Research Article

The Application and Research of the GA-BP Neural Network Algorithm in the MBR Membrane Fouling

Chunqing Li,[1] Zixiang Yang,[1] Hongying Yan,[1] and Tao Wang[2]

[1] *School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China*
[2] *School of Environmental and Chemical Engineering, Tianjin Polytechnic University, Tianjin 300387, China*

Correspondence should be addressed to Chunqing Li; franklcq@163.com

It is one of the important issues in the field of today's sewage treatment of researching the MBR membrane flux prediction for membrane fouling. Firstly this paper used the principal component analysis method to achieve dimensionality and correlation of input variables and obtained the three major factors affecting membrane fouling most obvious: MLSS, total resistance, and operating pressure. Then it used the BP neural network to establish the system model of the MBR intelligent simulation, the relationship between three parameters, and membrane flux characterization of the degree of membrane fouling, because the BP neural network has slow training speed, is sensitive to the initial weights and the threshold, is easy to fall into local minimum points, and so on. So this paper used genetic algorithm to optimize the initial weights and the threshold of BP neural network and established the membrane fouling prediction model based on GA-BP network. As this research had shown, under the same conditions, the BP network model optimized by GA of MBR membrane fouling is better than that not optimized for prediction effect of membrane flux. It demonstrates that the GA-BP network model of MBR membrane fouling is more suitable for simulation of MBR membrane fouling process, comparing with the BP network.

1. Introduction

As a new wastewater treatment technology, the membrane bioreactor has attracted much concern by its high quality of the effluent water quality and was considered as a water-saving technology with better economic, social, and environmental benefits. But it has been proved by practice that membrane fouling is a major bottleneck restricting its development. Then, by the study of the mechanism of membrane fouling, it is imminent to slow the membrane fouling rate and reduce the membrane pollution.

With the emergence of the computer simulation technology, it greatly reduces the time, space, and cost of the MBR experiment. Therefore, the MBR computer simulation technology has become a powerful tool for the research of the MBR, and its development will have a positive reference and guiding role for the practical engineering applications of the MBR.

Based on this idea, according to the real data of the experiment and industrial production of a MBR sewage treatment plant in Shijiazhuang, this paper focused on membrane fouling problem in the process of membrane bioreactor sewage wastewater. And it established intelligent prediction model with the correlation and the relationship between operation parameters and flux, by means of smart algorithm. Through the constant optimization of the algorithms and models, it achieved more accurate prediction of unknown membrane fluxes and thus predicted the degree of the membrane fouling. And seeking the optimum operating conditions of controlling membrane pollution trend, it improved membrane fouling problem of the MBR process. The research of this topic had a certain reference value for membrane fouling prediction, parameters selection, and operation of the MBR practical engineering. It also had a positive guiding significance for stable operation and further application of the membrane bioreactor in the field of wastewater treatment

and thus further expressed technological advantages of this process.

2. Prediction Model of MBR Membrane Fouling Based on BP Algorithm

2.1. Intelligent BP Simulation Model of MBR Membrane Fouling. Membrane flux is an important indicator of the degree of membrane fouling, so this paper used intelligent simulation of MBR membrane flux to feed back membrane fouling. And membrane fouling index can be expressed as

$$f = \left(1 - \frac{J_v}{J_0}\right) \times 100\%. \tag{1}$$

According to filtering model based on Darcy's law, the setting of the initial membrane flux J_0 is

$$J_v = \frac{\Delta p}{\mu \left(R_m + R_g + R_c\right)}, \tag{2}$$

where Δp is pressure difference on both sides of membranes (Pa); μ is filtrate viscosity (Pa.s); R_m is resistance of the membrane (1/m); R_g is resistance of membrane fouling (1/m); R_c is resistance of cake (1/m).

It is shown that R_c and R_g are all 0 and MLSS is also 0 in the initial state. When the initial viscosity value is μ_0, we can obtain the equation of initial membrane flux as follows:

$$J_0 = \frac{\Delta p}{\mu_0 R_m}. \tag{3}$$

Finally, the setting of the membrane flux is determined by many factors, which is what this paper discussed and researched questions.

