

Article

PID Control Model Based on Back Propagation Neural Network Optimized by Adversarial Learning-Based Grey Wolf Optimization

Huaiqin Liu, Qinghe Yu and Qu Wu *

School of Information and Control Engineering, Qingdao University of Technology, No. 777 Jialingjiang East Rd., Qingdao 266525, China

* Correspondence: wq_wuqu@126.com

Abstract: In processes of industrial production, the online adaptive tuning method of proportional-integral-differential (PID) parameters using a neural network is found to be more appropriate than a conventional controller with PID for controlling different industrial processes with varying characteristics. However, real-time implementation and high reliability require the adjustment of specific model parameters. Therefore, this paper proposes a PID controller that combines a back-propagation neural network (BPNN) and adversarial learning-based grey wolf optimization (ALGWO). To enhance the unpredictable behavior and capacity for exploration of the grey wolf, this study develops a new parameter-learning technique. Alpha gray wolves use the random walk of levy flight as their hunting method. In beta and delta gray wolves, a search strategy centering on the top gray wolf is employed, and in omega gray wolves, the decision wolves handle the confrontation strategy. A fair balance between exploration and exploitation can be achieved, as evidenced by the success of the adversarial learning-based grey wolf optimization technique in ten widely used benchmark functions. The effectiveness of different activation functions in conjunction with ALGWO were evaluated in resolving the parameter adjustment issue of the BPNN model. The results demonstrate that no unique activation function outperforms others in different controlled systems, but their fitnesses are significantly inferior to those of the conventional PID controller.



Citation: Liu, H.; Yu, Q.; Wu, Q. PID Control Model Based on Back Propagation Neural Network Optimized by Adversarial Learning-Based Grey Wolf Optimization. *Appl. Sci.* **2023**, *13*, 4767. <https://doi.org/10.3390/app13084767>

Academic Editor: Amerigo Capria

Received: 9 March 2023

Revised: 28 March 2023

Accepted: 6 April 2023

Published: 10 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: gray wolf optimization algorithm; back propagation neural network; activation functions; PID controller

1. Introduction

In control manufacturing, standard PID control algorithms are frequently utilized as a control strategy, providing the benefits of unequivocal simplicity and excellent performance [1], among others. PID control algorithms, alongside traditional methods, are mainly used for control-parameter optimization [2,3]. However, the PID control algorithm has several drawbacks, including difficulty in parameter setting and susceptibility to disturbance. More PID tuning algorithms, such as neural networks [4], machine learning [5,6], and heuristic algorithms [7,8], should be developed for use in actual control processes to address these issues.

However, as controlled systems become increasingly complex, there is a growing need for control strategies that can handle time-varying behavior and time delays. The complexity of the controlled system contributes to the frequently subpar control effects of a PID controller [9,10]. At present, many intelligent algorithms have been developed. Systems with time-varying and hysteresis delays can benefit significantly from these approaches [11,12]. By manually modifying the control rules, the expert PID control algorithm can be applied to specific simulation systems [13]. Fuzzy PID control can be included in the controller using a lookup table, making it simpler to obtain better control effects [14–19]. However,

the expert PID and fuzzy PID controllers always have subpar timing precision and limited anti-interference capabilities. Due to the benefits of rapid adjustment velocity and excellent accuracy [20,21], the neural network PID control approaches have been applied in numerous complicated systems [22–24]. In 2022, Wang et al. [25] used the PSO-BPNN-PID algorithm to control the nutrient solution, and the simulation results demonstrated that the reaction time and accuracy of the control system were superior to those of the conventional PID control method. The same year, a BPNN-PID controller was developed using a Xilinx field-programmable gate array (FPGA) technology in [26]. The findings revealed that the suggested system exhibited fast convergence and dependable performance. To better regulate the delay system in 2023, a particle swarm optimization-radial basis function (PSO-RBF) is utilized in place of a traditional PID controller [27]. In general, BP neural network research has thus shown positive results.

In addition, to achieve effective controller performance, these parameters of the BPNN-PID controller must be properly optimized. Moreover, heuristic methods are typically used in the process of parameter optimization [28–32]. In 2018, the BP neural network was optimized using a mind evolutionary algorithm to predict wave heights; the model achieved great accuracy and a fast running time [33]. In 2021, using an algorithm known as the water cycle method, Zhang et al. [34] improved the BP neural network model for landside prediction. In 2022, it was proposed that the Boost circuit's dynamic and generally pro capabilities be improved by combining a genetic algorithm with BP neural network PID control (GA-BP-PID) [35]. Therefore, an ALGWO-BPNN control model is proposed in this paper as a better way to control time-varying and time delay systems. The primary contributions of the proposed controller are as follows:

1. The GWO technique is used to propose and optimize a PID control model based on a BPNN. Considering that the GWO algorithm is not necessarily the best for solving complex problems, a stochastic adjustment convergence factor formula is proposed, and the beta and delta are adjusted in combination with a differential evolution (DE). With the enhanced GWO method, the connection weight of the BPNN achieved a verifiably better control effect.
2. Several benchmark functions are selected to analyze ALGWO.
3. Some activation functions for the output layer of the BPNN model are analyzed and compared.
4. The ALGWO-BPNN model is simulated and compared with the conventional PID controller.

2. Related Works

2.1. Back Propagation Neural Network

Using the error back propagation technique, a multilayer feedforward neural network is called to become a back propagation neural network. Due to its nature, it has outstanding performance in nonlinear mapping, such as function approximation and pattern recognition.

There are three layers in a back propagation neural network model: input, hidden, and output. The structure of the fundamental BPNN model is depicted in Figure 1. The input layer deals with the kind and volume of input. By the quantity of control layers and activation functions, the hidden layer introduces the possibility for nonlinear mapping. The output layer is in charge of producing certain information.

Output of the neuron model structure is usually expressed as a nonlinear combination of input and weight, as shown in (1).

$$\hat{y}_j = f\left(\sum_{i=1}^n w_{ij}x_i - b_{ij}\right) \quad (1)$$

where \hat{y}_j is output of the neuron, x_i is output of the neuron, b_{ij} and w_{ij} are the bias and weight of the neuron, and $f(\cdot)$ is an activation function.

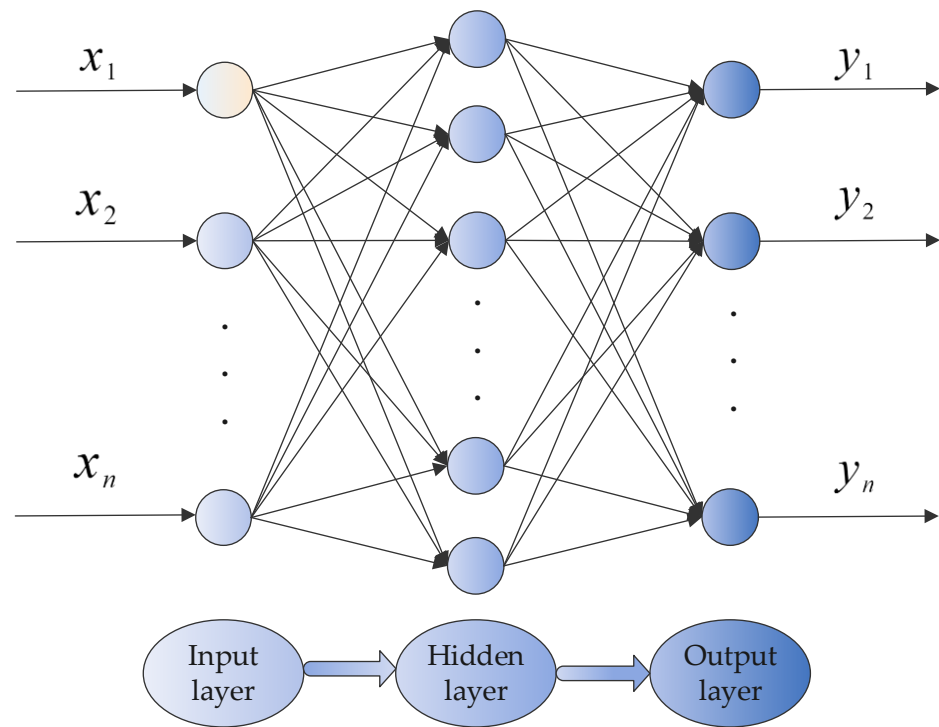


Figure 1. Basic structure of a typical back propagation neural network (BPNN).

Typical loss function for optimization is shown in (2).

$$E = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \tag{2}$$

where \hat{y}_i is the truth value and E is a loss function.

Gradient Descent (GD) technology is typically used for weight minimization [36].

$$w_{ij} = w_{ij} - \eta \frac{\partial E(w_{ij})}{\partial w_{ij}} \tag{3}$$

where η is a learning rate.

2.2. Grey Wolf Optimization Algorithm

Authors of [37] examined the grey wolf’s social behavior before modeling it and creating an algorithm for the grey wolf’s optimization.

The pros and cons of the actual problems’ remedies are used to categorize the entire grey wolf population. The best option is designated as the alpha (α), the second best option is designated as the beta (β), the third best option is designated as the delta (δ), and the remaining gray wolves are designated as the omega (ω).

The following describes how gray wolves’ behavior toward nearby prey is mathematically modeled:

$$\vec{D}_i = \left| \vec{W}_{prey} \cdot \vec{X}_{prey} - \vec{X}_i \right| \tag{4}$$

$$\vec{X}_i = \vec{X}_{prey} - \vec{A}_i \cdot \vec{D}_i \tag{5}$$

where \vec{X}_{prey} is the position of the prey, \vec{X}_i is the position vector of the grey wolf, and \vec{A}_i and \vec{W}_{prey} are coefficient vectors.

The \vec{A}_i and \vec{W}_{prey} are calculated as follows:

$$\vec{A}_i = 2 \cdot rand \cdot \vec{a} - \vec{a} \tag{6}$$

$$\vec{W}_{prey} = 2 \cdot rand \tag{7}$$

where \vec{a} decreases linearly and $rand$ are random vectors.

