in Fig. 8. The curves of the sine learning factor r and the spiral factor q are shown in Fig. 9.

In order to adapt to complex search spaces or multimodal problems, adaptive t-distribution disturbances with heavier tails than normal distributions are introduced. It is a probability distribution that achieves a smooth transition from Gaussian distribution to Cauchy distribution by adjusting its degree of freedom parameters. This adaptive approach enables the algorithm to dynamically adjust its search behavior, providing a more effective exploration of the solution space. The foraging behavior based on adaptive t-distributed disturbance is defined as follows:

$$\begin{cases} x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b)\ if\ rand < 0.5 \\ \quad +C_2 \times (x_i(t) - Ub^b) \\ x_i(t+1) = X^* + X^* \times trnd(e^{\left(4\left(\frac{t}{T_{\max}}\right)\right)^2})\ else \end{cases} \tag{26}$$

where $trnd$ is t-distribution function, the degrees of freedom in the distribution function represent disturbance probability, which changes dynamically with the number of iterations.

As the iteration proceeds, the concentration of the probability density reduces the variability of the perturbation values, and the perturbations are mainly generated around the current best solution. The probability density distribution heatmap and corresponding distribution function images of adaptive t-distribution disturbance at different iterations are shown in Fig. 10.

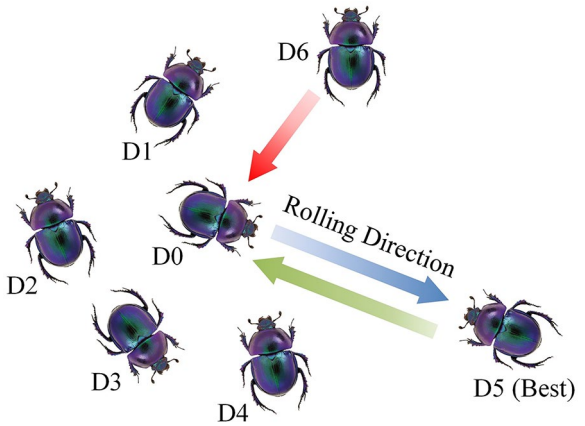

Fig. 7. Conceptual model of merit-oriented mechanism.

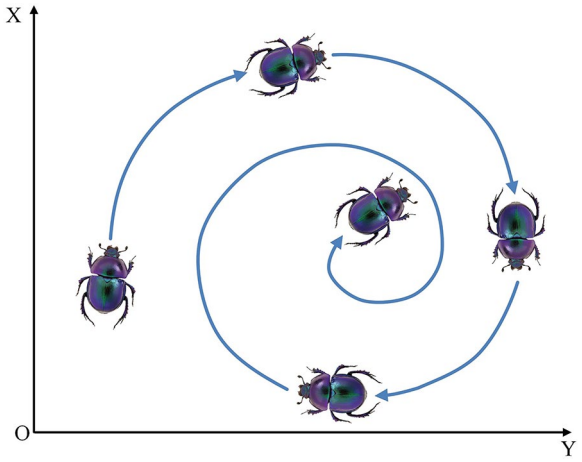

Fig. 8. Conceptual model of dynamic spiral search.

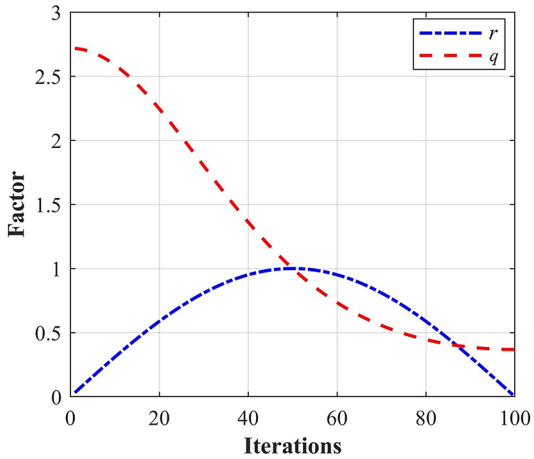

Fig. 9. Sine learning factor and spiral factor.

EDBO-BP hybrid optimization

Building upon the improvements introduced by the enhanced dung beetle optimizer, the integration with a BPNN further refines the optimization process. This hybrid approach leverages the strengths of both the EDBO and BP to deliver superior adaptability and precision in PID control systems.

To balance the speed and stability of the BPNN learning process, this paper employs the EDBO to globally optimize the learning rate and inertia coefficient. Hybrid optimization combining EDBO and BPNN is designed based on the BPNN PID control. The PID control structure optimized by the hybrid algorithm is shown in Fig. 11.

Application of EDBO-BP to tune the PID controller

BP neural networks (BPNN) have certain advantages in tuning PID controllers. Through the forward propagation of its input, hidden, and output layers, the values of K_p, K_i, and K_d can be obtained. After applying the gradient descent algorithm, the weight coefficients of the neurons in each layer are adjusted through backpropagation, and then forward propagation is used again to obtain a new set of values. However, setting the learning rate and inertia coefficient is challenging, as these parameters significantly affect optimization results. In simulation experiments, it was found that these two parameters had a considerable impact on the results, which is why the Enhanced Dung Beetle Optimizer (EDBO) was introduced to adjust these two parameters of the BP neural network, forming the EDBO-BP-PID method. This method allows the algorithm to quickly find an optimal set of K_p, K_i, and K_d, which are then fed into the simulation model for ITAE (Integral of Time-weighted Absolute Error) analysis. Based on changes in the fitness value, the superiority of the algorithm can be demonstrated.

The proposed algorithm formulates a mathematical model for the PID control system and defines its optimization objective. The optimization problem is centered around enhancing the PID control algorithm through the integration of EDBO and BPNN. The topology of the BPNN is determined. The EDBO algorithm initializes the network parameters. Then, parameter combinations are iteratively optimized. The EDBO algorithm

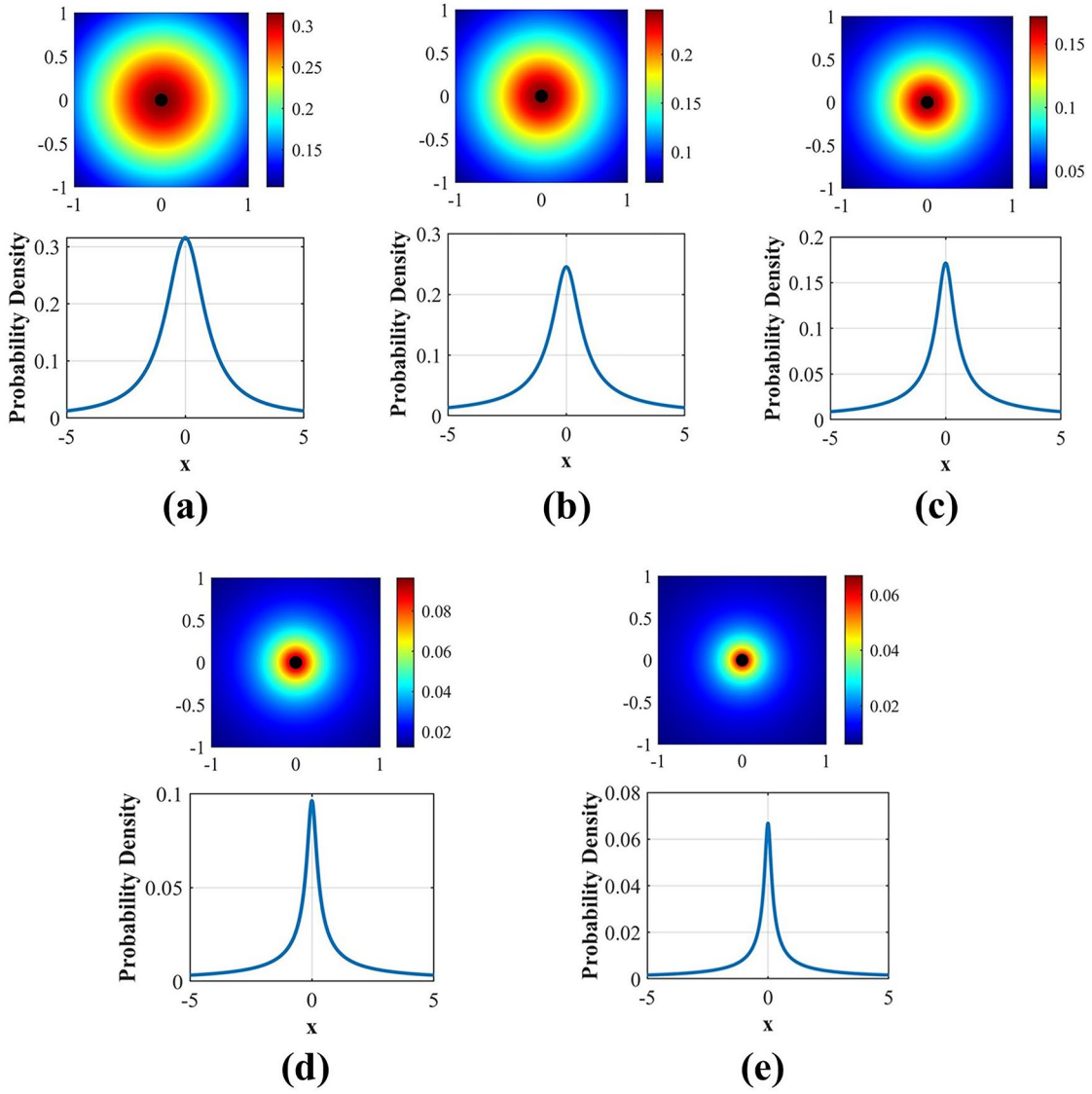

Fig. 10. Probability density of adaptive t-distribution disturbance. (**a**) $t=10$. (**b**) $t=50$. (**c**) $t=70$. (**d**) $t=90$. (**e**) $t=100$.