Mathematical simulation models are built on the basis of certain assumptions. This paper assumed that the entire membrane bioreactor system had been running in a stable state, activated sludge was completely trapped in the MBR, and materials in the MBR were fully mixed.

On this basis, for the historical data, a total of 80 groups, industrial production, and test of a MBR sewage treatment plant in Shijiazhuang city, after treating them by collecting, statistical analysis, and principal component analysis, we determine the MLSS of inlet, operating pressure, and the total resistance as three main factors affecting the MBR membrane flux. It is shown in Table 1 that the sample data and its format were finished. And for four columns of this table, we used MATLAB programming software to determine gradually the optimal number of neurons in hidden layer of neural network as well as simulate and predict eventually MBR membrane fouling. After determining the input and output vectors, we processed them normally.

In the experiment, we recorded that the BP network needs the evolution generations to meet preset accuracy requirements 3% in the different numbers of units in hidden layer. And after the evolution of each network was completed, we input test data to verify its accuracy and calculated the respective prediction error of average. After the network structure was determined, we input training data for network training, saved network structure after training, and input forecasting data validation. And we analyzed the prediction error between the actual output data and the desired one, thus evaluating the performance of BP network prediction of the MBR membrane flux.

2.2. The Simulation and Results. The setting of BP algorithm parameters was as follows: the rate of the initial learning was 0.08; the initial space of the weights was $B = (-1, 1)$; the transfer function of input layer to hidden layer was logsig and hidden layer to output layer was purelin; the training function was trainlm; the index of error was set as $2e - 4$. We selected randomly 74 groups from experimental samples as training samples to train the BP neural network. And we selected the remaining 6 groups as test samples to analyze the degree of fitting between the measured values and the predicted values of neural network model, so as to test the effect of learning.

By the simulation of training samples and testing samples, we get the following chart (Figures 1 and 2 and Table 2).

As shown in Figure 1, network error of the model stabilized through the 4254 iterations, which indicated that the convergence speed of BP network is slow. And the error was bigger in a few iterations. But at the end of the training, mean square error of network was 0.000199978 and reached the value 0.0002 of predetermined set. From the histogram of the prediction results in Figure 2, we can see that the network predicting and measured values were basically consistent. It suggested that the BP network model of MBR membrane fouling was more successful and can realize basically the prediction of the MBR membrane flux.

However, it is shown from the relative prediction error values in Table 2 that the prediction maximum error is 16.27% and it is relatively large. It explains that the forecast of the BP neural network is not enough accurate for individual points and the average relative error is 7.40%, which needs to improve.

2.3. Defects and Improvement of BP Neural Network

2.3.1. Defects. In the artificial neural network, although BP neural network is the most widely used and has many remarkable advantages, some limitations still exist [1, 2]. They are mainly the following aspects:

(1) slow convergence speed in the learning process;

(2) easy to fall into local minima;

(3) difficult to determine network structure;

(4) lack of effective selection method about the learning step.

So, in order to make the BP neural network play a greater role in the field of forecasting, we need to make appropriate improvements in the actual modeling.

TABLE 1: Data of membrane flux, pressure, and so forth collected.

Time (h)	Temperature (°C)	Pressure (MPa)	MLSS of inflow (mg/L)	Total resistance ($\times 10^{12}$ m^{-1})	Fluxes (L/m^2 h)
1	24	0.016	343.38	2.05	46.4
2	24	0.0168	378.15	2.403	45.5
3	24	0.0175	392.42	2.989	45.3
4	24	0.0243	472.43	3.796	42.2
5	24	0.0268	483.73	3.957	45.1
6	24	0.0291	583.16	4.41	42.2
7	24	0.0325	556.43	5.119	39.7
8	24	0.0362	503.56	6.075	37.3
9	24	0.0351	591.41	7.143	31.4
10	24	0.0385	561.85	8.421	28.9
11	24	0.0269	612.42	7.623	21.7
12	24	0.0226	655.47	9.737	14.5
13	24	0.0198	712.43	11.659	11.2
14	24	0.0193	615.12	12.911	10.5
15	24	0.0187	715.89	12.54	9.4
16	24	0.0157	331.28	1.793	51.8
17	24	0.0164	380.15	2.4175	46.5
18	24	0.0197	391.41	3.197	42.3
19	24	0.0239	442.44	3.673	41.2
20	24	0.0264	493.72	3.961	43.4
21	24	0.0283	587.23	4.438	39.2
22	24	0.0319	568.43	5.233	37.8
23	24	0.0355	583.56	6.019	36.7
24	24	0.0369	585.41	7.013	32.5
25	24	0.0384	571.67	8.025	28.9
26	24	0.0268	601.32	9.043	21.6
27	24	0.0234	654.15	9.941	15.9
28	24	0.0196	698.23	10.892	11.3
29	24	0.0191	695.72	12.136	10.1
30	24	0.0185	709.46	12.759	8.4