Because we do not know where the prey is, we can only roughly determine where the gray wolf is by using our decision-making level to mimic the gray wolf’s hunting behavior. Through the following, each gray wolf can update its location:

$$\begin{cases} \vec{D}_\alpha = \left| \vec{W}_\alpha \cdot \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta = \left| \vec{W}_\beta \cdot \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta = \left| \vec{W}_\delta \cdot \vec{X}_\delta - \vec{X} \right| \end{cases} \tag{8}$$

$$\begin{cases} \vec{X}_{i\alpha} = \left| \vec{X}_\alpha - \vec{A}_\alpha \cdot \vec{D}_\alpha \right| \\ \vec{X}_{i\beta} = \left| \vec{X}_\beta - \vec{A}_\beta \cdot \vec{D}_\beta \right| \\ \vec{X}_{i\delta} = \left| \vec{X}_\delta - \vec{A}_\delta \cdot \vec{D}_\delta \right| \end{cases} \tag{9}$$

$$\vec{X}_i = \frac{\vec{X}_{i\alpha} + \vec{X}_{i\beta} + \vec{X}_{i\delta}}{3} \tag{10}$$

3. Model Optimization

The standard grey wolf optimization algorithm has all omega members update their positions until they hit the termination conditions and discover the best answer (alpha) by learning from the first three best leaders. Even when the alpha falls into the local optimum, the standard grey wolf optimization algorithm still has good performance and the ability of fast collection, but the effect is relatively general when solving the problem of complex search space. As a conclusion, the proposed ALGWO enhances strategy optimization from two perspectives, namely the adaptive weight and gray wolf location update methods, predicated on the standard GWO.

3.1. Random Adjustment Strategy

The parameter established by the grey wolf algorithm to balance the capabilities of exploration and exploitation is called the convergence factor. The non-decreasing strategy increases the probability of selecting a larger step size at the beginning of the iteration, increasing the exploration capabilities. At the end of the iteration, the grey wolf is forced to select a smaller step size to increase its development capabilities. The convergence factor \vec{a} in this study has the potential to take on a greater or smaller value during the algorithm’s iterative procedure by applying the random selection technique. Hence, by achieving a smaller value in the early stages, the convergence speed can be sped up, and by acquiring a higher value in the latter stages, the local optimum can be jumped. These three curves are shown in Figure 2.

$$a_1 = 2 - 2 * \frac{t}{T} \tag{11}$$

$$a_2 = 2 - 2 * \ln\left(1 + \frac{t}{T}(e - 1)\right) \tag{12}$$

$$a_3 = 2 - 2 * \frac{e^{\frac{t}{T}} - 1}{e - 1} \tag{13}$$

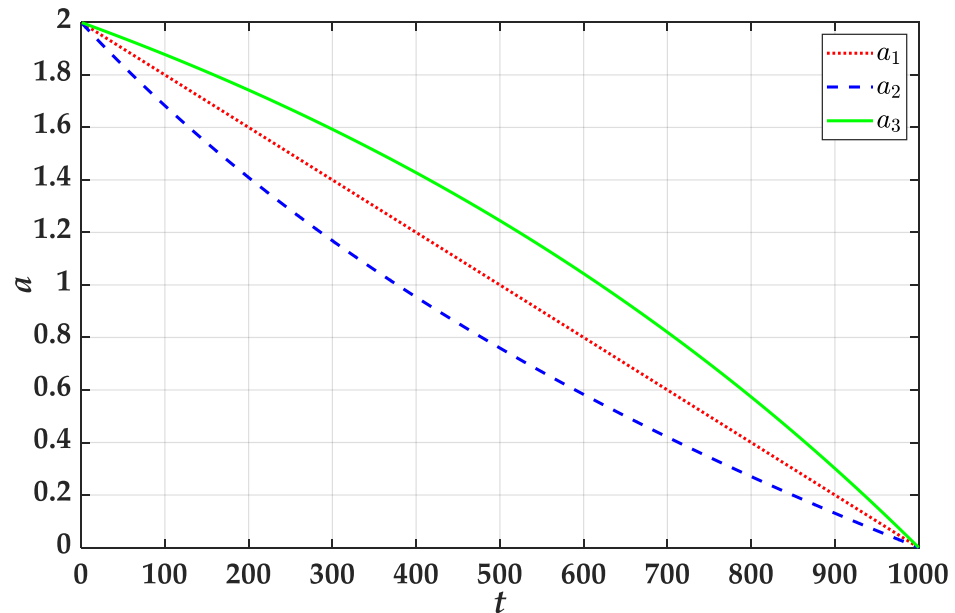


Figure 2. The change of convergence factor.

3.2. Decision-making Level Update Scheme

Since the alpha gray wolf already represents the optimal value in the current population of gray wolves, and since the gray wolf does not benefit from the gray wolf’s optimal solution, the location update in this study uses levy flight instead of the sub-optimal or third optimal solutions.

$$\vec{X}_\alpha = \vec{X}_\alpha + \vec{s} \tag{14}$$

where \vec{s} is calculated from the formula proposed by Mantegna.

The differential evolution idea is utilized to update the position to strengthen the leadership influence of the alpha gray wolf over potential leaders in beta and delta gray wolves because of the differential evolution algorithm’s strong local search capability in multimodal function search and grey wolf.

$$\begin{cases} \vec{X}_\beta = \vec{X}_\beta + \vec{F} \cdot (\vec{X}_\alpha - \vec{X}_\delta) \\ \vec{X}_\delta = \vec{X}_\delta + \vec{F} \cdot (\vec{X}_\alpha - \vec{X}_\beta) \end{cases} \tag{15}$$

The populations of the beta (β) and the delta (δ) are set to one or more, and \vec{F} is the variation factor. It declines in line with the amount of iterations from 0.6 to 0.3.

The entire grey wolf group may slip into the local optimum and lose the ability to perform global searches under the formula and formula update if the alpha (α), beta (β), and delta (δ) grey wolves are in the local optimum in the multimodal function. Hence, the conflict between candidate leaders and leaders can be employed to resolve the decline in optimization accuracy induced by this condition.

$$\begin{cases} \vec{X}_i = \frac{\vec{X}_{i\alpha} + \vec{X}_{i\beta} + \vec{X}_{i\delta}}{3}, r > q \\ \vec{X}_i = \frac{\vec{X}_{i\alpha} - \vec{X}_{i\beta} - \vec{X}_{i\delta}}{3}, r \leq q \end{cases} \tag{16}$$

where the beta (β) and the delta (δ) are the second and third candidate solutions of the problem; q is the preset selection threshold, and it decreases linearly with the number of iterations from 0.9 to 0.6.

3.3. PID Controller Based on Back Propagation Neural Network

The back propagation neural network maps the input, output, and the errors nonlinearly to the PID controller’s three parameters, k_p , k_i , and k_d . In addition, the BP neural network has three neuron points for the input layer, five for the buried layer, and three for the output layer. The frequently employed Tanh function is utilized in the hidden layer.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{17}$$

The three no-negative gain parameters of the PID control scheme are output by BP neural networks, hence sigmoid functions and other functions with no-negative output values are applied.

$$g(x) = ub \cdot \frac{1}{1 + e^{-x}} \tag{18}$$

$$t(x) = ub \cdot \frac{e^x}{e^x + e^{-x}} \tag{19}$$

$$h(x) = \min(\max(0, x), ub) \tag{20}$$

where ub is upper bound of the output. It is employed to regulate the output range.

The structure of the PID control scheme based on a back propagation neural network is shown in Figure 3.

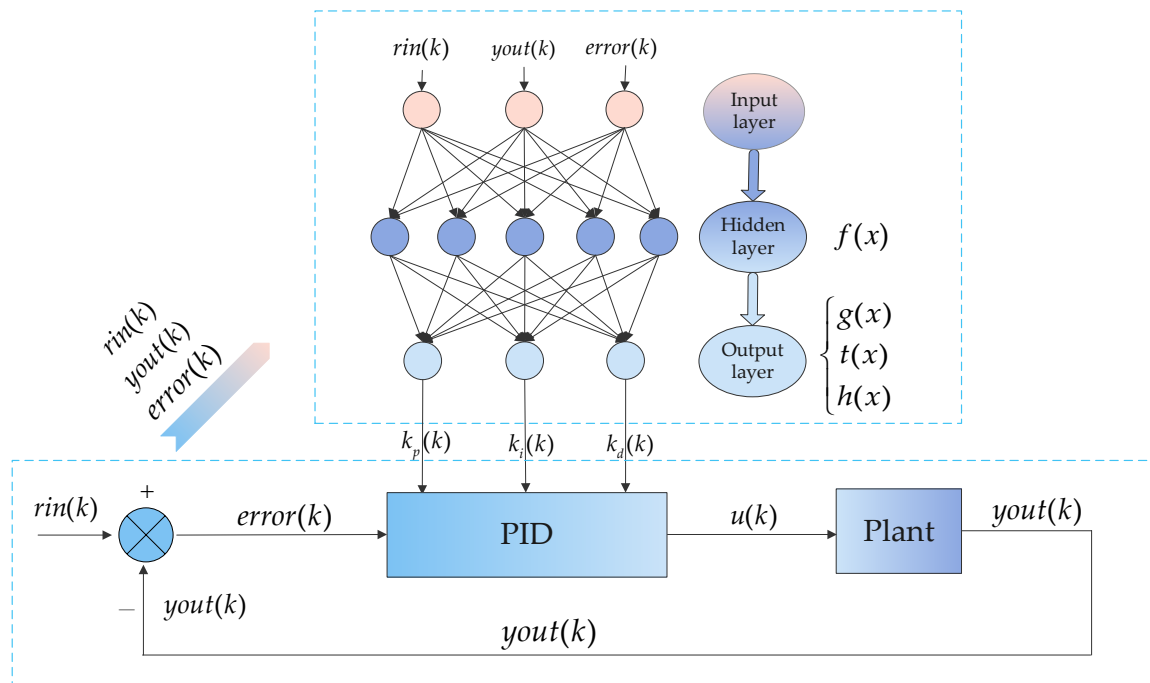


Figure 3. The PID controller structure based on back propagation neural network.

3.4. Structure of the Total Algorithm

The search space in this paper consists of the network weight and hyperparameters, since they have a major impact on the performance of the back propagation neural network model. The system’s process is shown in Figure 4.