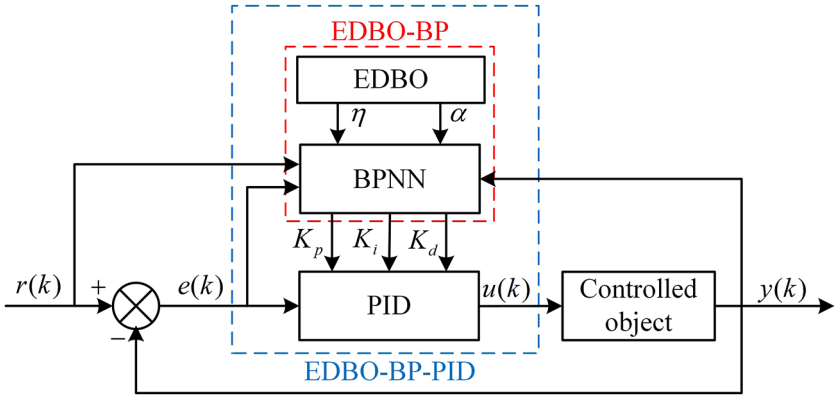

Fig. 11. The PID control structure based on EDBO-BP hybrid algorithm.

adapts to real-time network performance. Finally, the BPNN and EDBO algorithms are applied alternately to globally optimize the PID parameter set. This hybrid optimization combines model-based optimization and data-driven learning. After defining the structure and optimization process of the BPNN, the fitness function of the EDBO is linked to the ITAE (Integral of Time-weighted Absolute Error) as the objective function of the algorithm. During the optimization process, the position of the dung beetle corresponds to the PID parameters and network parameters to be optimized. The flowchart of the PID control algorithm with EDBO-BP is shown in Fig. 12.

In order to obtain satisfactory dynamic characteristics of the transition process, the system performance evaluation index ITAE is introduced as the objective function of the algorithm. ITAE is a widely accepted performance index for control systems. The discrete equation of the ITAE is expressed as follows:

$$ITAE = \sum_{k=0}^{n} |e(kT)| \cdot kT \cdot T \tag{27}$$

where T is the sampling period and K is the sampling time.

Experiments and result analysis

EDBO performance test

Comparison of convergence curves

To show the convergence effect of each algorithm in test functions, the corresponding algorithm convergence curves were drawn according to the generated data. Figure 13 shows the fitness curves of EDBO and other algorithms in the optimization process of the partial benchmark functions. The comparison algorithm includes dung beetle optimizer (DBO), dwarf mongoose optimization algorithm (DMOA), marine predator algorithm

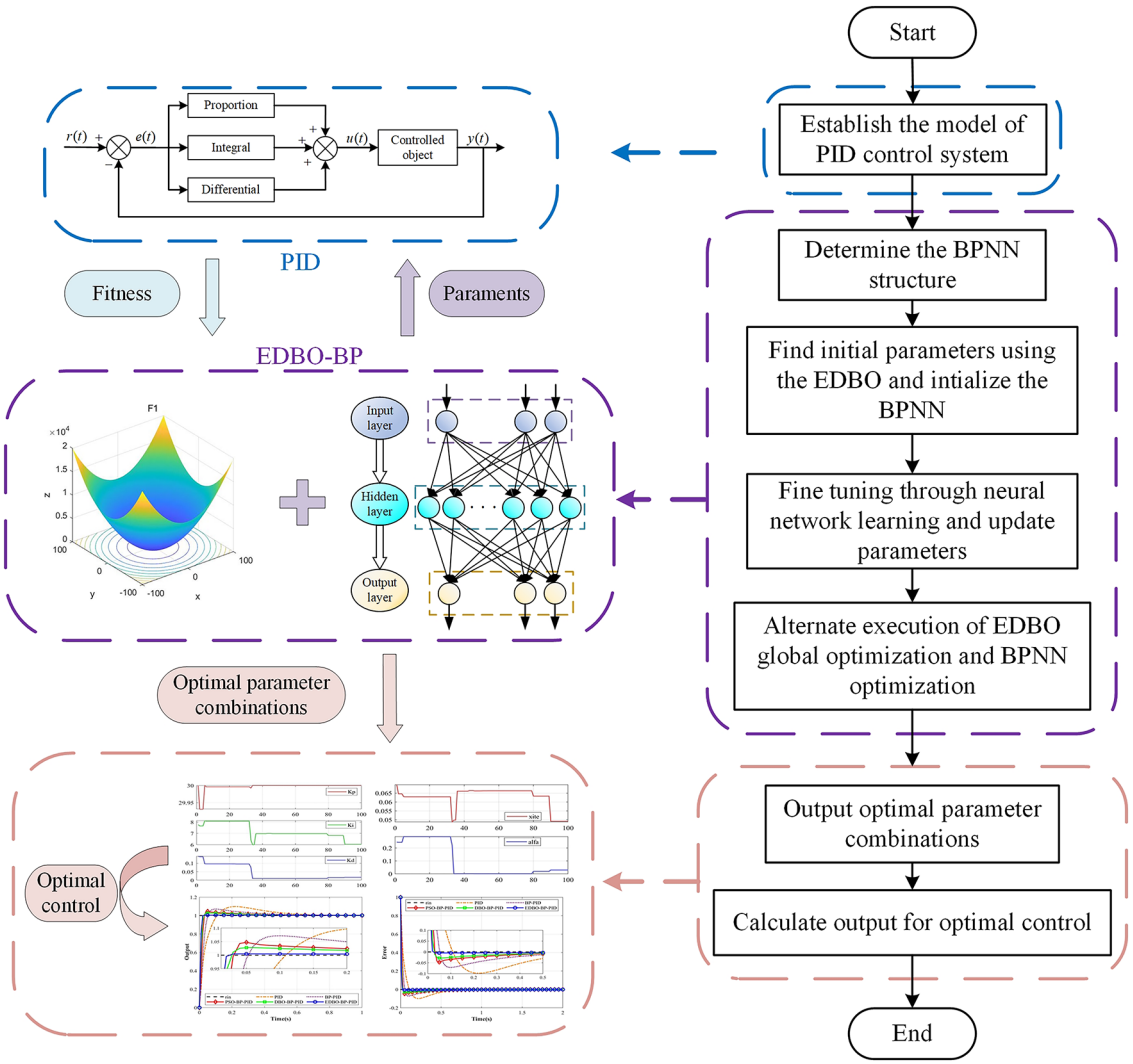

Fig. 12. Flowchart of PID control algorithm with hybrid optimization of EDBO-BP.

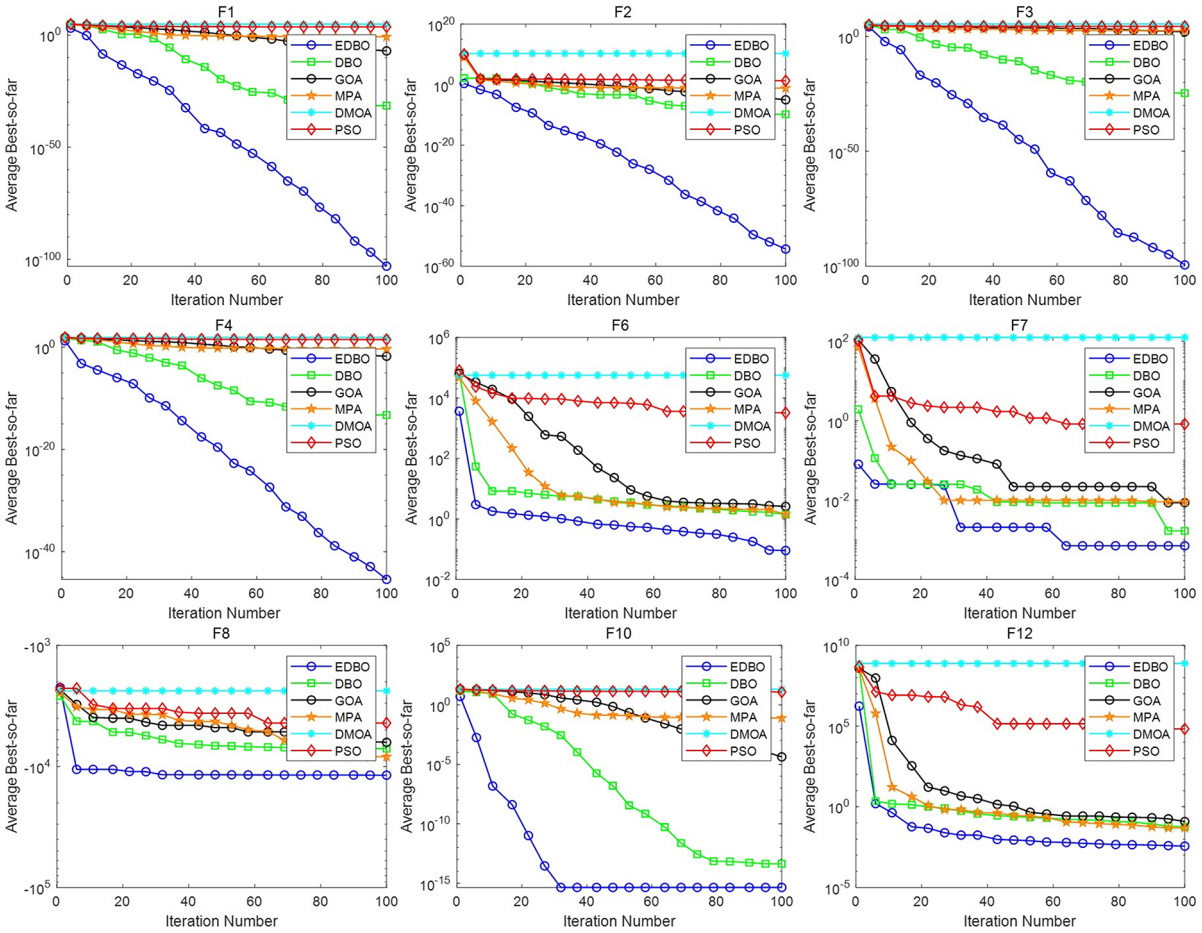

Fig. 13. Convergence curves on test function.