TABLE 2: Relative errors of membrane flux prediction of BP network.

Desired values (L/m^2 h)	11.2000	45.3000	42.3000	9.4000	21.7000	28.9000
Actual values (L/m^2 h)	10.2366	47.8755	41.6611	8.9128	23.2551	24.1967
Relative error	0.0860	0.0569	0.0151	0.0518	0.0717	0.1627
Average relative error			0.0740			

2.3.2. Improvement. In response to these problems, researchers have proposed many effective improvements; for example, consider the following.

(1) Momentum Additional Method. Momentum additional method is based on the method of reverse propagation and at the same time takes into account the effect of the error in spatial gradient and errors surface. And changes of the new weight are, plus the value proportional to last-connection weight, on the basis of the change of each of connection weights. When being without additional momentum, the network may fall into the local minimum of shallow, and it can slip the minimum in the role of additional momentum.

(2) Adaptive Learning Rate. The learning rate is usually obtained by experience or experiment, but sometimes even the effect of the learning rate is better in initial training, which can also not guarantee appropriate training in the later. So it needs to correct adaptively learning rate. Its criteria are as follows. Check whether it really reduces the error function after the weight is corrected. If it is, then explain that the learning rate selected is small and can be increased a quantity and if not, then the value of learning rate should be reduced.

(3) Optimize Initial Weights and the Threshold of the Network. The connection weights, between neurons of each layer of the neural network, contain all the knowledge of the neural
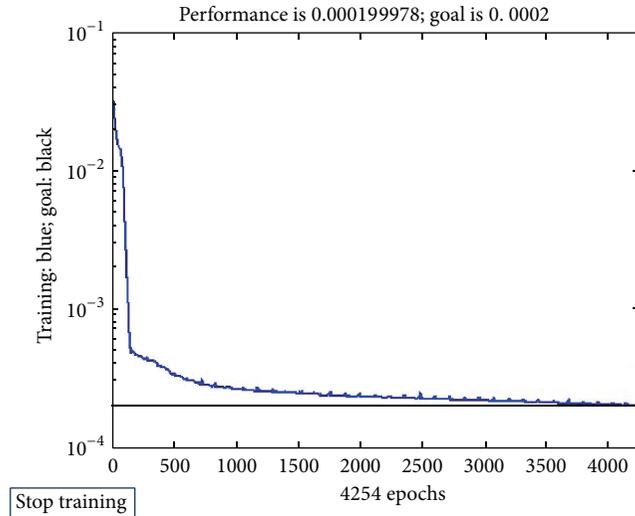
FIGURE 1: The error curve of membrane flux in BP network training process.

FIGURE 2: Measured values of membrane flux and predicted values of BP network.

network system. As the BP network is based on gradient descent technique, it may produce an entirely different result for initial different values given randomly in traditional. Once the value is improper, it will cause oscillation or inconvenience of the network, and it is also easy to fall into local extremism in complex situations. The genetic algorithm has independence of the initial value, global search optimization ability, faster convergence rate, and so forth. So we can use genetic algorithm to search properly initial weights of BP neural network for optimizing network and then it is assigned to BP network for training.

3. Optimize Parameters of BP Model of the MBR Membrane Fouling by Genetic Algorithm

Known by the last section, the BP neural network can achieve the detection of the MBR membrane flux, but the accuracy and speed need to further improve. BP algorithm only adjusted connection weights of neural network from the local angle but did not examine the whole learning process form the global perspective. So it is easy to converge to local minimum. The first two methods for improving BP algorithm, proposed in the last section, are essentially to adjust network weights in the local area and cannot overcome the local searching feature of BP algorithm.