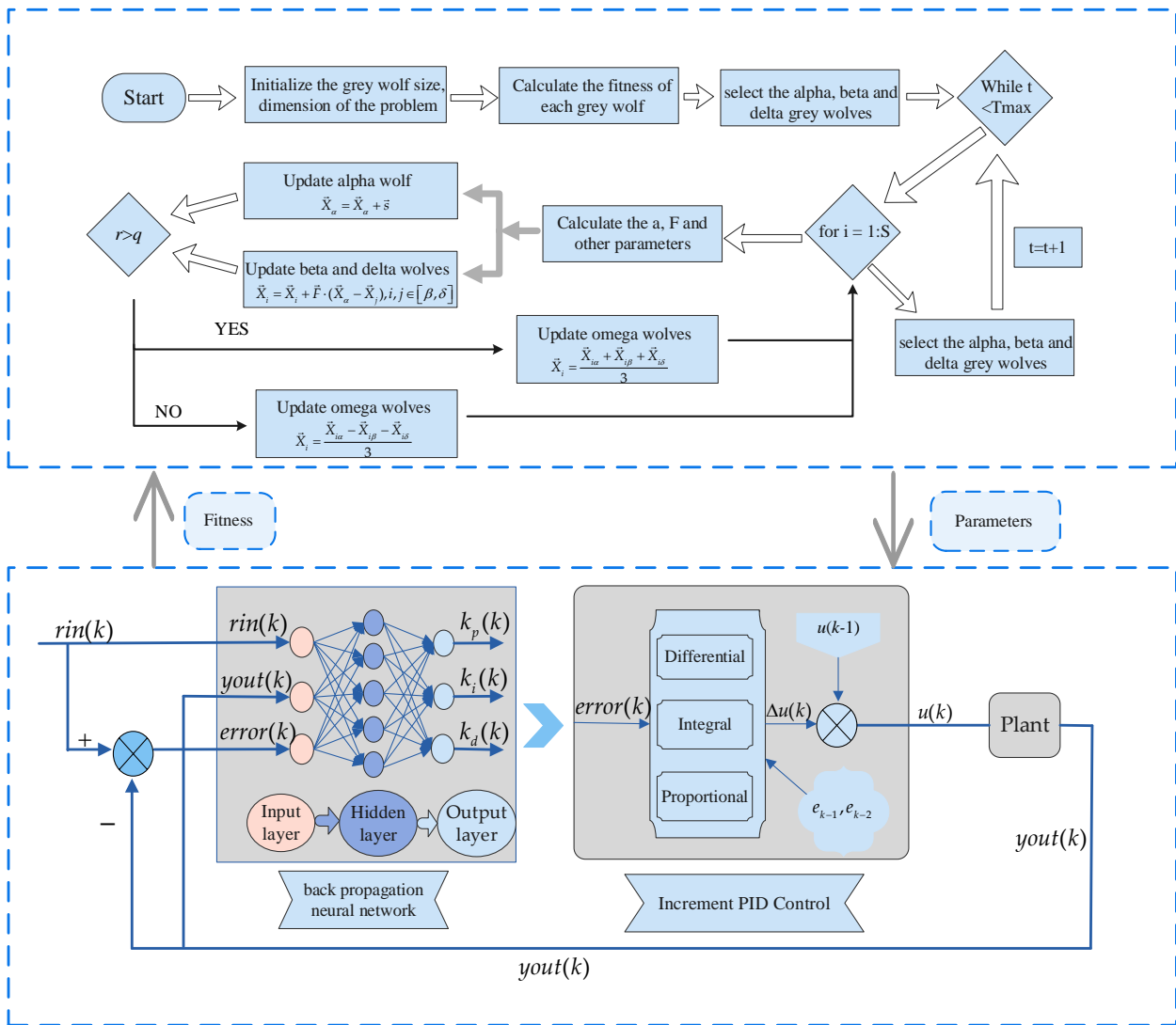


Figure 4. The process of total algorithm.

4. Results and Discussion

4.1. Experiment Settings

Ten different test functions are employed to gauge the performance of the proposed method in order to demonstrate its usefulness. To guarantee a fair comparison, the population size in this paper is fixed at 30 and the number of iterations is set at 1000. The parameters of different algorithms that might be used are displayed in Table 1. In order to avoid random variation, tests are carried out 10 times independently.

Table 1. Parameter setups of different algorithms.

Algorithm	Values of the Parameters
GWO	a = 2 (linear reduction during iteration)
SOGWO [38]	a = 2 (linear reduction during iteration)
PSO	c ₁ = 2, c ₂ = 2, w _{Max} = 0.9, w _{Min} = 0.2
DE	F = 0.5, CR = 0.5

However, additional analysis is required to determine how well the ALGWO-BPNN-PID algorithm performs in control issues. In order to assess the superiority of the ALGWO-BPNN control technique, the four tests are also performed using the fitness value, overshoot, and settling time performances. The simulation’s settings are similar. As a controller

model, the incremental digital PID is employed. The sampling interval is set at one. The $g(x)$ and the other function are the activation functions described above. The output control signal of the controller is restricted to $[-10, 10]$. Equation (21) is used to assess the algorithm’s effectiveness.

$$J = \int_0^{\infty} (w_1|e(t)| + w_2u^2(t))dt \tag{21}$$

where $e(t)$ is the system error and $u(t)$ is the controller output, and w_1 and w_2 are the weights, with values of 0.001 and 0.999. Furthermore, overshoot will cause the fitness to increase more quickly.

4.2. Test Systems

4.2.1. Test System 1

Six unimodal functions are included in the test functions. PSO and GWO are contrasted with the suggested ALGWO algorithm’s solution. The unimodal test functions are described in Table 2. Figure 5 displays the two-dimensional parameter space for these six unimodal functions to show them more easily.

Table 2. Details of the unimodal test function.

Function Name	Expression	Search Space	Dim
F1	$F1 = \sum_{i=1}^n x_i^2$	$[-100,100]$	30
F2	$F2 = \sum_{i=1}^n x_i - \prod_{i=1}^n x_i $	$[-10,10]$	30
F3	$F3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)$	$[-100,100]$	30
F4	$F4 = \max_i\{ x_i , 1 \leq i \leq n\}$	$[-100,100]$	30
F5	$F5 = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]$	30
F6	$F6 = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100,100]$	30

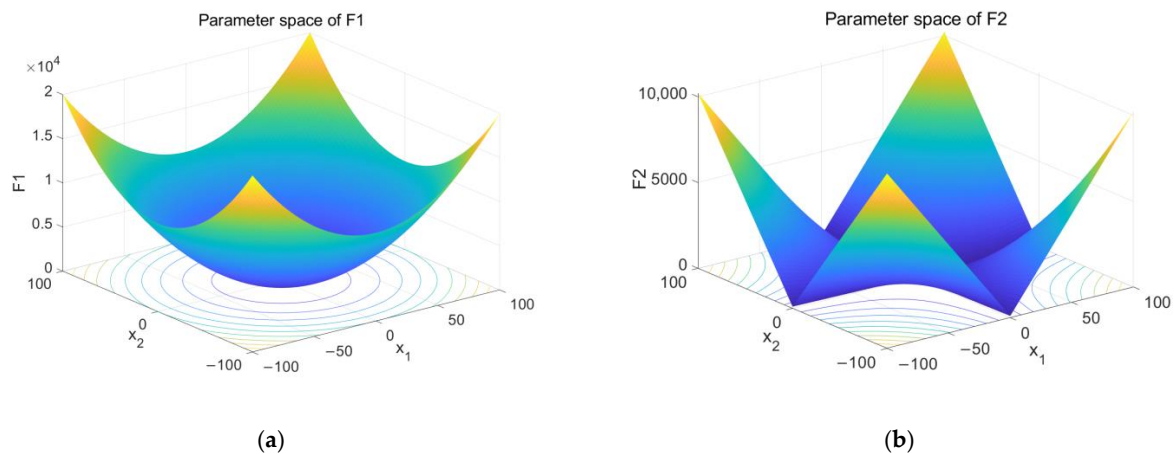


Figure 5. Cont.

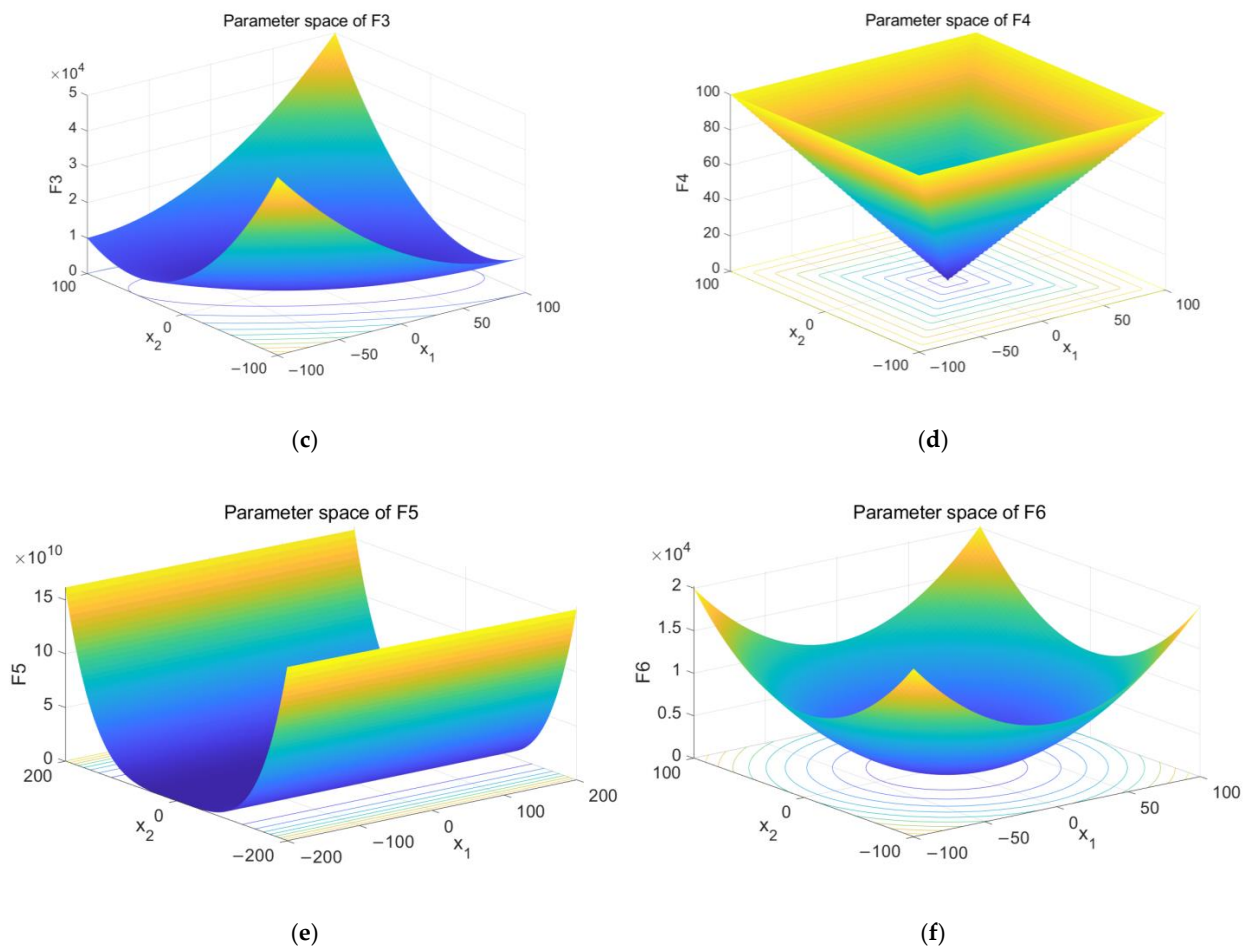


Figure 5. The results of parameter space. (a) The parameter space of F1 and (b) the parameter space of F2 and (c) the parameter space of F3 and (d) the parameter space of F4 and (e) the parameter space of F5 and (f) the parameter space of F6.