(MPA), gazelle optimization algorithm (GOA), and particle swarm optimization (PSO). It can be seen that the EDBO was superior to other algorithms in terms of optimization speed and convergence accuracy.

Wilcoxon sign rank test

To further compare the differences between the EDBO and other optimization algorithms, a statistical test called the Wilcoxon signed-rank test was conducted. The significance of the statistical results was determined by calculating the p-value. If the p-value was <0.05, it was concluded that there was a significant difference between the two algorithms. The results of the calculation are shown in Table 1. Table 1 presents the performance differences between EDBO and other optimization algorithms, including dung beetle optimizer (DBO), dwarf mongoose optimization algorithm (DMOA), marine predator algorithm (MPA), gazelle optimization algorithm (GOA), and particle swarm optimization (PSO), across 23 benchmark functions. DBO, DMOA, MPA, and GOA are optimization algorithms recently proposed for addressing optimization problems, while PSO is a widely used classic optimization algorithm.

The results indicate that the proposed EDBO algorithm has a lower similarity in search outcomes compared to its competitors. Therefore, the optimization performance of the proposed EDBO across the 23 benchmark functions shows significant differences from other metaheuristic algorithms.

DC motor control system

The motor control system represents a quintessential and widely adopted paradigm within the realm of industrial control systems. To rigorously discuss and validate the efficacy of the control algorithm proposed in this paper, a DC motor control system has been selected as the controlled object for the simulation experiment. This choice is predicated on the DC motor's prevalent application in various industrial scenarios due to its simplicity and robustness. The simplified DC motor model utilized is shown in Fig. 14.

The DC motor transfer function with armature voltage as input and speed as output can be expressed as:

$$G(s) = \frac{\omega(s)}{u(s)} = \frac{\frac{1}{k_1}}{\frac{LJ}{k_1 k_2}s^2 + \frac{RJ}{k_1 k_2}s + 1} \tag{28}$$

Function	DBO	DMOA	MPA	GOA	PSO
F1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F2	2.63E-11	2.63E-11	2.63E-11	2.63E-11	2.63E-11
F3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F4	2.95E-11	2.95E-11	2.95E-11	2.95E-11	2.95E-11
F5	4.08E-11	3.02E-11	0.019112	0.020681	3.02E-11
F6	3.02E-11	3.02E-11	0.005084	3.02E-11	3.02E-11
F7	2.57E-07	3.02E-11	1.69E-09	6.70E-11	3.02E-11
F8	3.2E-09	3.02E-11	1.96E-10	4.50E-11	3.02E-11
F9	0.160802	1.21E-12	NaN	0.021577	1.21E-12
F10	NaN	1.21E-12	7.42E-13	2.07E-13	1.21E-12
F11	NaN	1.21E-12	NaN	NaN	1.21E-12
F12	5.97E-09	3.02E-11	0.958731	5.57E-10	3.02E-11
F13	1.01E-08	3.02E-11	0.876635	0.000655	3.02E-11
F14	0.430495	3.38E-11	0.009343	0.237113	0.362794
F15	6.52E-09	3.02E-11	3.02E-11	3.82E-10	5.57E-10
F16	0.190957	1.01E-11	3.71E-05	0.000100	3.68E-11
F17	NaN	1.21E-12	0.081493	1.30E-07	5.71E-09
F18	0.201687	7.39E-11	0.863672	0.070997	5.1E-10
F19	0.065941	1.01E-11	0.569127	0.118950	1.01E-11
F20	0.000272	4.10E-12	4.10E-12	4.10E-12	4.10E-12
F21	5.09E-09	9.27E-12	1.06E-08	3.48E-09	1.34E-10
F22	0.069909	7.87E-12	1.69E-06	5.31E-07	8.54E-11
F23	0.000472	1.96E-11	2.64E-08	6.12E-09	5.09E-10

Table 1. Wilcoxon sign rank test.

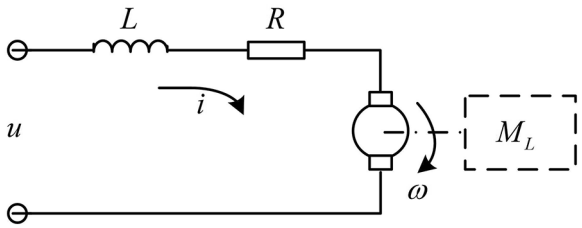

Fig. 14. Simplified DC motor model.

The specific parameters of the DC motor mathematical model are shown in Table 2. Finally, the DC motor transfer function can be calculated as:

$$G(s) = \frac{20}{0.04s^2 + 6s + 1} \tag{29}$$

Results and analysis

The BP neural network takes $r(k)$, $e(k)$, and $y(k)$ as inputs and predicts successive values of PID parameters. In the training process of the BPNN, data collection is first performed, including $r(k)$, $e(k)$, and $y(k)$, and the optimal K_p, K_i, and K_d values under various conditions. Then the BPNN is initialized with random weights, followed by backpropagation to compute the output of the network for each training sample. After that, error computation is carried out to measure the errors between the predicted and the actual optimal parameters of the PID parameters. Finally, the weights are adjusted using the gradient descent method to minimize the error. In the EDBO optimization process, a random location representing the candidate solution is initialized. The optimization is performed by multiple search strategies. BPNN and EDBO are used alternately to iteratively find the optimal PID parameters under different conditions.

The simulation scheme consists of implementing the BPNN and EDBO algorithms to optimize the PID parameters of the DC motor control system and simulating and testing the performance under various conditions including disturbances, step changes, and noise variations. The PID control system is modeled by Simulink. The BPNN structure for predicting the PID parameters is defined through .m files and the EDBO algorithm is implemented through .m files to optimize the BPNN. These codes work together to iteratively improve the PID parameters and the performance of the control system.

Parameter name	Symbol	Unit	Parameter value
Rotor moment of inertia	J	kg/s^2	0.01
Current-torque constant	k_1	$N.m/A$	0.05
Voltage-speed constant	k_2	$Rad/s/v$	0.05
Armature resistance	R	Ω	1.5
Armature inductance	L	H	0.01

Table 2. Mathematical model of DC motor.

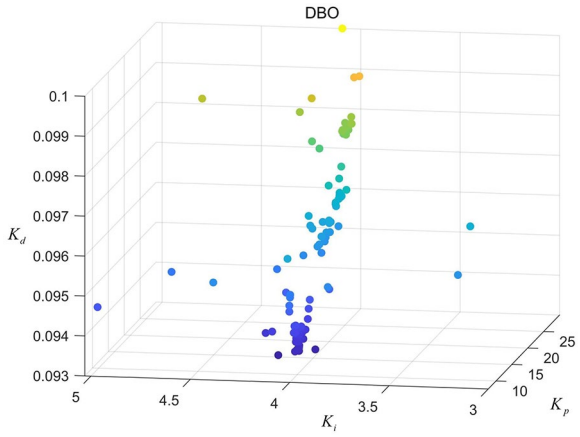

Fig. 15. Particle optimization trajectories for DBO.

For the control group, the particle swarm optimization (PSO) and dung beetle optimizer (DBO) were selected as comparison targets. PSO is easy to implement and performs well in the optimal design of neural networks and PID control. Additionally, DBO serves as the foundation for the EDBO proposed in this paper. By comparing the performance of the control system optimized by the traditional DBO, the superiority of the proposed method can be verified.

Particle optimization trajectories for DBO and EDBO

This section aims to illustrate the improvements made by the EDBO in tuning PID parameters by comparing the optimization trajectories. By plotting the three-dimensional optimization trajectories of particles for the PID controller parameters K_p, K_i, and K_d, it visually explains why the original DBO tends to get trapped in local optima and how EDBO addresses this limitation. Representative runs of both the DBO and EDBO optimization algorithms were selected, each running for 100 iterations, with particle trajectories recorded in the K_p, K_i, and K_d parameter space during each iteration. Three-dimensional scatter plots were then generated to showcase the differences in exploration and exploitation behaviors between the two algorithms.

From the three-dimensional trajectory plot of DBO, it is clear that the optimization particles quickly converge around $K_i = 4$, indicating that DBO tends to get trapped in local optima. This premature convergence reduces the search range and limits the algorithm's ability to find better solutions in the global search space. This behavior demonstrates the imbalance between the exploration and exploitation phases in the original DBO. In contrast, the optimization trajectory of EDBO shows a much broader search range, covering more areas in the parameter space. The particles do not converge prematurely at local optima, and the global exploration capability is significantly enhanced. The wider search range indicates that EDBO can overcome the local optima problem and significantly improve global optimization performance. The particle optimization trajectory of DBO is shown in Fig. 15, while that of EDBO is shown in Fig. 16.