To avoid this, we need to consider it from the global angle. One of the ways to improve them is using the third method proposed in the last section. It is introducing the searching method of global intelligent optimization in the learning process of the neural network genetic algorithm. So this section considered adopting genetic algorithm to search for the initial weights and the threshold suitable to BP neural network, so as to realize the optimization of the network and achieve the purpose improving MBR membrane fouling prediction.
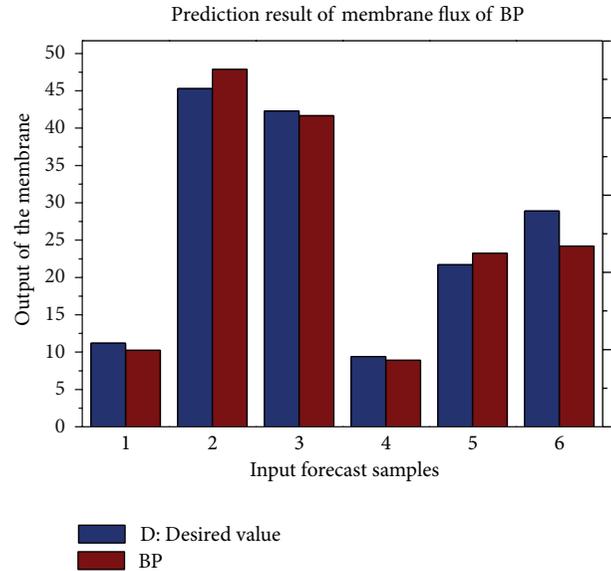
3.1. Genetic Algorithm. Genetic algorithm (GA) is a kind of random searching algorithm, referencing natural selection, and genetic mechanism of biological evolution. The algorithm is a new class of general optimization algorithm, for solving the optimization problem of general nonlinear searching. It is not required for linear, continuous, and micro of the model and also less affected by the number of variables and constraints. And it can be adaptive to learn and accumulate the information of search space, so as to find the optimal solution. GA has been applied for function optimization, combinatorial optimization, automatic control, robotics, parameter optimization of neural network, structure design, image processing systems, and other fields.

Compared with other optimization algorithms, GA optimization is a kind of robust and global searching algorithm, and it mainly has the following features:

(1) adopting the encoding of the decision variables as operands;

(2) directly taking values of the objective function as searching information;

(3) using multiple searching points to search for information at the same time;

(4) using probabilistic search technology.

3.2. Optimization of BP Network Parameters by GA

3.2.1. Description of the Algorithm. BP algorithm is a kind of local searching method based on gradient descent. After the initial weight vector is determined, its searching direction is also determined. So the algorithm can only start from the initial value and gradually goes for the direction, until it meets the stopping criteria of algorithm. Its advantage is the better ability of local optimization, but the drawback

is easy to fall into local minimum and slow convergence rate. This is because the initial weight vector of the BP network is selected randomly, which makes the direction of the process optimization without certainty. So the effect of the solution is sometimes good and sometimes bad. As a kind of probabilistic optimization algorithm based on the law of biological evolution, genetic algorithm has a strong adaptive searching and global optimization ability and can avoid local minimum points with great probability. Meanwhile, it does not need gradient information for solving problems, and its operation is more convenient. This shows that the genetic algorithm is suitable to be used to enhance the learning ability of BP algorithm. Therefore, we introduced GA ideas in the theory of BP network. The two can well complement each other and achieve the purpose of global optimization fast and efficiently. In fact, the combination of genetic algorithm and neural network has become one of the focuses of the research and application of artificial intelligence algorithms [3]. With making full use of the advantage of both, the new algorithm has robustness and self-learning ability of neural network and also has the global searching ability of genetic algorithm.

Optimizing weights and the threshold of the BP neural network by GA is divided into two parts. The first part used genetic algorithm embedding neural network and searched the best individual in the probable scope of the weights of BP network. The second part continued to use BP algorithm. On the basis of GA optimization, we used the optimal individuals searched by GA as initial weights and the threshold of BP network and then directly used BP algorithm to train the network. So this section mainly described achieved method, under the premise of fixed structure of BP network; that is, we first used GA to optimize initially connected weights and the threshold of BP network and then used BP algorithm to search accurately the optimal solution of the weight and the threshold.