The average and standard deviations for ten independent runs used to assess the resilience and average accuracy of algorithms are shown in Table 3. For each function, the best results are displayed in bold. Figure 6 shows the convergence curves of the six test functions.

Table 3. Comparison results of three algorithms on six groups of benchmark functions.

Base Function		GWO	ALGWO	PSO
F1	STD	6.69×10^{-70}	0	1.40×10^{-11}
	AVE	2.49×10^{-70}	0	5.48×10^{-12}
F2	STD	3.49×10^{-41}	0	1.01×10^{-5}
	AVE	3.89×10^{-41}	3.01×10^{-238}	7.24×10^{-6}
F3	STD	1.57×10^{-20}	0	3.47×10^0
	AVE	8.72×10^{-21}	0	7.16×10^0
F4	STD	2.04×10^{-17}	0	1.20×10^{-1}
	AVE	1.77×10^{-17}	2.70×10^{-206}	3.90×10^{-1}
F5	STD	8.78×10^{-1}	4.90×10^{-1}	3.46×10^1
	AVE	2.66×10^1	2.62×10^1	5.04×10^1
F6	STD	2.71×10^{-1}	1.09×10^0	1.59×10^{-11}
	AVE	3.00×10^{-1}	3.58×10^{-1}	8.16×10^{-12}

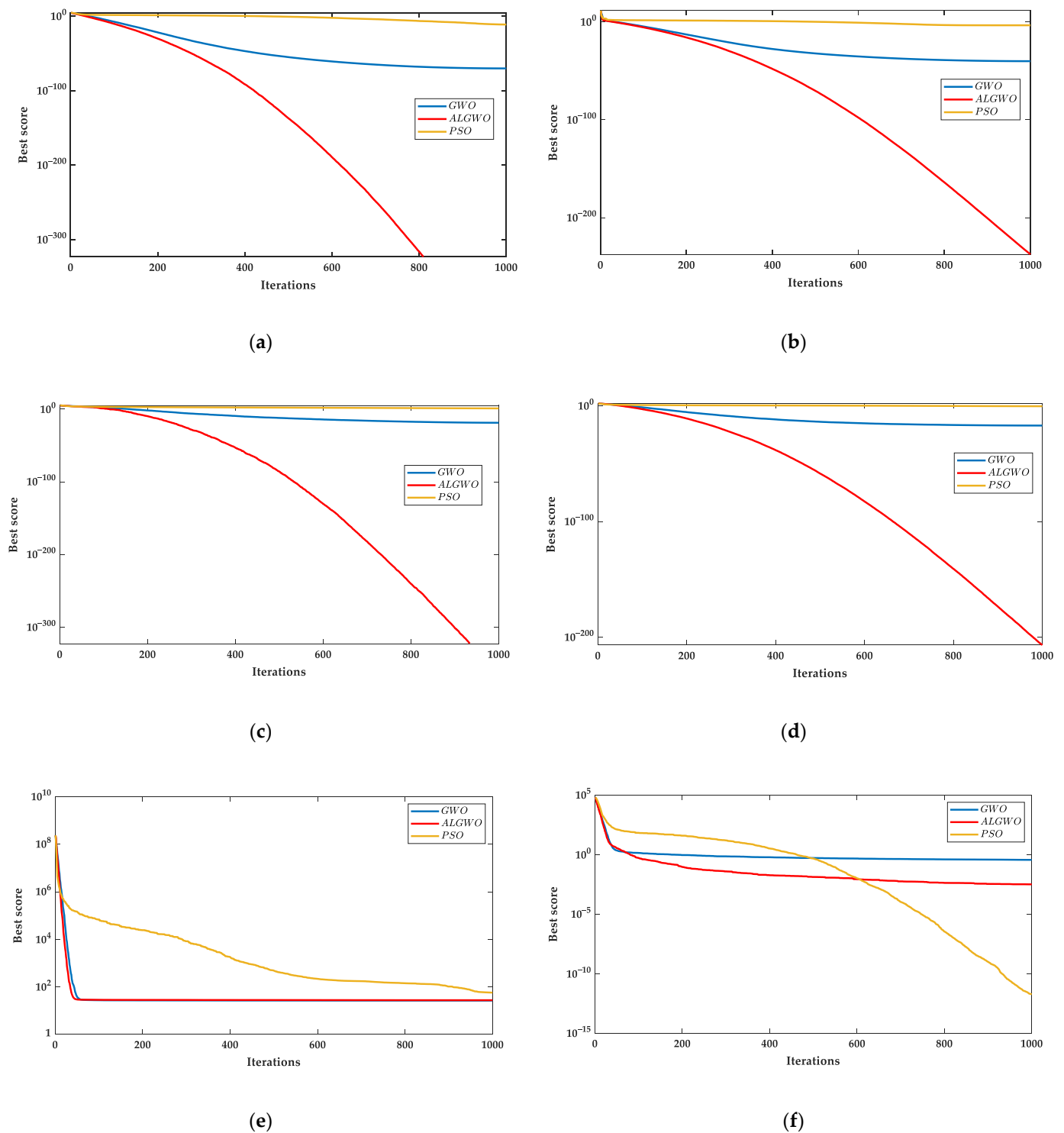


Figure 6. The results of convergence curves. (a) The convergence curves of F1 and (b) the convergence curves of F2 and (c) the convergence curves of F3 and (d) the convergence curves of F4 and (e) the convergence curves of F5 and (f) the convergence curves of F6.

As shown in Table 3, the ALGWO algorithm indications are usually noticeably better than those of other algorithms. Figure 6 illustrates that the ALGWO algorithm provides better solution outcomes and faster convergence. As a result, the ALGWO method performs the best overall when solving unimodal test functions when compared to other algorithms.

4.2.2. Test System 2

Four multimodal functions are included in the test functions. They are compared to four metaheuristic algorithms that are currently in use. PSO, DE, GWO, and SOGWO are the algorithms utilized for comparison. Table 4 displays the specifics of the multimodal test function. The two-dimensional parameter space of these four unimodal functions is depicted in Figure 7.

Table 4. Details of the multimodal test function.

Function Name	Expression	Search Space	Dim
F8	$F8 = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	[-500,500]	30
F9	$F9 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]	30
F10	$F10 = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} + 20 + e$	[-32,32]	30
F11	$F11 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	30

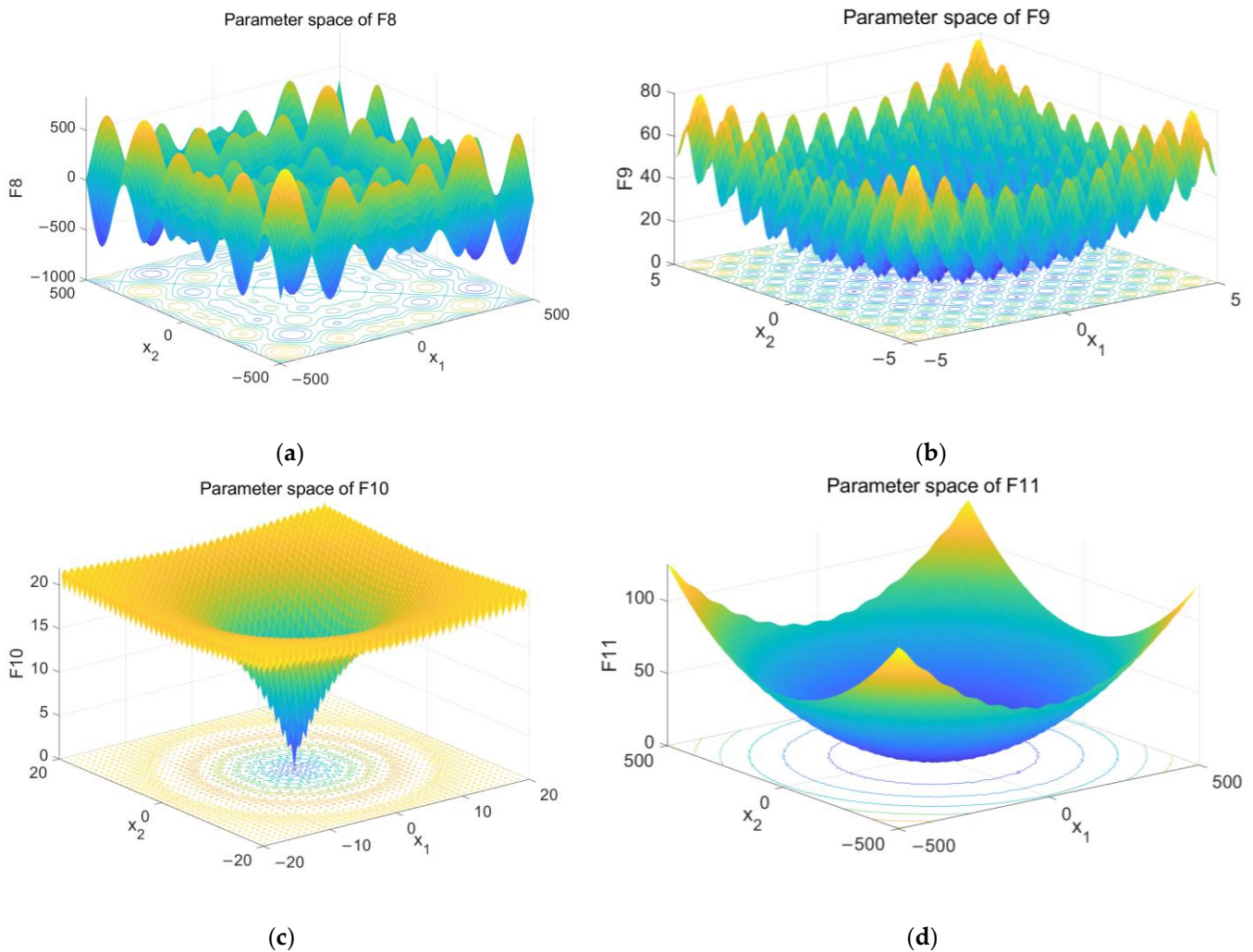
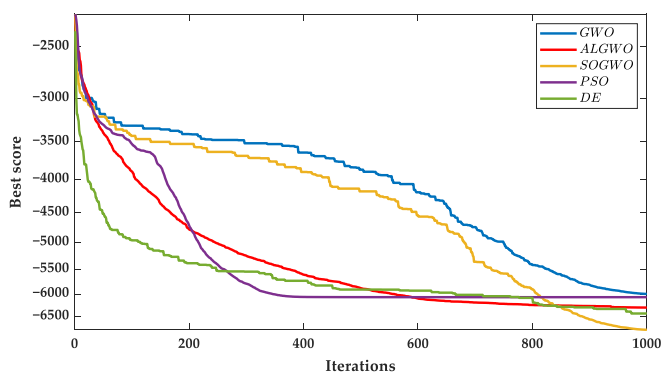


Figure 7. The results of parameter space. (a) The parameter space of F8 and (b) the parameter space of F9 and (c) the parameter space of F10 and (d) the parameter space of F11.