Fitness convergence results

The ITAE fitness curve represents the cumulative weighted value of the error between the system output and the expected reference signal over time. By calculating the integral of the error and weighting it according to time, the overall performance of the control system can be comprehensively evaluated. The fitness value convergence curves are shown in Fig. 17.

DBO has the fastest initial convergence speed and quickly achieves a decrease in the value of the fitness function in the first five rounds of the search. In addition, as the number of iterations increases, EDBO can continue to identify new solutions that are better than the previous stage solution, which is manifested in the fitness value convergence curve sloping toward the lower right corner. The statistical results of the optimal, worst, mean, and standard deviation values of the fitness for each algorithm are given in Table 3.

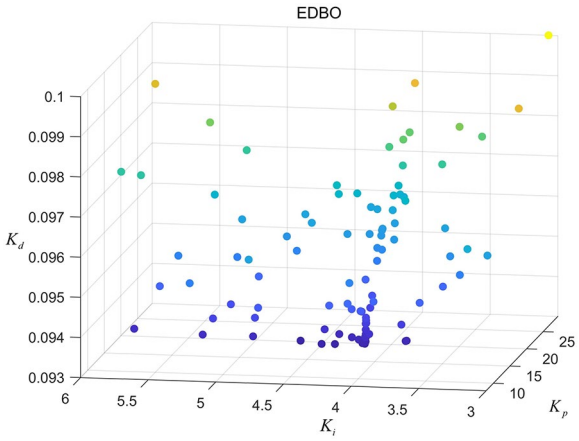

Fig. 16. Particle optimization trajectories for EDBO.

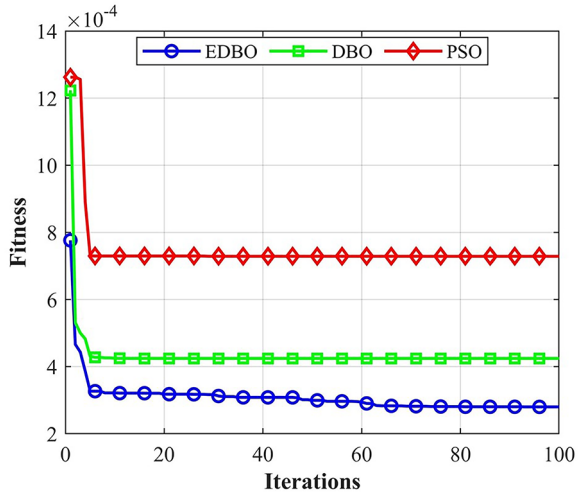

Fig. 17. Convergence curve of fitness value.

Algorithm	Best	Worst	Mean	Standard deviation
EDBO	0.000280	0.000776	0.000308	0.000055
DBO	0.000424	0.001223	0.000435	0.000081
PSO	0.000729	0.001262	0.000746	0.000092

Table 3. Comparison results of fitness.

EDBO achieved the smallest fitness value of 0.00028, followed by DBO with an optimal fitness of 0.00424. PSO obtained the highest fitness value. In terms of the worst fitness value, the worst fitness of the EDBO is 0.000776, which is significantly smaller than DBO and PSO. The above results show that the EDBO can achieve better control performance and stability than DBO and PSO in terms of global optimal solution identification ability, convergence speed, and performance robustness.

Parameter optimization results

Table 4 presents the tuning results of the three output layer parameters and the hyperparameters across 20 rounds of repeated experiments. By examining the best, worst, mean, and standard deviation values, the differences and performances of different algorithms in parameter optimization are highlighted.

Under the EDBO-BP-PID algorithm, the standard deviation of each parameter is relatively smaller, indicating that this algorithm can achieve more precise parameter tuning during the optimization process. In contrast, the DBO-BP-PID and PSO-BP-PID algorithms exhibit larger standard deviations, suggesting greater variability in their optimization processes. The reduced standard deviation in EDBO-BP-PID demonstrates its capability to

Algorithm	Parameters	Best	Worst	Mean	Standard deviation
EDBO-BP-PID	K_p	30.0000	29.9309	29.9967	0.0117
	K_i	6.0237	8.1161	7.1638	0.6725
	K_d	0.0171	0.1415	0.0421	0.0427
	η	0.0491	0.0697	0.0626	0.0056
	α	0.0296	0.2883	0.0983	0.1292
DBO-BP-PID	K_p	25.0000	20.9121	24.9591	0.4087
	K_i	4.2536	3.3388	4.2387	0.1133
	K_d	0.0100	0.1986	0.0220	0.0364
	η	0.0948	0.0394	0.0936	0.0083
	α	0.0025	0.0217	0.0046	0.0056
PSO-BP-PID	K_p	25.0000	23.3558	24.9815	0.1654
	K_i	4.1027	25.0000	24.9815	0.1654
	K_d	0.0100	25.0000	24.9815	0.1654
	η	0.0014	0.0826	0.0045	0.0119
	α	0.0787	0.1661	0.0773	0.0136
EDBO-PID	K_p	30.0000	26.5768	28.1226	0.4887
	K_i	5.0018	5.2710	5.0037	0.0276
	K_d	0.1335	0.2000	0.1373	0.0102
DBO-PID	K_p	30.0000	24.6680	29.9466	0.5331
	K_i	5.1040	3.8568	5.0864	0.1301
	K_d	0.2000	0.1647	0.1996	0.0035
PSO- PID	K_p	25.0000	22.5365	24.7527	0.7787
	K_i	4.1674	3.7249	4.1275	0.1421
	K_d	0.0847	0.1048	0.0848	0.0076
BP- PID	K_p	25.0000	19.5365	20.7527	0.8425
	K_i	4.1971	3.2258	4.1985	0.2674
	K_d	0.1267	0.1022	0.0957	0.0121

Table 4. Output results of neural networks optimized by different algorithms.

consistently find parameter combinations close to the global optimum across multiple experiments, further proving its superior tuning performance.

Figures 18 and 19 respectively illustrate the tuning processes of PID parameters and neural network parameters over 100 iterations for the three algorithms. As shown in the figures, the EDBO-BP-PID algorithm quickly converges and maintains stability in the early stages of iteration. In contrast, while the DBO-BP-PID and PSO-BP-PID algorithms also converge rapidly initially, they exhibit significant fluctuations in parameter adjustments as iterations increase, making further optimization difficult.

In real industrial production processes, the stability, accuracy, and responsiveness of control systems are crucial. Due to its precise parameter adjustments during tuning, the EDBO-BP-PID algorithm better adapts to the demands of actual working environments and enhances the overall performance of the system. Specifically, by optimizing the hyperparameters of the neural network, the EDBO-BP-PID algorithm not only ensures control accuracy but also improves the system's response speed and robustness, maintaining high control performance even in the face of disturbances and parameter variations.

EDBO-BP-PID algorithm achieves optimal tuning of the control system by seeking a balance among multiple performance indices. The principles of the ESO method are reflected in the EDBO-BP-PID's optimization strategy, where a systematic adjustment mechanism ensures that the control system consistently delivers superior performance in complex and variable industrial environments.

Sensitivity analysis

Sensitivity analysis is the study of uncertainty in the output of a mathematical model and how it is divided and assigned to different sources of uncertainty in the input. During the simulation experiment data is recorded and sensitivity analysis is performed, after traversing the two parameters such as learning rate η and inertia coefficient α in the BP neural network, both of which are taken from 0 to 0.2, the relationship between the two parameters and the adaptation value is obtained and the sensitivity analysis graph of the parameters is shown in Fig. 20.

Therefore, the two parameters have a greater influence on the whole control system, and if they are set to a fixed value, it reduces the algorithm's optimization-seeking effect, so we chose to adjust the parameters of the BP neural network with EDBO.

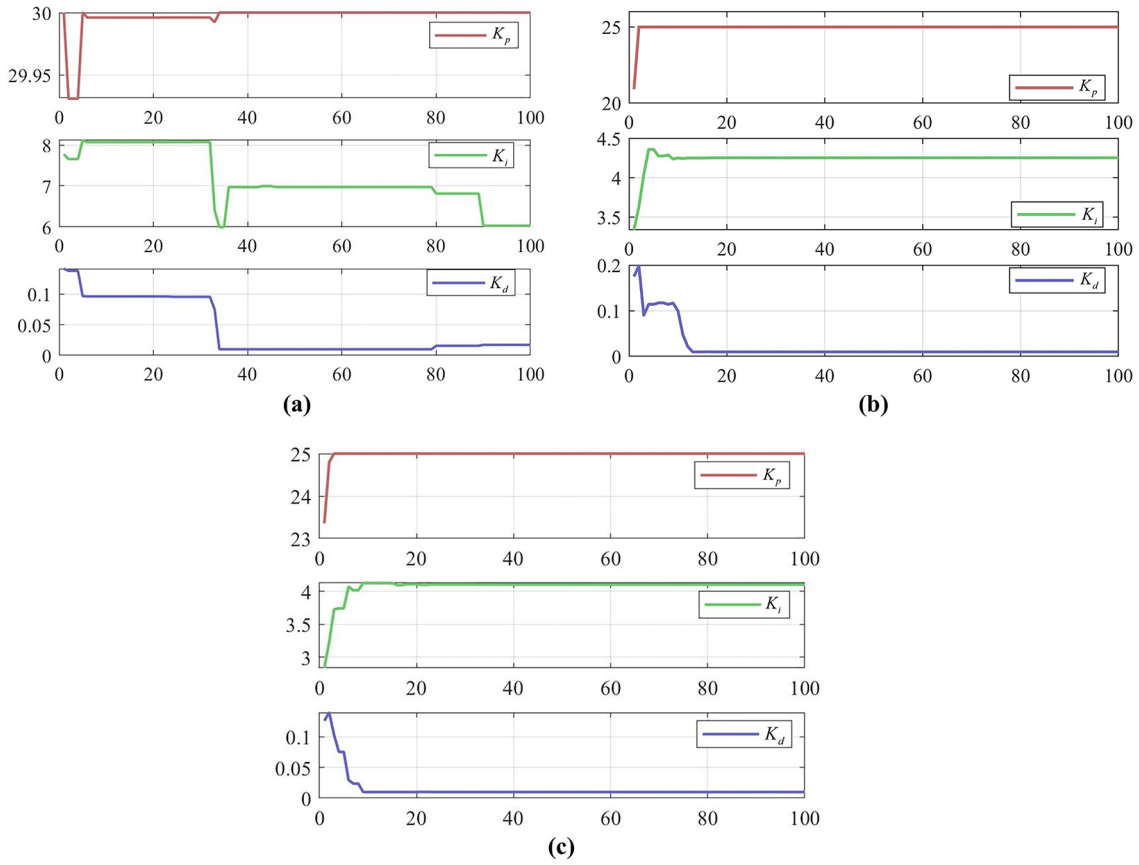

Fig. 18. Comparison of PID parameters tuning. (**a**) EDBO-BP-PID. (**b**) DBO-BP-PID. (**c**) PSO-BP-PID.