3.2.2. Steps of Implementation. GA optimizes weights and the threshold of neural network of BP algorithm, whose steps of implementation are as follows.

(1) Encoding and Initialization. Using the genetic algorithm to optimize the neural network, its main focus is to optimize connection weights and threshold between neurons in the neural network. Initial populations are whose connection weights and thresholds of neural networks are initialized. Because connection weights and the threshold of the network are real, so each of connection weights and the threshold directly is expressed by real number. This encoding is more intuitive comparing with binary and whose accuracy is also higher than that. We set the size of the group as N, and each individual of the population contained S genes (connection weights): $S = R * S_1 + S_1 * S_2 + S_1 + S_2$, where R, S_1, S_2 represented the number of neurons in the input layer, the number of neurons in the hidden layer, and the number of neurons in the output layer. The scope of each real was set as $(-1, 1)$, and we get the initial population $P(t)$.

(2) Calculation of Fitness. Calculate fitness value of all individuals of the population. We assigned S weights and thresholds

of the initial group to the BP neural network, for forward propagation of the input signal, and calculated the sum of squared errors between the output value and the desired of the network. So the fitness function can be measured by the reciprocal of sum of squared errors [4] as follows:

$$\text{fitness (sol)} = \frac{1}{\sum_{i=1}^{n_0} (o_i - y_i)^2}, \tag{4}$$

where sol means any individual of the population, 0_i means the actual output of the ith output neuron, y_i means the desired output of the ith output neuron, and n_0 means the number of the output neurons. The smaller the difference between the actual output and the target output was, the bigger the value of fitness function was and also the more the desired requirement was.

(3) Selection Sort. After calculating the fitness of each individual in the population, we sorted all individuals in the population according to their respective fitness. According to this order, it determined the probability of each individual selected, that is, proportional selection method, as follows:

$$\text{fitness (sol)} = \frac{f_i}{\sum_{i=1}^{M} f_i}. \tag{5}$$

It ensured that the bigger the fitness value of individual is, the greater it is selected, and the lower the fitness value of individual is, the lower it is selected. But there is a small fitness value of the individual that may also be selected. Therefore, while choosing, we joined the best choice strategy and retained the optimal individual of each generation to the offspring.

(4) Genetic Operators

(a) Crossover Operator. Cross is the most important means getting the new best individual. This paper used the arithmetic crossover, which produced two new individuals by a linear combination of two individuals. We set two individuals X_A^i and X_B^i that crossed each other; then the new individuals produced are

$$\begin{aligned} X_A^{i+1} &= \alpha X_B^i + (1 - \alpha) X_A^i, \\ X_B^{i+1} &= \alpha X_A^i + (1 - \alpha) X_B^i, \end{aligned} \tag{6}$$

where $\alpha \leq P_c$ and $\alpha(0 < \alpha < 1)$ is a random number generated by the parent individual of each pair and P_c is a crossover rate of the parent individual of each pair.

(b) Mutation Operator. Use uniform mutation. We set mutation probability as P_m and set mutation point as x_k whose range is $[U_{\min}^k, U_{\max}^k]$. Then new genetic value of mutation point is $x_k' = U_{\min}^k + r(U_{\max}^k - U_{\min}^k)$, where r is a random number in the range $[0, 1]$.

(5) Generate New Populations. We inserted the new individual into the population $P(t)$, which generated a new population $P(t+1)$. Then we gave connection weights of the individual of

new population to neural network and computed the fitness function of the new individual. If it reached the predetermined value, then we entered the next step. Otherwise, we continued their genetic operations.

(6) Assign Values to the Initial Weights of BP Network. We decoded the optimal individual searched by GA and assigned it to the initial connection weight corresponding with BP network. Then we continued to use BP algorithm to train network, until the error sum of squares achieved the specified accuracy $E_{bp}(E_{bp} < E_{ga})$ or achieved the maximum number of iterations set, and the algorithm ends.