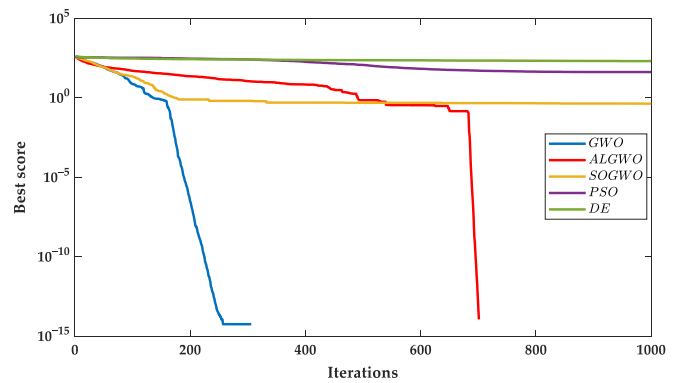
Table 5 contains the average and standard deviation for ten independent runs. The convergence rates of the four test functions for several algorithms are displayed in Figure 8.

Table 5. Comparison results of five algorithms on four groups of benchmark functions.

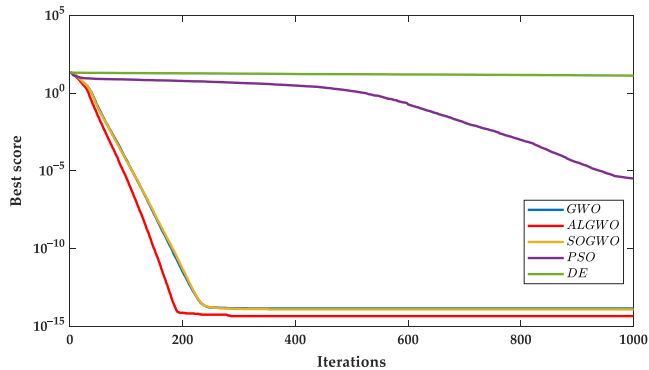
Base Function		GWO	ALGWO	SOGWO	PSO	DE
F8	STD	8.45×10^2	1.26×10^3	7.11×10^2	8.35×10^2	6.86×10^2
	AVE	-5.98×10^3	-5.98×10^3	-6.44×10^3	-6.80×10^3	-5.97×10^3
F9	STD	0	0	0	8.22×10^0	1.57×10^1
	AVE	0	0	0	3.47×10^1	2.57×10^2
F10	STD	1.49×10^{-15}	0	2.39×10^{-15}	3.59×10^{-6}	7.20×10^{-1}
	AVE	1.43×10^{-14}	4.44×10^{-15}	1.40×10^{-14}	2.56×10^{-6}	1.72×10^{-1}
F11	STD	0	0	0	7.30×10^{-3}	2.67×10^0
	AVE	0	0	0	7.10×10^{-3}	3.63×10^{-1}



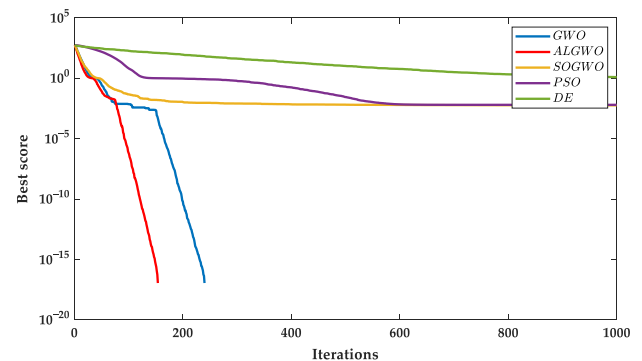
(a)



(b)



(c)



(d)

Figure 8. The results of convergence curves. (a) The convergence curves of F8 and (b) the convergence curves of F9 and (c) the convergence curves of F10 and (d) the convergence curves of F11.

Table 5 demonstrates that, despite the ALGWO algorithm’s poor performance in the multimodal test function in the F8 function, the average search accuracy achieves the minimum value of 4.44×10^{-15} in F10 and the optimal solution of 0 in F9 and F11. As a result, the ALWO algorithm’s indication is much superior to that of other algorithms. Figure 8 shows that, for the test functions of F10 and F11, the ALGWO algorithm yields superior solution outcomes and accelerates convergence speed. Therefore, when solving multimodal test functions, the ALGWO method generally outperforms other algorithms.

4.2.3. Test System 3

The first-order linear system is chosen for testing in order to confirm the superiority of the ALGWO-BPNN in the PID control process, and the model transfer function is indicated in (22). The value of the ub is 10, so the mapping range of the activation function is set to

0 to 10. The control findings are shown in Table 6 and Figure 9. The best performance is demonstrated by the $g(x)$ activation function. The convergence characteristics can be seen in Figure 10.

$$G(s) = \frac{1.7}{320s + 1} \tag{22}$$

Table 6. The comparison of final fitness of algorithms on test system 3.

Algorithm		$g(x)$	$t(x)$	$h(x)$	PID
GWO	AVE	277.2945	295.4016	280.0701	301.2233
	STD	13.5263	13.4028	13.5383	0.2515
ALGWO	AVE	275.7396	280.0453	278.7984	301.0937
	STD	2.4216	12.3446	11.8726	0.2811

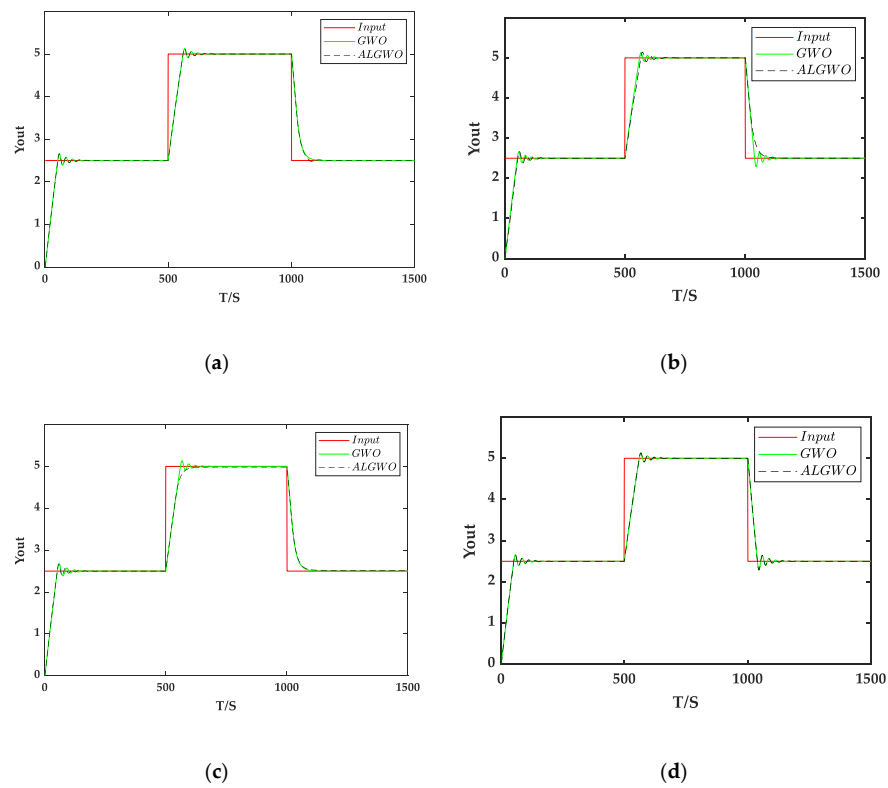


Figure 9. The results of control effect. (a) The simulation results of BPNN-PID using $g(x)$ and (b) simulation results of BPNN-PID using $t(x)$ and (c) simulation results of BPNN-PID using $h(x)$ and (d) simulation results of conventional PID.

The variance and standard deviation of utilizing the ALGWO method are noticeably lower, as seen in Table 6, as compared to using the GWO algorithm. For instance, the PID control system utilizing the $g(x)$ BPNN reduces the standard deviation from 13.5263 to 2.4216 and also lowers the average value by 1.5. As a result, it has improved resilience and average accuracy. The average value is decreased by roughly 20 after using the BPNN model, so the average accuracy is increased. The $g(x)$ activation function has a better impact when utilized as an activation function.

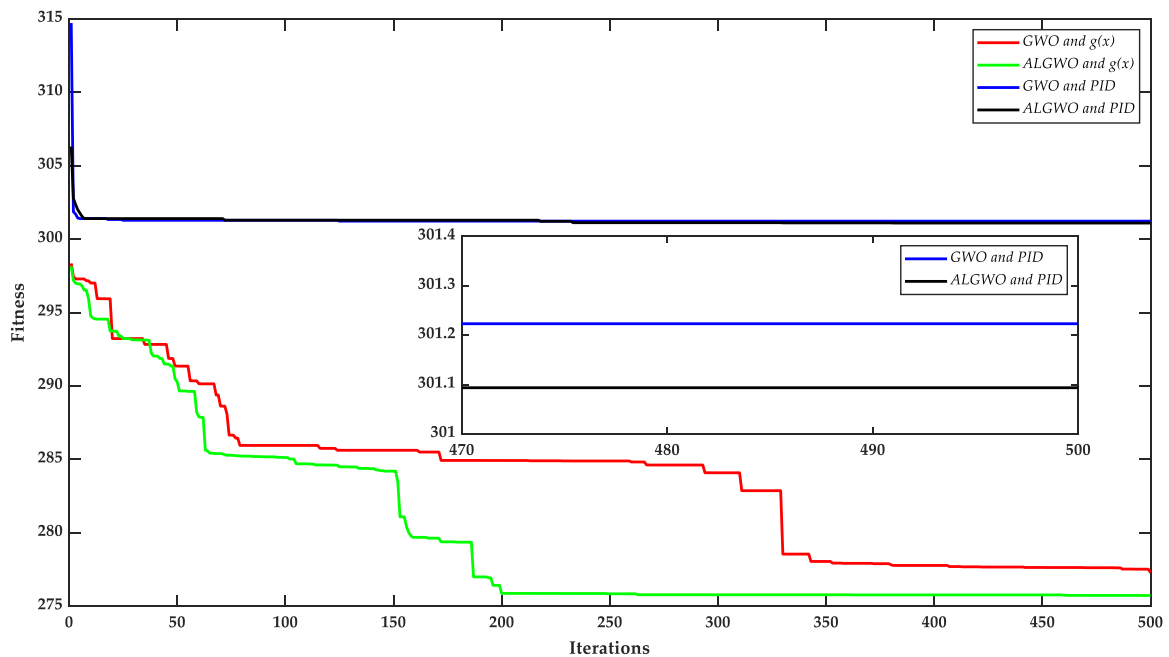


Figure 10. The comparison results of algorithm optimization search on test system 3.