Control system performance results

In order to more fully verify the performance of the control algorithm in this paper, a series of performance evaluation experiments are further set up. Under the settings of the nondisturbance experiment, disturbance experiment, step change experiment, and noise variation experiment, the performance of the control algorithm such as step response state and error distribution are observed.

1) In this experiment, the control system is only affected by the predetermined set value, excluding all external disturbances, to observe the performance of the control strategy under ideal conditions. The step output response curve of the control system under the nondisturbance experimental setting is shown in Fig. 21. The error curve of the control system is shown in Fig. 22.

Table 5 shows the performance metrics of various control algorithms applied to the control system. These performance indicators are typical indicators used to evaluate the effectiveness of control systems.

In the EDBO-BP-PID control algorithms, the system can obtain the lowest overshoot value (0.5), the lowest rising time value (0.012), the lowest settling time value (0.02), the lowest peak value (1.005) and the lowest state error value (0.001), and in these five evaluation dimensions, the results of the remaining control groups are significantly different from the optimal results. Its peak time is slightly longer than DBO-BP-PID and PSO-BP-PID, indicating that although it may take longer to reach peak response, it can self-correct faster and have smaller overall errors.

Figures 21 and 22, and Table 5 show that the system under the EDBO-BP-PID control algorithm has better response speed, stability, and accuracy. It can achieve control requirements more effectively and improve control quality.

2) The amplitude of this disturbance is set to 20% of the system's stable output value to represent a moderately strong external disturbance. The selection of this disturbance is based on some real industrial processes, such as chemical reactors and power systems, where disturbances of similar proportions may be experienced. Figures 23 and 24 show the results of the output response curve and error curve of different control algorithms in the control system when a disturbance is added.

Traditional PID control shows a large deviation. This is because traditional PID control is designed based on the mathematical model of the system. When external disturbances interfere with the system, traditional PID control cannot adjust the control quantity in a timely and accurate manner, causing the system to deviate from the desired trajectory. In contrast, the control algorithm proposed in this paper shows a more stable and accurate control effect.

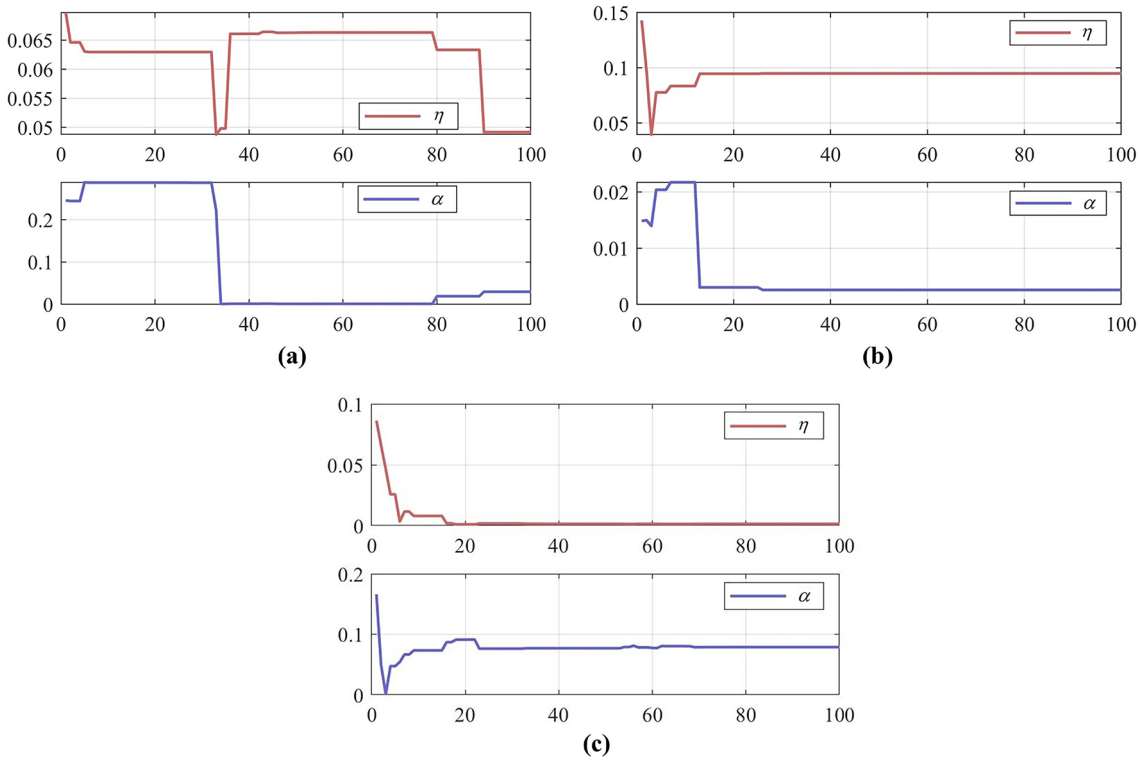

Fig. 19. Comparison of neural network parameters tuning. (**a**) EDBO-BP-PID. (**b**) DBO-BP-PID. (**c**) PSO-BP-PID.

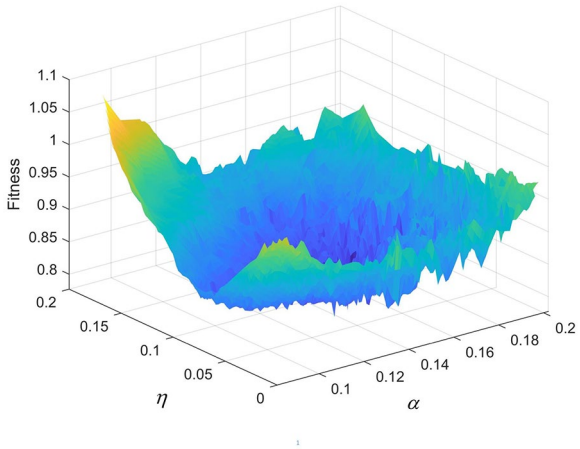

Fig. 20. Sensitivity analysis graph of the parameters.

The EDBO-BP-PID algorithm demonstrates optimal steady-state response and rapid error correction. After the disturbance, the system output shows minimal deviation from the setpoint and quickly returns to the desired output, highlighting its excellent disturbance rejection and system stability. In contrast, traditional PID and BP-PID algorithms exhibit larger overshoot and longer recovery times, with noticeable oscillations, especially in the PID controller, indicating its vulnerability to external disturbances.

While the PSO-BP-PID and DBO-BP-PID algorithms improve stability and recovery speed to some extent, they still fall short of the EDBO-BP-PID's performance. These algorithms show some fluctuations in output response and less smooth error correction, indicating limited effectiveness in handling external disturbances.

3) In order to further evaluate the tracking ability of the control system to external changes and the robustness of the control algorithm, a step change experiment was set up on the system's step response. Set the control system to simulate a step change at the 1st and 2nd seconds respectively. The input of the given system is 1. The

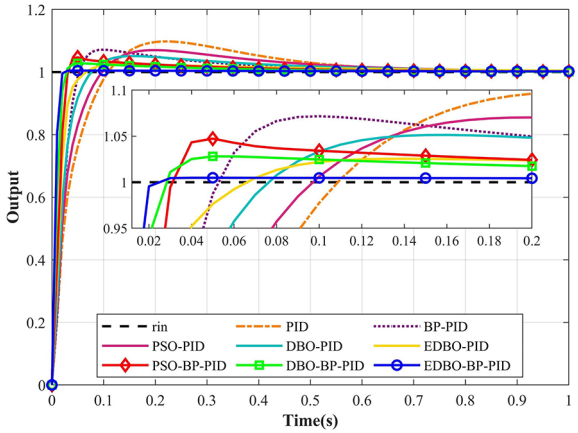

Fig. 21. System output curve.