The first part of the genetic BP algorithm was that we embedded neural network by genetic algorithm and searched out the optimal individual within the approximate range of the neural network weights. If the error sum of squares achieved E_{ga}, we entered the next step without requiring to reach the highest accuracy. The second part continued to adopt BP algorithm to train the network. On the basis of optimization by genetic algorithm, we set the optimal individual searched by GA as initial weights of neural network and then directly used BP algorithm until the error sum of squares achieved the minimum error of the problem. Theoretically, genetic algorithm has the global searching ability and can quickly determine the area of global optimal solution. But its local searching ability is weak and cannot search the optimal solution quickly. So, after the genetic algorithm searched the approximate scope of the global optimal solution, we gave full play to local searching of BP algorithm. With two parts combining organically, it can better accelerate operation efficiency and increase the learning ability of neural network.

3.3. Model of GA-BP Neural Network of the MBR Membrane Fouling Prediction

3.3.1. The Design of the GA-BP Model of the MBR Membrane Fouling

(1) Set BP Network Parameters. BP network structure had been determined by Section 2.2. In order to compare with BP neural network, we used the same parameters setting with the second section. The rate of the initial learning was 0.08; the transfer function of input layer to hidden layer was logsig, and hidden layer to output layer was purelin; the training function was trainlm; the index of error was set as $2e - 4$.

(2) Optimize Weights and the Threshold of BP Network by the GA. Because the weight can be expressed by real number, sp, we used the real coding method. In addition, because the variable itself is real number, we can directly calculate the target value and fitness value without decoding, which can accelerate the searching speed [4] and be easy to combine with BP algorithm. Each connection weight is directly expressed by a real number; the bit string obtained by encoding all weights and thresholds was called an individual sol = $[s_1, s_2, \ldots, s_L]$. Each weight or threshold corresponded to a bit of the individual, which was called a gene. We get all weights and thresholds together with the order W_1, B, W_2, S

which formed chromosome form of the real-coded GA, and its length was $L = n_i \times n + n + n \times n_0 + n_0$. In this paper, there are 3 input units, 7 hidden units, and one output unit. And adding the threshold of hidden layer and output layer units, each individual consists of $3 \times 7 + 7 + 7 \times 1 + 1 = 36$, and the range of each real number is $(-1, 1)$.

Genetic algorithm basically does not use external information in the process of the evolutionary search and is only based on the fitness function. So it is crucial for choosing fitness function, which directly affects the convergence rate of genetic algorithm and the ability to find the optimal solution. Since the objective function is error sum of squares of the neural network, and for getting its minimum value, so the fitness function adapts the inverse of error function.

The selecting operation used fitness proportion, which is the most common way, the cross used the method of arithmetic crossover, and variation method used the operation of nonuniform mutation. It was implemented with MATLAB statements as follows:

$$[x, endPop, bPop, trace]$$
$$= ga(aa,'gabpEval', [], initPpp, [1e - 611],'$$
$$maxGenTerm', gen,'normGeomSelect', [0.09],$$
$$['arithXover'], [2],'nonUnifMutation'[2gen3]).$$

If the value of the fitness function, the optimal individual corresponding to, met accuracy requirements or reached iterations ($T = 100$) that we set, or the range of the average fitness value changed lastly that was less than a certain constant and over a certain algebra, then we finished the training. At this time, we set the individual that had the largest fitness as output of the optimal solution, and it was the initial weight and threshold that need to be optimized. Otherwise, we continued the cycle. After reordering the current parent and offspring, we chose N individuals that had higher fitness value as the next generation of the group. And we computed its fitness and trained them again until satisfying the termination conditions of the above.

(3) Train the BP Network. We decoded the optimal solution which was obtained from genetic algorithm and we assigned them to BP network without starting training as initial weights of BP network. Then, according to BP algorithm, we input training samples for training and learning of the network and calculated the error between the output value and the expected value. If it was beyond the precision requirement, then we turned to the backpropagation process and returned the error signal along the way. At the same time, according to the size of the error of each layer, we adjusted its weights and thresholds layer and layer. Until the error E was less than given value $2e - 4$, or it reached the training number predetermined, then we ended BP algorithm. We saved the weight and the threshold of hidden layer that had been trained as the new initial weights and the threshold of the network and then input the test sample for simulation.