As shown in Figure 9, the controllers employing BPNN-PID outperform the PID controllers in reducing overshoot when the system reference value changes. As shown in Figure 10, the ALGWO algorithm is used to control the BPNN-PID control scheme in comparison to GWO, and it approaches the ideal solution at iteration 200 with a faster convergence rate, demonstrating ALWO’s more deterministic global search capability and quicker discovery speed of high-dimensional optimal solutions. The optimization values utilizing the ALGWO algorithm are 275.7396 and 301.0937, respectively, in terms of optimization accuracy. It is shown that both in high-dimensional and low-dimensional scenarios, it is superior to GWO. According to the convergence curve, the ALGWO algorithm needs 60 iterations to find the local optimal solution and 90 iterations to depart from it. Compared to the 150 iterations needed by the GWO method, this is a huge reduction. As a result, ALGWO can eliminate local optimal solutions more quickly.

4.2.4. Test System 4

For testing, the second-order linear system is used, and formula 23 illustrates the model transfer function. The activation function’s mapping range is set to 0 to 30. The performance characteristics and step response data are displayed in Table 7 and Figure 11.

$$G(s) = \frac{133}{s^2 + 25s} \tag{23}$$

Table 7. Performance characteristics on test system 4.

Algorithm	Overshoot (%)	Rising Time (s)	Settling Time (s)	Fitness
GWO and $g(x)$	0.62	3	3	1.0651
GWO and $t(x)$	0	2	4	1.0234
GWO and $h(x)$	0	2	2	1.0978
GWO and PID	10	2	3	11.9876
ALGWO and $g(x)$	0.02	2	3	1.0360
ALGWO and $t(x)$	0	2	4	1.0238
ALGWO and $h(x)$	0	2	2	1.0804
ALGWO and PID	6	2	4	11.8913

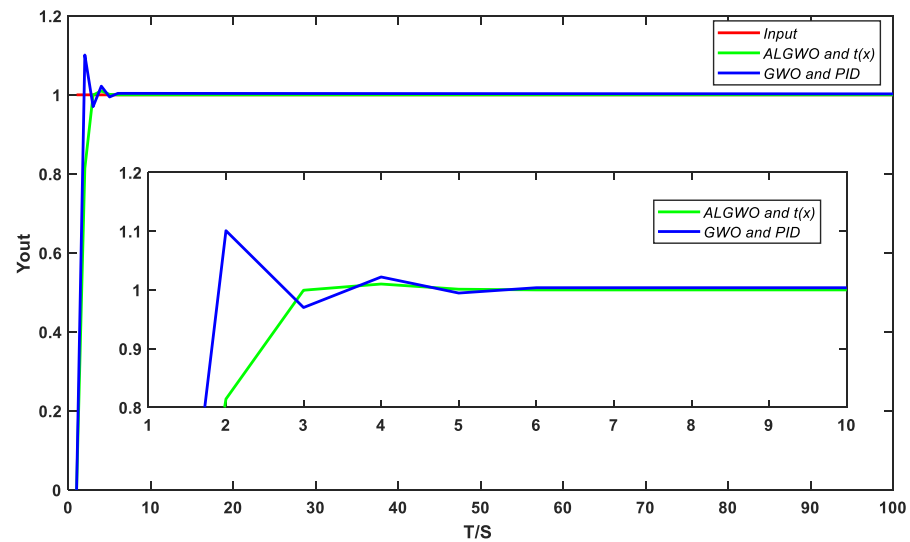


Figure 11. Step response.

Table 7 demonstrates how the ALGWO method has lower fitness than the GWO algorithm in some control schemes. As an illustration, the average precision is decreased in a BPNN PID control system applied to $g(x)$ from 1.0651 to 1.0360, and the overshoot index is also decreased from 0.62% to 0.02%. The average accuracy is therefore increased by the ALWO technique. In terms of whether to use BPNN technology or not, the average accuracy decreased from 11.8913 to 1.0238, which significantly indicates that the ALGWO-BPNN-PID control scheme is more effective in second-order linear systems. Furthermore, the $g(x)$ activation function has better results, as shown in Table 7.

As can be clearly seen from Figure 11, the maximum value of the tracking values decreased from 1.10 to 1.007. As a result, ALGWO-BPNN-PID controllers have a lot of benefits over traditional PID controllers in terms of reducing overshoot.

4.2.5. Test System 5

The first-order plus time delay model is selected for testing, and the model transfer function is shown in (24). The mapping range of the activation function is set to 0 to 30. Table 8 shows the mean value and standard deviation of 10 runs. Table 9 and Figure 12 illustrate the performance characteristics and step response outputs represented by one of the results, while Figure 13 displays the 10-time average convergence characteristics of several algorithms.

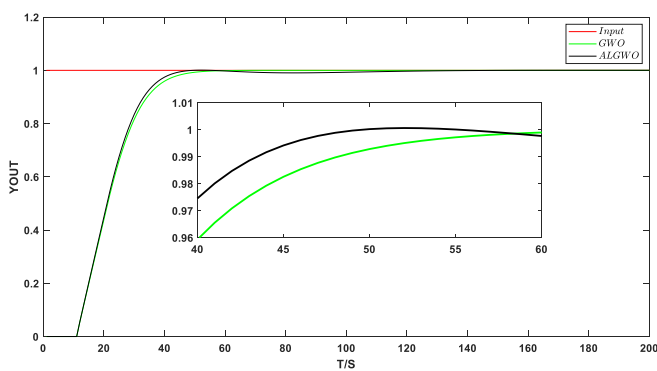
$$G(s) = \frac{1.7e^{-10s}}{320s + 1} \tag{24}$$

Table 8. The comparison of final fitness of algorithms on test system 5.

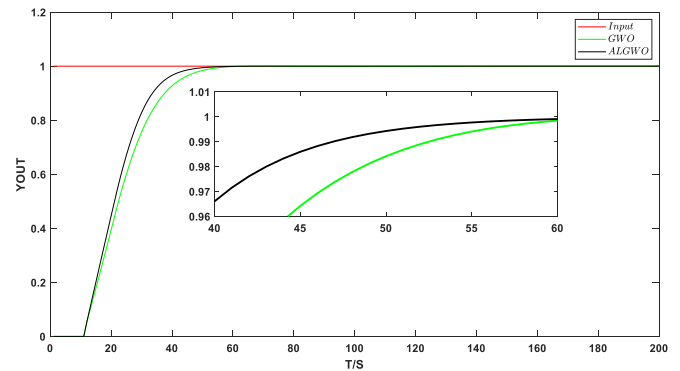
Algorithm		$g(x)$	$t(x)$	$h(x)$	PID
GWO	AVE	24.9222	25.0190	24.8169	27.0366
	STD	0.6144	0.7097	0.6658	0.0773
ALGWO	AVE	24.1989	23.8964	24.8090	27.0068
	STD	0.1029	0.1883	0.6113	0.0245

Table 9. Performance characteristics on test system 5.

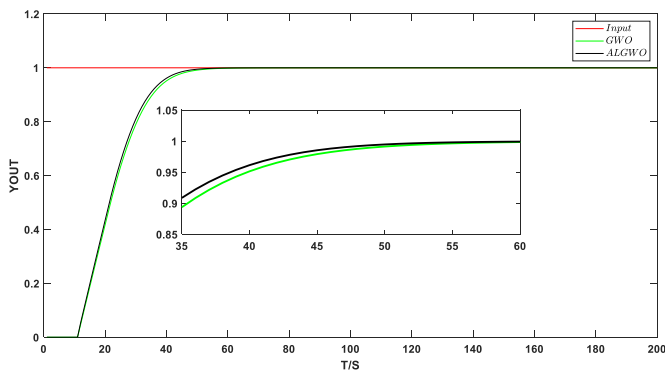
Algorithm	Overshoot (%)	Rising Time (s)	Settling Time (s)	Fitness
GWO and $g(x)$	0.06	34	41	24.2989
GWO and $t(x)$	0	38	49	25.0178
GWO and $h(x)$	0	35	44	24.7274
GWO and PID	0.18	41	52	26.9809
ALGWO and $g(x)$	0	35	45	24.2034
ALGWO and $t(x)$	0	34	44	23.9817
ALGWO and $h(x)$	0	30	36	24.2842
ALGWO and PID	0.11	41	53	26.9800



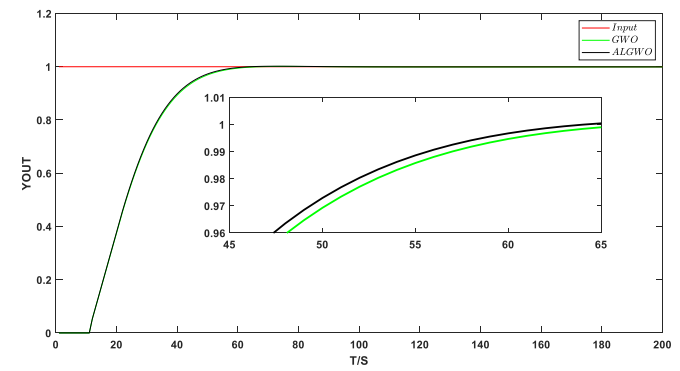
(a)



(b)



(c)



(d)

Figure 12. The step response results on test system 5. (a) The simulation results of BPNN-PID using $g(x)$ and (b) simulation results of BPNN-PID using $t(x)$ and (c) simulation results of BPNN-PID using $h(x)$ and (d) simulation results of traditional PID.