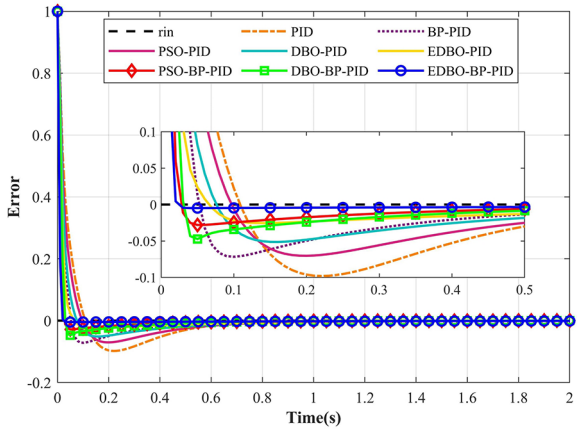

Fig. 22. System error curve.

Control algorithm	Overshoot (%)	Rising time (s)	Settling time (s)	Peak time (s)	Peak	Steady-state error
EDBO-BP-PID	0.5	0.012	0.02	0.032	1.005	0.0010
DBO-BP-PID	2.8	0.016	0.15	0.050	1.280	0.0018
PSO-BP-PID	5.0	0.024	0.25	0.045	1.050	0.0017
BP-PID	7.1	0.040	0.40	0.100	1.071	0.0018
PID	11.0	0.076	0.56	0.220	1.110	0.0025

Table 5. Comparison of control algorithm performance metrics.

dynamic behavior of the system output under different PID control algorithms when the system input undergoes a predetermined step change is shown in Fig. 25.

As shown in Fig. 25, there are significant differences in the dynamic behavior of the system output under various PID control algorithms when the system input undergoes step changes. The EDBO-BP-PID algorithm demonstrates the fastest response and the smallest overshoot at each step change point. The system output almost immediately follows the setpoint adjustment and stabilizes within a short time, showcasing its excellent tracking ability and strong robustness in response to external changes.

In contrast, traditional PID and BP-PID algorithms exhibit noticeable overshoots and oscillations when the step changes occur, particularly at the 1-second and 2-second step change points. The output signal shows a significant overshoot and takes longer to return to a stable state, indicating slower response speed and poorer stability when dealing with rapidly changing external inputs. While the PSO-BP-PID and DBO-BP-PID algorithms somewhat improve the overshoot issue, their response speed and final stability still do not match that of the EDBO-BP-PID algorithm.

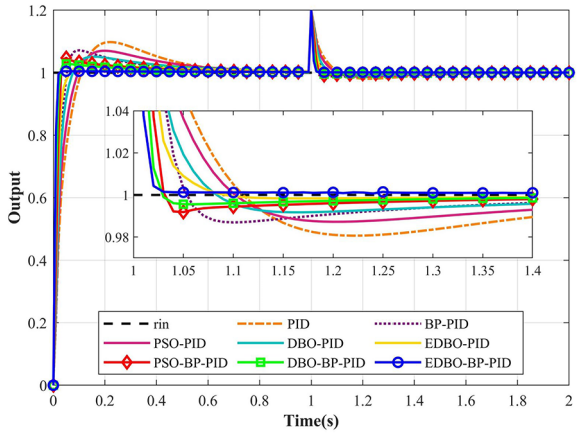

Fig. 23. System output curve at 20% disturbance.

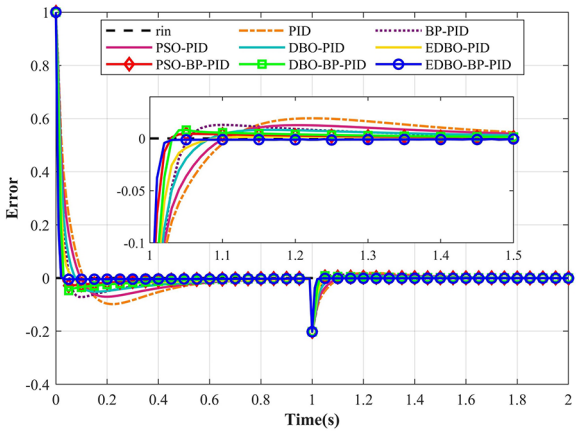

Fig. 24. System error curve at 20% disturbance.

4) In real industrial environments, control systems often need to operate in changing and unpredictable environments. Therefore, noise variation experiments are crucial to verify the effectiveness and robustness of control strategies. Introduce random noise signals into the control system to simulate the random interference that the control system may encounter in actual working conditions, and further evaluate the performance of the PID controller under nonideal and nondeterministic conditions. The noise signal is set to white noise with known statistical characteristics. The noise amplitude varies within the range of [0.95, 1.05]. The sampling time of the noise fluctuation is 0.001 s, and the simulation time of the entire system is set to 1 s. The system input after adding the external noise signal is shown in Fig. 26.

In Fig. 27, the system output performance under the influence of noise is shown for different PID control algorithms. The EDBO-BP-PID control algorithm demonstrates the smallest disturbance error when dealing with external noise, indicating its significant advantages in control accuracy and stability. Despite the noise interference, the system output quickly approaches the reference input value and maintains high stability. In contrast, other control algorithms, such as PID, BP-PID, PSO-BP-PID, and DBO-BP-PID, exhibit greater fluctuations under noise, with the traditional PID algorithm showing the most noticeable deviation.

Study case of DC motor with parameter uncertainties

Based on the previous model, the transfer function is replaced with a brushless DC motor simulation model. This model is characterized by nonlinearity and a complex control system, which is convenient to test the robustness of the proposed algorithm in this paper. Uncertainties such as changes in the desired speed of the motor and changes in the load of the motor are also introduced in the later experiments to analyze the following of the motor speed in this case. The brushless DC motor in this model is driven by an H-bridge, and four power tubes and an inductor are used to form an inverter circuit, with each phase of the windings being independent of each other, which can flexibly change the current size and direction of the windings.

After building the simulation model of a brushless DC motor, the effects of three algorithms such as BP-PID, DBO-BP-PID, and EDBO-BP-PID are compared. No-load acceleration and deceleration experiments, loaded acceleration and deceleration experiments, and uniform speed loading and unloading experiments of the motor

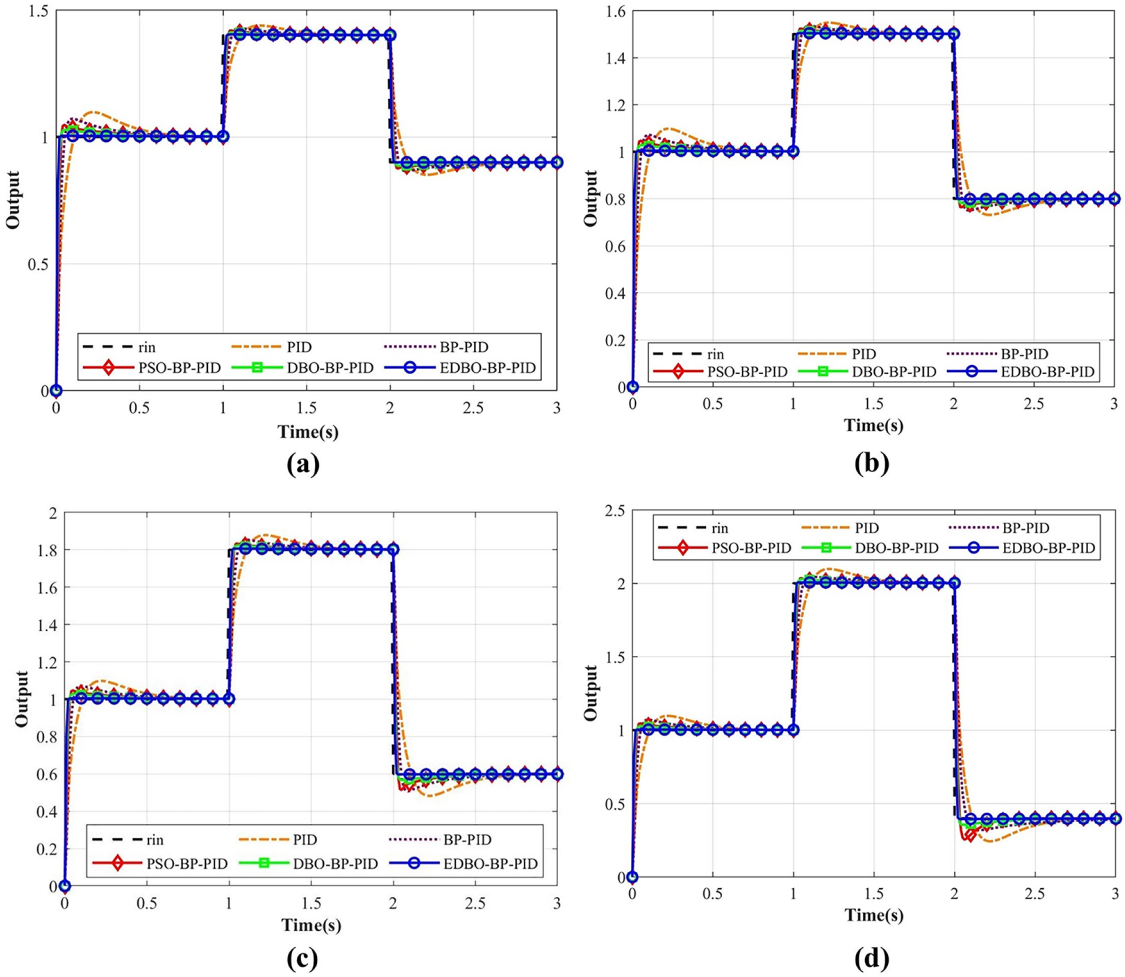

Fig. 25. System output curve under step change.