In summary, that used MATLAB7.10.0 (R2010) with installing the genetic algorithm toolbox gaot to compile file M, which achieved the optimization of weights and thresholds; the core code is as follows.

TABLE 3: Relative error of the prediction of GA-BP network.

Desired values (L/m^2 h)	39.2000	11.3000	43.4000	41.2000	46.5000	11.2000
Actual values (L/m^2 h)	36.7028	10.7743	44.6790	41.5090	45.3245	10.9101
Relative error	0.0637	0.0465	0.0295	0.0075	0.0253	0.0259
Average relative error			0.0331			

% Train the weight and the threshold of BP by GA.

$[P, T, R, S1, S2, S]$ = nninit; % P and T are a pair of training samples for the input and output of network. $R, S1,$ and $S2$ are, respectively, the input of network, hidden layer, and the number of output neurons. S is the encoding length of genetic algorithm. Consider the following:

$aa = ones(S, 1) * [-1, 1];$

popu = 50; % The scope of the group.

$initPop = initializega(popu, aa, 'gabpEval');$ %Initial population. Among, gabpEval is target function, which also is called fitness function.

$gen = 100;$ %The number of genetic.

$[x, endPop, bPop, trace]$

$= ga(aa, 'gabpEval', [], initPpp, [1e-611],'$

$maxGenTerm', gen, \ldots 'normGeomSelect',$

$[0.09], ['arithXover'], [2],'$

$nonUnifMutation'[2gen3]);$

% Decode the weight and the threshold form code X, and assign them to BP network without starting training.

$[W1, B1, W2, B2, P, T, A1, A2, SE, val]$ = $gadecod(x);$ % gadecod is decoding function.

$net.IW1, 1 = W1;$

$net.LW2, 1 = W2;$

$net.b1 = B1;$

$net.b2 = B2;$

% Use the new weight and thresholds for training.

$net\& = newff(inputXX, outputYY, 7,'$

$tansig', 'purelin', 'trainlm');$

$[net, tr] = train(net, input\ XX, output\ YY);$ %Train network.

% Simulation test

$GABP = sim(net, inputn_test).$

3.4. Experimental and Simulation Results. To illustrate the performance of genetic algorithm for designing and optimizing weights and the threshold of neural networks, this paper used the same sample data. That is, we randomly selected 74 groups as the training sample and the rest were selected as test data to observe the training and testing performance of MBR membrane flux by the GA-BP network.

In the experiment, the error sum of square and the fitness curve of genetic algorithm are shown in Figure 3. The figure shows that, when the genetic algorithm converges, error sum of squares and fitness change with the increase of genetic generations. As it is shown, on the whole, the error is reducing and the fitness is increasing.

In the experiment, the BP network, optimized by genetic algorithm, only needs to go through 106 iterations to achieve the same predetermined error, which significantly accelerates the training speed of the network.

To detect the remaining 6 groups of data, the experimental result is shown in Figure 4. Obviously, the GA-BP network also realized the reasonable prediction about membrane flux. And we can see from Table 3 that the biggest relative error is reduced to 6.37% and the average relative error is reduced to 3.31%. Obviously, it had a prediction effect with more accuracy comparing to BP network with optimization. And it realized more reasonable forecast for the process of the MBR membrane fouling. It is visible that, after optimization, the self-learning ability of GA-BP neural network was stronger than that of the BP neural network without optimization. And it had relatively higher accuracy of prediction, better generalization and nonlinear mapping, and better network performance.

4. Conclusion and Outlook

For MBR membrane fouling factor, the traditional mathematical model of MBR membrane fouling has some defects in different degrees, which cannot accurately explain the phenomenon of membrane fouling; this paper introduced some intelligent algorithms by consulting relevant references, such as neural network and genetic algorithm, to build the model of MBR membrane fouling and implement relevant optimization. First, we established the system model of MBR intelligent simulation based on BP neural network, about the relationship between the membrane fouling factors and membrane fouling that expresses the degree of membrane flux. The experiment found that it can be basically consistent between predicted value and desired value and only the error of a few points was larger. It indicated that the prediction model of BP neural network for membrane fouling was more successful. Referencing the characteristics of genetic algorithm, followed by optimizing the weight and the threshold of the BP neural network by using genetic algorithm, this paper established the prediction model of MBR membrane fouling that was based on GA-BP neural network. It was shown from experimental study that GA-BP network optimized was better than BP network model nonoptimized on the rate of convergence and prediction precision. It shows that,