As shown in Table 8, compared to the control scheme using GWO and PID, the control scheme using ALGWO-BPNN-PID reduces the fitness from 27.0366 to 23.8964, improving the average accuracy. Between them, the activation function of $t(x)$ achieves the best effect, with the lowest average fitness of 23.8964. As can be seen from Table 9, using the ALGWO-BPNN-PID control scheme causes, the overshoot to decrease from 0.18% to 0.00%, the rise time to increase by about 5s, and the adjustment time to decrease by about 8s. Between the schemes, the control scheme using the $g(x)$ activation function has a better overall effect.

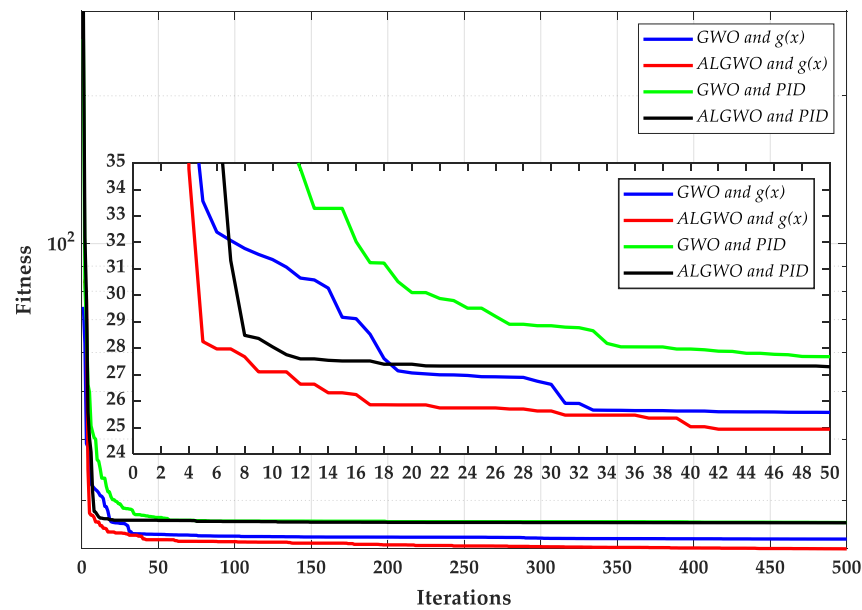


Figure 13. The comparison results of algorithm optimization search on test system 5.

As can be clearly seen in Figure 12, the response speed of the GWO-PID control scheme is slower, reaching 0.98 in about 52 seconds, while the BPNN-PID control scheme is advanced to about 45 seconds. As can be seen from Figure 13, the ALGWO-BPNN-PID control scheme has a faster convergence speed and better optimization accuracy, with a decrease of about two. Therefore, the ALGWO-BPNN-PID control scheme has more advantages in the first-order plus time-delay model.

4.2.6. Test System 6

The time-varying model is selected for testing, and the model’s difference equation is displayed as follows. The mean value and standard deviation for 10 runs are displayed in Table 10. The mapping range of the activation function is set to 0 to 30. Figure 14 shows the step response, while Figure 15 displays the comparison results. The results demonstrate that the ALGWO-BPNN method shows the optimal control effect in slow time-varying systems.

$$w = 1 - 0.8 \times e^{-0.1k} \tag{25}$$

$$y(k) = 0.9969 \times w \times y(k - 1) + 0.0053 \times u(k - 1) \tag{26}$$

Table 10. The comparison of final fitness of algorithms on test system 6.

Algorithm		$g(x)$	$t(x)$	$h(x)$	PID
GWO	AVE	40.8749	46.6955	41.2529	56.4116
	STD	0.61855	8.8126	1.2789	0
ALGWO	AVE	40.2174	41.1174	40.5169	49.2536
	STD	0.5555	0.6698	0.6763	6.6099

First, as shown in Table 10, using the ALGWO method produces lower variances and standard deviations compared to using the GWO algorithm. For example, a PID control system using $t(x)$ BPNN reduced the standard deviation from 8.8126 to 0.6698, and also reduced the average value by 5.5. Therefore, it improves elasticity and average accuracy. Then, the final average fitness is reduced by roughly 10, making the ALGWO-BPNN-PID controllers significantly superior to conventional PID controllers. Finally, the $g(x)$ activation function has a better impact when used as an activation function, with a fitness of 40.2174.

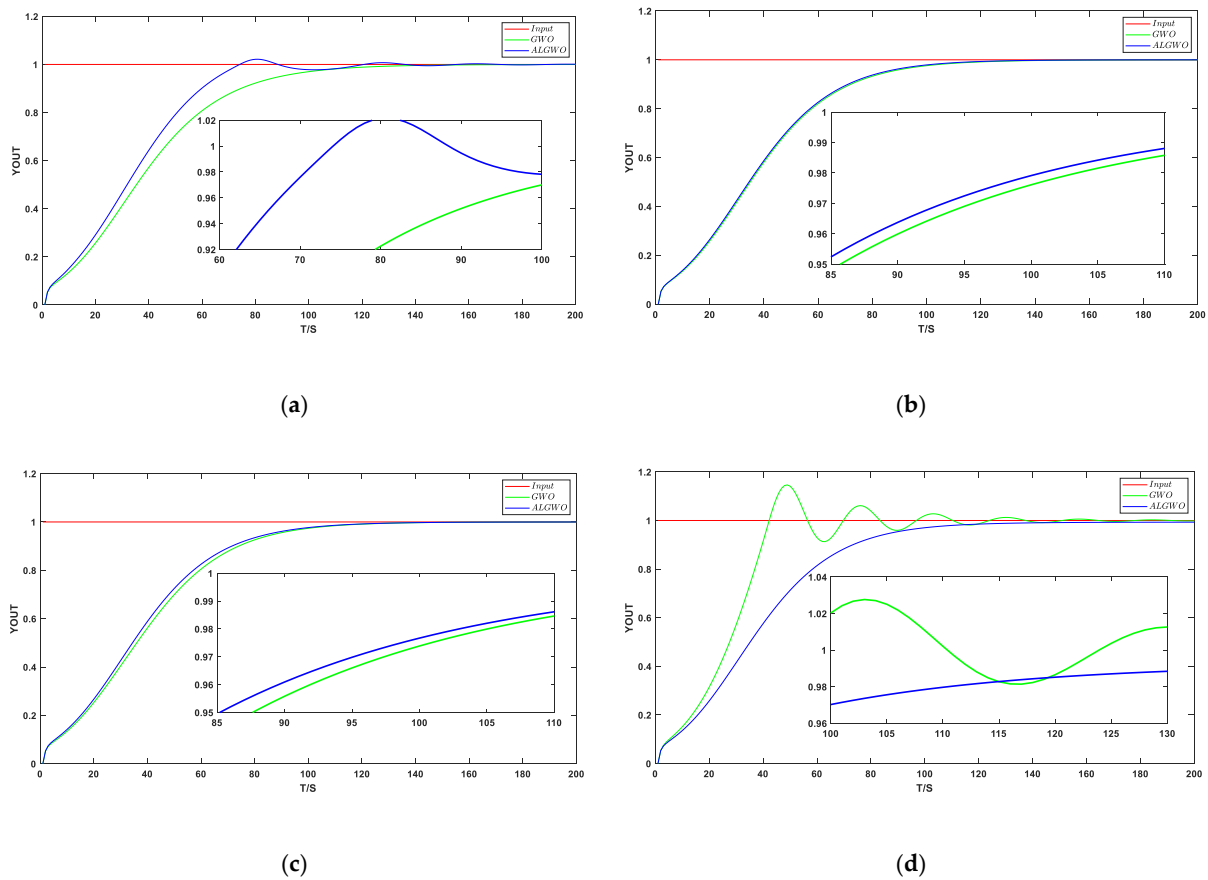


Figure 14. The step response results on test system 6. (a) The simulation results of BPNN-PID using $g(x)$ and (b) simulation results of BPNN-PID using $t(x)$ and (c) simulation results of BPNN-PID using $h(x)$ and (d) simulation results of traditional PID.

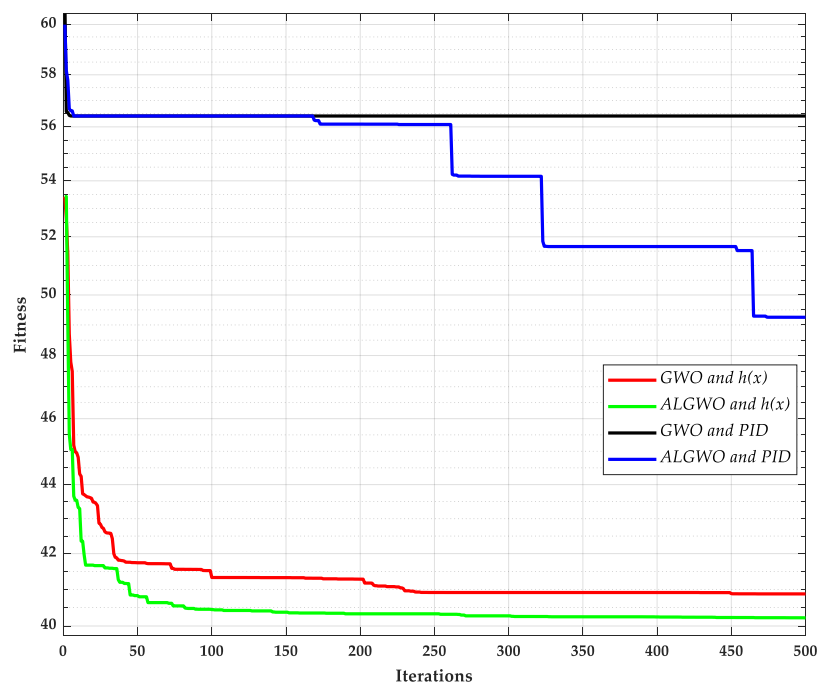


Figure 15. The comparison results of algorithm optimization search on test system 6.

Figure 14 shows that the controller employing BPNN-PID outperforms the PID controller in terms of reaction speed, shortening the time by around ten seconds, when tracking the unit step response. As seen in Figure 15, the ALGWO algorithm is employed to regulate the BPNN-PID control scheme in comparison to GWO, and by iteration 100, it comes close to an optimal solution, demonstrating ALGWO's superior global search capabilities. The optimization values for the ALGWO method are 49.2536 and 40.5169, respectively, in terms of optimization accuracy. This demonstrates that it outperforms GWO. In conclusion, GWO-BPNN-PID controllers have better performance when applied to slow time-varying systems.