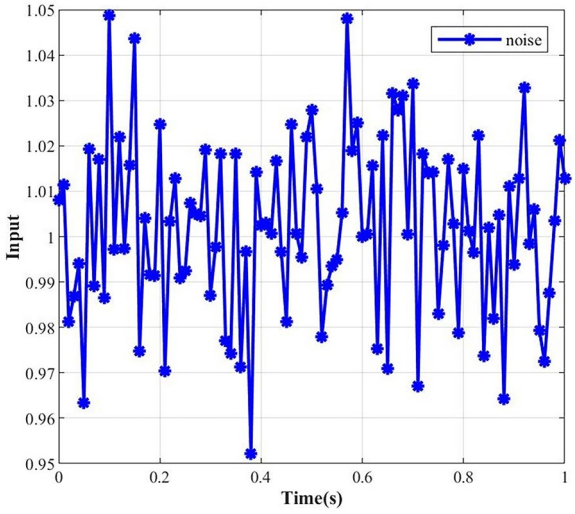

Fig. 26. System input curve under external noise.

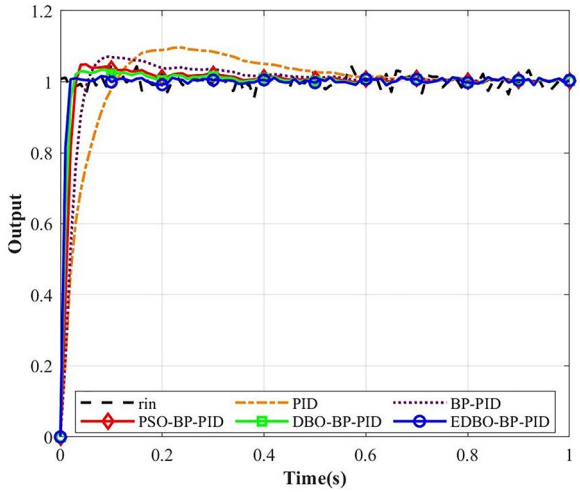

Fig. 27. System output curve under external noise.

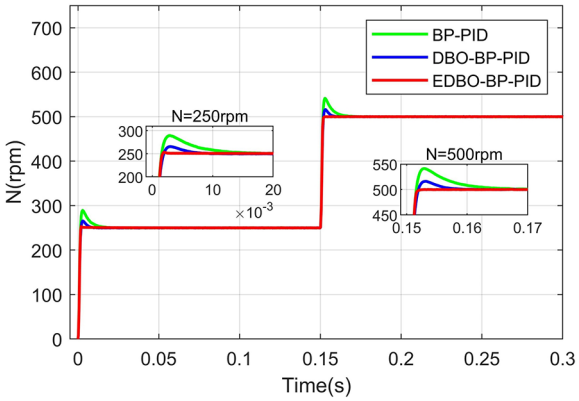

Fig. 28. Speed response curve of motor no-load acceleration experiment.

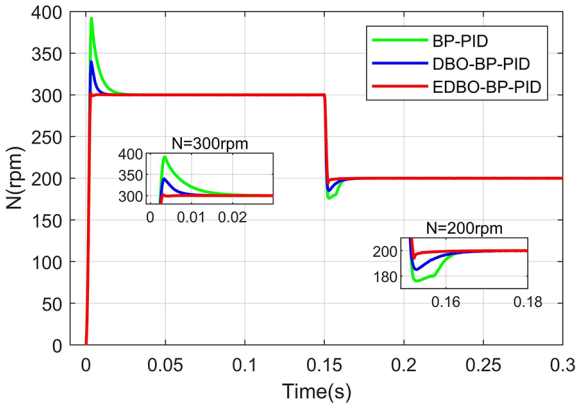

Fig. 29. Speed response curve of motor no-load deceleration experiment.

were carried out to compare the motor speed following the three algorithms. The motor speed response curves of the controllers are shown in Figs. 28, 29, 30, 31 and 32. It can be concluded from the simulation experiments of the brushless DC motor that the motor speed response under the EDBO-BP-PID algorithm is faster and the overshoot is minimized, and the robustness is better than the other two algorithms.

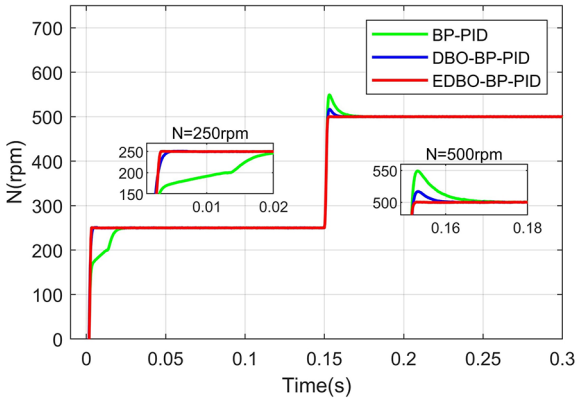

Fig. 30. Speed response curve of motor loaded acceleration experiment.

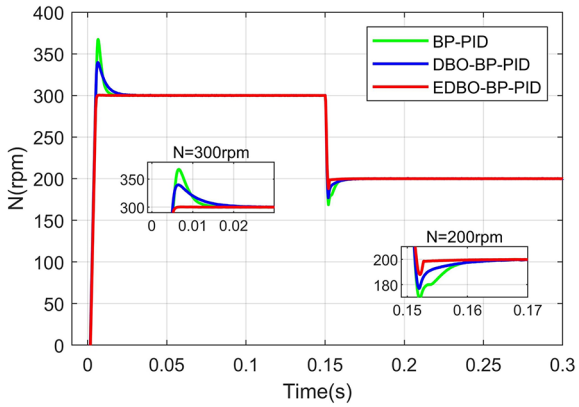

Fig. 31. Speed response curve of motor loaded deceleration experiment.

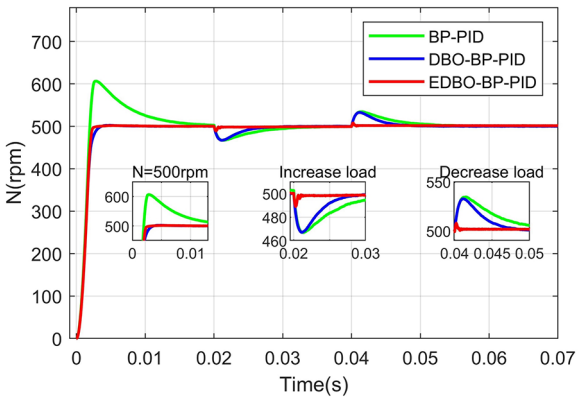

Fig. 32. Speed response curve of motor uniform speed loading and unloading experiments.

A concise quantitative and qualitative comparative analysis of the experimental results is shown in Table 6. The quantitative analysis of five simulation experiments on DC motors concludes that the EDBO-BP-PID algorithm provides the best optimization of the PID controller parameters, with the lowest overshoot and the shortest response time of the control system.

Conclusions

This paper presents a PID control algorithm based on the hybrid optimization of the EDBO and BPNN. Unlike existing PID control systems, the proposed algorithm uses BPNN to identify parameter combinations without prior information on the PID control parameters. To alleviate overfitting and improve the generalization ability

Control algorithm	Type of experiment	Overshoot (%)	Rising time (s)
EDBO-BP-PID	No-load acceleration	0.7	0.0010
	No-load deceleration	1.6	0.0028
	Acceleration under load	1.9	0.0037
	Deceleration under load	0.6	0.0019
	Loading and unloading at constant speed	2.0	0.0016
DBO-BP-PID	No-load acceleration	5.4	0.0050
	No-load deceleration	16	0.0100
	Acceleration under load	4.0	0.0100
	Deceleration under load	13.3	0.0200
	Loading and unloading at constant speed	60.	0.0070
PSO-BP-PID	No-load acceleration	16	0.0100
	No-load deceleration	33	0.0200
	Acceleration under load	10	0.0200
	Deceleration under load	20	0.0150
	Loading and unloading at constant speed	6.0	0.0100

Table 6. Quantitative and qualitative comparative analysis of five experiments on DC motor.

of the optimal parameter combination, EDBO incorporates enhanced strategies into the DBO. This approach optimizes the neural network parameters, enhancing the effectiveness and robustness of the overall optimization mechanism. Simulation experiment results demonstrate that the proposed algorithm offers faster and more accurate adjustment capabilities, improved control robustness, and greater practical usability.