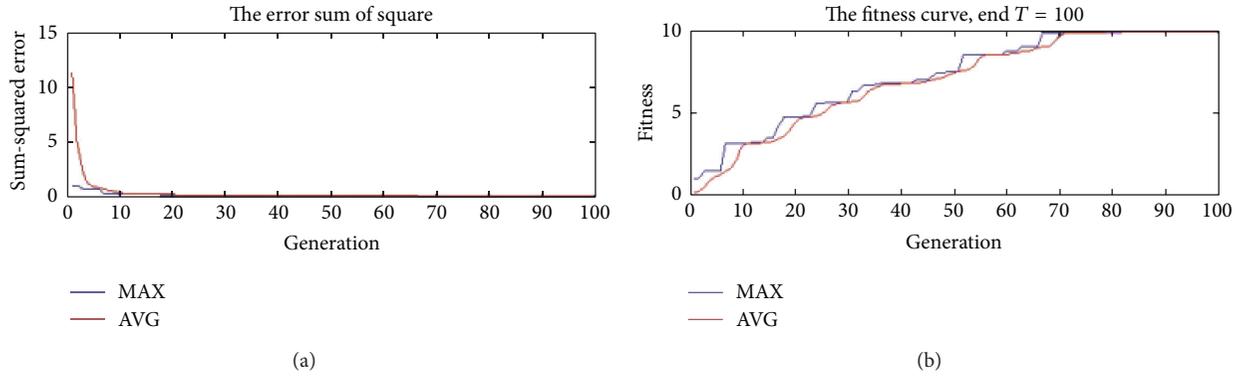(a)   (b)

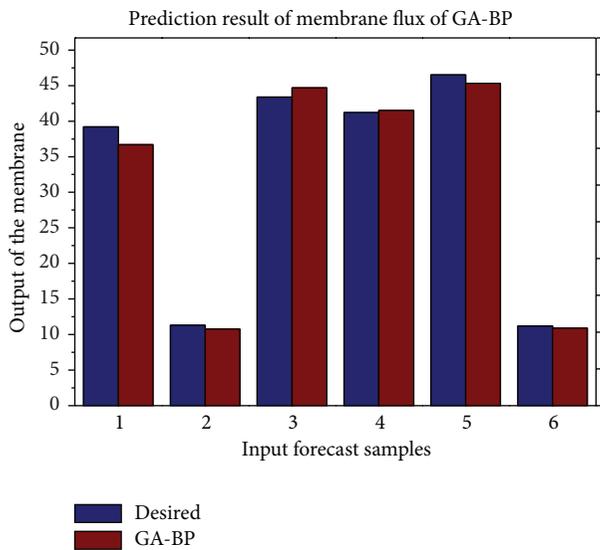FIGURE 3: The error sum of square and the fitness curve.

FIGURE 4: Predicted and measured values of the GA-BP membrane flux.

comparing to the relatively simple BP network, the GA-BP network was more suitable for the prediction of MBR membrane fouling.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] H.-M. Ke and C.-Z. Chen, "Study on BP neural network classification with optimization of genetic algorithm for remote sensing imagery," *Journal of Southwest University*, vol. 32, no. 7, article 128, 2010.

[2] Y.-S. Xu and J.-H. Gu, "Handwritten character recognition based on improved BP neural network," *Communications Technology*, vol. 5, no. 44, article 107, 2011.

[3] A. A. Javadi, R. Farmani, and T. P. Tan, "A hybrid intelligent genetic algorithm," *Advanced Engineering Informatics*, vol. 19, no. 4, pp. 255–262, 2005.

[4] X.-F. Meng, H.-R. Lu, and L. Guo, "Approximately duplicate record detectionmethod based on neural network and genetic algorithm," *Computer Engineering and Design*, vol. 31, no. 7, pp. 1550–1553, 2010.