5. Conclusions

The most popular type of industrial process control is PID control, which has been utilized extensively in the metallurgical industry, the electromechanical industry, and other industries. However, as the controlled system becomes increasingly complex, the time-varying and time-delayed systems of the PID control algorithm exert only a general control effect. In this paper, an incremental PID control algorithm based on a BP neural network was selected. Furthermore, by utilizing a novel natural excitation optimization technique, namely the gray wolf optimization algorithm based on confrontation learning, we sought to enhance the control impact of the BPNN-PID controller. In the first step, a fresh adversarial search approach for GWO was suggested. Ten test functions were used to evaluate the ALGWO algorithm. This technique performs better than other algorithms because it can increase convergence speed and search capability. The ALGWO-BPNN model is based on the ALGWO algorithm, which was the second step. To achieve real-time online modification of the PID control parameters and acquire the optimal control rules, the key parameters of the BPNN were adjusted using ALGWO. In addition, the impact of changing the activation function on BPNN was discussed and comparative tests were conducted using the conventional PID control. The findings indicate that the ALGWO-BPNN controller exerts a superior control effect compared to the conventional PID controller in time-varying or time-delay systems.

Author Contributions: Conceptualization: H.L.; methodology: H.L.; software: H.L.; validation: Q.Y. and Q.W.; investigation: H.L. and Q.Y.; writing—original draft: H.L.; writing—review and editing: Q.Y. and Q.W.; visualization: H.L.; supervision: Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shandong Provincial Natural Science Foundation of China, grant number ZR2017BF043.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Phu, N.D.; Hung, N.N.; Ahmadian, A.; Senu, N. A new fuzzy PID control system based on fuzzy PID controller and fuzzy control process. *Int. J. Fuzzy Syst.* **2020**, *22*, 2163–2187. [\[CrossRef\]](#)
2. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, *64*, 759–768. [\[CrossRef\]](#)
3. Cohen, G.H.; Coon, G.A. Theoretical consideration of retarded control. *Trans. ASME* **1953**, *75*, 827–833. [\[CrossRef\]](#)
4. Lee, Y.S.; Jang, D.W. Optimization of Neural Network-Based Self-Tuning PID Controllers for Second Order Mechanical Systems. *Appl. Sci.* **2021**, *11*, 8002. [\[CrossRef\]](#)
5. Dogru, O.; Velswamy, K.; Ibrahim, F.; Wu, Y.; Sundaramoorthy, A.S.; Huang, B.; Xu, S.; Nixon, M.; Bell, N. Reinforcement learning approach to autonomous PID tuning. *Comput. Chem. Eng.* **2022**, *161*, 107760. [\[CrossRef\]](#)
6. Sierra-Garcia, J.E.; Santos, M.; Pandit, R. Wind turbine pitch reinforcement learning control improved by PID regulator and learning observer. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104769. [\[CrossRef\]](#)
7. Lee, C.L.; Peng, C.C. Analytic Time Domain Specifications PID Controller Design for a Class of 2nd Order Linear Systems: A Genetic Algorithm Method. *IEEE Access* **2021**, *9*, 99266–99275. [\[CrossRef\]](#)

8. Ozana, S.; Docekal, T. PID Controller Design Based on Global Optimization Technique with Additional Constraints. *J. Electr. Eng.* **2016**, *67*, 160–168. [[CrossRef](#)]
9. Kang, J.; Meng, W.; Abraham, A.; Liu, H. An adaptive pid neural network for complex nonlinear system control. *Neurocomputing* **2014**, *135*, 79–85. [[CrossRef](#)]
10. Wei, J.; Zhang, L.; Li, Z. Design and implementation of neural network pid controller based on fpga. *Autom. Instrum.* **2017**, *10*, 106–108, 113.
11. Hernández-Alvarado, R.; García-Valdovinos, L.G.; Salgado-Jiménez, T.; Gómez-Espinosa, A.; Fonseca-Navarro, F. Neural Network-Based Self-Tuning PID Control for Underwater Vehicles. *Sensors* **2016**, *16*, 1429. [[CrossRef](#)]
12. Bari, S.; Zehra Hamdani, S.S.; Khan, H.U.; Rehman, M.U.; Khan, H. Artificial neural network based self-tuned pid controller for flight control of quadcopter. In Proceedings of the 2019 International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan, 21–22 February 2019; pp. 1–5.
13. Nowaková, J.; Pokorný, M. Double Expert System for Monitoring and Re-adaptation of PID Controllers. In *Innovations in Bio-inspired Computing and Applications*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 85–93.
14. Mitra, P.; Dey, C.; Mudi, R. Fuzzy rule-based set point weighting for fuzzy PID controller. *SN Appl. Sci.* **2021**, *3*, 651. [[CrossRef](#)]
15. Han, S.-Y.; Dong, J.-F.; Zhou, J.; Chen, Y.-H. Adaptive Fuzzy PID Control Strategy for Vehicle Active Suspension Based on Road Evaluation. *Electronics* **2022**, *11*, 921. [[CrossRef](#)]
16. Zhou, H.; Chen, R.; Zhou, S.; Liu, Z. Design and analysis of a drive system for a series manipulator based on orthogonal-fuzzy PID control. *Electronics* **2019**, *8*, 1051. [[CrossRef](#)]
17. Najariyan, M.; Zhao, Y. Granular fuzzy PID controller. *Expert Syst. Appl.* **2021**, *167*, 114182. [[CrossRef](#)]
18. Zhao, X.; Wang, X.; Ma, L.; Zong, G. Fuzzy approximation based asymptotic tracking control for a class of uncertain switched nonlinear systems. *IEEE Trans. Fuzzy Syst.* **2019**, *28*, 632–644. [[CrossRef](#)]
19. Mohammadi Doulabi Fard, S.J.; Jafari, S. Fuzzy Controller Structures Investigation for Future Gas Turbine Aero-Engines. *Int. J. Turbomach. Propuls. Power.* **2021**, *6*, 2. [[CrossRef](#)]
20. Rubio, J.D.J.; Angelov, P.; Pacheco, J. Uniformly Stable Backpropagation Algorithm to Train a Feedforward Neural Network. *IEEE Trans. Neural Netw.* **2011**, *22*, 356–366. [[CrossRef](#)]
21. Kolbusz, J.; Rozycki, P.; Lysenko, O.; Wilamowski, B.M. Error back propagation algorithm with adaptive learning rate. In Proceedings of the 2019 International Conference on Information and Digital Technologies (IDT), Zilina, Slovakia, 25–27 June 2019; pp. 216–222.
22. Milovanović, M.B.; Antić, D.S.; Milojković, M.T.; Nikolić, S.S.; Perić, S.L.; Spasić, M.D. Adaptive PID control based on orthogonal endocrine neural networks. *Neural Netw.* **2016**, *84*, 80–90. [[CrossRef](#)]
23. Hong, H.; Ni, L.; Sun, H. Design and simulation of a self-driving precision compass based on BP+PID control. *Mech. Des.* **2021**, *38*, 78–84.
24. Pei, G.; Yu, M.; Xu, Y.; Ma, C.; Lai, H.; Chen, F.; Lin, H. An Improved PID Controller for the Compliant Constant-Force Actuator Based on BP Neural Network and Smith Predictor. *Appl. Sci.* **2021**, *11*, 2685. [[CrossRef](#)]
25. Wang, Y.; Liu, J.; Li, R.; Suo, X.; Lu, E. Application of PSO-BPNN-PID Controller in Nutrient Solution EC Precise Control System: Applied Research. *Sensors* **2022**, *22*, 5515. [[CrossRef](#)] [[PubMed](#)]
26. Wang, J.; Li, M.; Jiang, W.; Huang, Y.; Lin, R. A Design of FPGA-Based Neural Network PID Controller for Motion Control System. *Sensors* **2022**, *22*, 889. [[CrossRef](#)] [[PubMed](#)]
27. You, D.; Lei, Y.; Liu, S.; Zhang, Y.; Zhang, M. Networked Control System Based on PSO-RBF Neural Network Time-Delay Prediction Model. *Appl. Sci.* **2023**, *13*, 536. [[CrossRef](#)]
28. Ivanov, O.; Neagu, B.-C.; Grigoras, G.; Gavrilas, M. Optimal Capacitor Bank Allocation in Electricity Distribution Networks Using Metaheuristic Algorithms. *Energies* **2019**, *12*, 4239. [[CrossRef](#)]
29. Łysiak, A.; Paszkiel, S. A Method to Obtain Parameters of One-Column Jansen–Rit Model Using Genetic Algorithm and Spectral Characteristics. *Appl. Sci.* **2021**, *11*, 677. [[CrossRef](#)]
30. Bilandžija, D.; Vinko, D.; Barukčić, M. Genetic-Algorithm-Based Optimization of a 3D Transmitting Coil Design with a Homogeneous Magnetic Field Distribution in a WPT System. *Energies* **2022**, *15*, 1381. [[CrossRef](#)]
31. Kashyap, A.K.; Parhi, D.R. Particle swarm optimization aided pid gait controller design for a humanoid robot. *ISA Trans.* **2021**, *114*, 306–330. [[CrossRef](#)]
32. Feleke, S.; Satish, R.; Pydi, B.; Anteneh, D.; Abdelaziz, A.Y.; El-Shahat, A. Damping of Frequency and Power System Oscillations with DFIG Wind Turbine and DE Optimization. *Sustainability* **2023**, *15*, 4751. [[CrossRef](#)]
33. Wang, W.; Tang, R.; Li, C.; Liu, P.; Luo, L. A BP neural network model optimized by mind evolutionary algorithm for predicting the ocean wave heights. *Ocean Eng.* **2018**, *162*, 98–107. [[CrossRef](#)]
34. Zhang, Y.; Tang, J.; Liao, R.; Zhang, M.; Zhang, Y.; Wang, X.; Su, Z. Application of an enhanced BP neural network model with water cycle algorithm on landslide prediction. *Stoch. Environ. Res. Risk Assess.* **2021**, *35*, 1273–1291. [[CrossRef](#)]
35. Wang, Q.; Xi, H.; Deng, F.; Cheng, M.; Buja, G. Design and analysis of genetic algorithm and BP neural network based PID control for boost converter applied in renewable power generations. *IET Renew. Power Gener.* **2022**, *16*, 1336–1344. [[CrossRef](#)]
36. Curry, H.B. The method of steepest descent for non-linear minimization problems. *Q. Appl. Math.* **1944**, *2*, 258–261. [[CrossRef](#)]

37. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
38. Dhargupta, S.; Ghosh, M.; Seyedali, M.; Ram, S. Selective Opposition based Grey Wolf Optimization. *Expert Syst. Appl.* **2020**, *151*, 113389. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.