The proposed adaptive PID control algorithm has broad application potential in industrial systems, robotics, and energy management. It can adjust PID parameters in real time to adapt to complex, nonlinear systems. This approach enhances precision in robotic motor control, allows for real-time adjustments in industrial automation, and ensures load frequency control in microgrids, providing robustness and stability under changing conditions.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 7 September 2024; Accepted: 11 November 2024
Published online: 16 November 2024

References

1. Clarke, D. W. PID algorithms and their computer implementation. *Trans. Inst. Meas. Control*. **6**, 305–316. https://doi.org/10.1177/014233128400600605 (1984).
2. Wang, J. et al. PID control of multi-DOF industrial robot based on neural network. *J. Ambient Intell. Hum. Comput*. **11**, 6249–6260. https://doi.org/10.1007/s12652-020-01693-w (2020).
3. Aboelhassan, A., Abdelgeliel, M., Zakzouk, E. E. & Galea, M. Design and implementation of model predictive control based PID controller for industrial applications. *Energies*. **13**, 6594. https://doi.org/10.3390/en13246594 (2020).
4. Yang, T., Yi, X., Lu, S., Johansson, K. H. & Chai, T. Intelligent manufacturing for the process industry driven by industrial artificial intelligence. *Engineering*. **7**, 1224–1230. https://doi.org/10.1016/j.eng.2021.04.023 (2021).
5. Can, E. & Sayan, H. H. PID and fuzzy controlling three phase asynchronous machine by low level DC source three phase inverter. *Teh Vjesn*. **23**, 753–760. https://doi.org/10.17559/TV-20150106105608 (2016).
6. Somefun, O. A., Akingbade, K. & Dahunsi, F. The dilemma of PID tuning. *Annu. Rev. Control*. **52**, 65–74. https://doi.org/10.1016/j.arcontrol.2021.05.002 (2021).
7. Ding, X., Li, R., Cheng, Y., Liu, Q. & Liu, J. Design of and research into a multiple-fuzzy PID suspension control system based on road recognition. *Processes*. **9**, 2190. https://doi.org/10.3390/pr9122190 (2021).
8. Saleem, O. EKF-based self-regulation of an adaptive nonlinear PI speed controller for a DC motor. *Turkish J. Electr. Eng. Comput. Sci.* **25**, 4131–4141. https://doi.org/10.3906/elk-1611-311 (2017).
9. Zhu, F. et al. Research and design of hybrid optimized backpropagation (BP) neural network PID algorithm for integrated water and fertilizer precision fertilization control system for field crops. *Agronomy*. **13**, 1423. https://doi.org/10.3390/agronomy13051423 (2023).
10. Maraba, V. A. & Kuzucuoglu, A. E. PID neural network based speed control of asynchronous motor using programmable logic controller. *Adv. Electr. Comput. Eng.* **11**, 23–28. https://doi.org/10.4316/aece.2011.04004 (2011).
11. Cong, S. & Liang, Y. PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. *IEEE Trans. Ind. Electron.* **56**, 3872–3879. https://doi.org/10.1109/TIE.2009.2018433 (2009).
12. Ambroziak, A. & Chojecki, A. The PID controller optimisation module using fuzzy self-tuning PSO for Air Handling Unit in continuous operation. *Eng. Appl. Artif. Intell.* **117**, 105485. https://doi.org/10.1016/j.engappai.2022.105485 (2023).
13. Aygun, H., Demirel, H. & Cernat, M. Control of the bed temperature of a circulating gluidized bed boiler by using particle swarm optimization. *Adv. Electr. Comput. Eng.* **12**, 27–32. https://doi.org/10.4316/AECE.2012.02005 (2012).
14. Dahiya, P., Sharma, V. & Naresh, R. Solution approach to automatic generation control problem using hybridized gravitational search algorithm optimized PID and FOPID controllers. *Adv. Electr. Comput. Eng.* **15**, 23–34. https://doi.org/10.4316/AECE.2015.02004 (2015).
15. Suid, M. H. & Ahmad, M. A. Optimal tuning of sigmoid PID controller using nonlinear sine cosine algorithm for the automatic voltage regulator system. *ISA Trans.* **128**, 265–286. https://doi.org/10.1016/j.isatra.2021.11.037 (2022).

16. Sahin, A. K., Cavdar, B. & Ayas, M. S. An adaptive fractional controller design for automatic voltage regulator system: sigmoid-based fractional-order PID controller. *Neural Comput. Applic.* **36**, 14409–14431. https://doi.org/10.1007/s00521-024-09816-6 (2024).
17. Ghazali, M. R., Ahmad, M. A. & Raja Ismail, R. M. T. A multiple-node hormone regulation of neuroendocrine-PID (MnHR-NEPID) control for nonlinear MIMO systems. *IETE J. Res.* **68**, 4476–4491. https://doi.org/10.1080/03772063.2020.1795939 (2020).
18. Kumar, M. & Hote, Y. V. Maximum sensitivity-constrained coefficient diagram method-based PIDA controller design: application for load frequency control of an isolated microgrid. *Electr. Eng.* **103**, 2415–2429. https://doi.org/10.1007/s00202-021-01226-4 (2021).
19. Shi, X., Zhao, H. & Fan, Z. Parameter optimization of nonlinear PID controller using RBF neural network for continuous stirred tank reactor. *Meas. Control.* **56**, 1835–1843. https://doi.org/10.1177/00202940231189307 (2023).
20. Hanna, Y. F., Khater, A. A., El-Bardini, M. & El-Nagar, A. M. Real time adaptive PID controller based on quantum neural network for nonlinear systems. *Eng. Appl. Artif. Intell.* **126**, 106952. https://doi.org/10.1016/j.engappai.2023.106952 (2023).
21. Zhao, W. & Gu, L. Adaptive PID controller for active suspension using radial basis function neural networks. *Actuators*. **12**, 437. https://doi.org/10.3390/act12120437 (2023).
22. Kebari, M., Wu, A. S. & Mathias, H. D. PID-inspired modifications in response threshold models in swarm intelligent systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 39–46. (2023). https://doi.org/10.1145/3583131.3590442
23. Gupta, S., Debnath, S. & Biswas, P. K. Control of an active magnetic bearing system using swarm intelligence-based optimization techniques. *Electr. Eng.* **105**, 935–952. https://doi.org/10.1007/s00202-022-01707-0 (2023).
24. Faria, R. M. et al. Particle swarm optimization solution for roll-off control in radiofrequency ablation of liver tumors: optimal search for PID controller tuning. *PLOS ONE*. **19**, e0300445. https://doi.org/10.1371/journal.pone.0300445 (2024).
25. Nanyan, N. F., Ahmad, M. A. & Hekimoğlu, B. Optimal PID controller for the DC-DC Buck converter using the improved sine cosine algorithm. *Results Control Optim.* **14**, 100352. https://doi.org/10.1016/j.rico.2023.100352 (2024).
26. Mourtas, S. D., Kasimis, C. & Katsikis, V. N. Robust PID controllers tuning based on the beetle antennae search algorithm. *Memories – Mater. Devices Circuits Syst.* **4**, 100030. https://doi.org/10.1016/j.memori.2023.100030 (2023).
27. Ghith, E. S. & Tolba, F. A. A. tuning PID controllers based on hybrid arithmetic optimization algorithm and artificial gorilla troop optimization for micro-robotics systems. *IEEE Access.* **11**, 27138–27154. https://doi.org/10.1109/ACCESS.2023.3258187 (2023).
28. Xue, J. & Shen, B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. *J. Supercomput.* **79**, 7305–7336. https://doi.org/10.1007/s11227-022-04959-6 (2023).
29. Tang, J., Duan, H. & Lao, S. Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: a comprehensive review. *Artif. Intell. Rev.* **56**, 4295–4327. https://doi.org/10.1007/s10462-022-10281-7 (2023).
30. Xu, M., Cao, L., Lu, D., Hu, Z. & Yue, Y. Application of swarm intelligence optimization algorithms in image processing: a comprehensive review of analysis, synthesis, and optimization. *Biomimetics*. **8**, 235. https://doi.org/10.3390/biomimetics8020235 (2023).
31. Tawhid, M. A. & Ibrahim, A. M. An efficient hybrid swarm intelligence optimization algorithm for solving nonlinear systems and clustering problems. *Soft Comput.* **27**, 8867–8895. https://doi.org/10.1007/s00500-022-07780-8 (2023).
32. Miao, Y. et al. Research on optimal control of HVAC system using swarm intelligence algorithms. *Build. Environ.* **241**, 110467. https://doi.org/10.1016/j.buildenv.2023.110467 (2023).
33. Atacak, I. & Küçük, B. PSO-based PID controller design for an energy conversion system using compressed air. *Teh Vjesn.* **24**, 671–679. https://doi.org/10.17559/TV-20150310170741 (2017).
34. Mirjalili, S. Genetic algorithm. In Evolutionary Algorithms and Neural Networks: Theory and Applications (ed Mirjalili, S.) 43–55 (Springer, Cham, https://doi.org/10.1007/978-3-319-93025-1_4. (2019).
35. Nayak, J., Swapnarekha, H., Naik, B., Dhiman, G. & Vimal, S. 25 years of particle swarm optimization: flourishing voyage of two decades. *Arch. Computat Methods Eng.* **30**, 1663–1725. https://doi.org/10.1007/s11831-022-09849-x (2023).
36. Chakraborty, A. & Kar, A. K. Swarm intelligence: a review of algorithms. In Nature-Inspired Computing and Optimization: Theory and Applications (eds Patnaik, S., Yang, X. S. & Nakamatsu, K.) 475–494 (Springer, Cham, https://doi.org/10.1007/978-3-319-50920-4_19. (2017).
37. Zhu, F. et al. Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems. *Expert Syst. Appl.* **236**, 121219. https://doi.org/10.1016/j.eswa.2023.121219 (2024).
38. Hasan, M. M., Rana, M. S., Tabassum, F., Pota, H. R. & Roni, M. H. K. optimizing the initial weights of a PID neural network controller for voltage stabilization of microgrids using a PEO-GA algorithm. *Appl. Soft Comput.* **147**, 110771. https://doi.org/10.1016/j.asoc.2023.110771 (2023).
39. He, Y., Zhou, Y., Wei, Y., Luo, Q. & Deng, W. Wind driven butterfly optimization algorithm with hybrid mechanism avoiding natural enemies for global optimization and PID controller design. *J. Bionic Eng.* **20**, 2935–2972. https://doi.org/10.1007/s42235-023-00416-z (2023).

Acknowledgements

This research was funded by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX23-1871, and No. KYCX24-XZ054); Yancheng Institute of Technology Teaching Reform Research Project (Grant No. YKT2022A028); The Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 24KJB140020).

Author contributions

Data collection, analysis, and interpretation, writing—original draft preparation, H.Z.; methodology, formal analysis, W.K.; writing—review and editing, X.Y., Z.Y., R.W., W.Y., and J.Z. All authors have read and agreed to the published version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.Y.