

## Research Article

# Study on MPGA-BP of Gravity Dam Deformation Prediction

Xiaoyu Wang,<sup>1</sup> Kan Yang,<sup>1</sup> and Changsong Shen<sup>2</sup>

<sup>1</sup>College of Hydrology and Water Resources, Hohai University, Nanjing 210098, China

<sup>2</sup>College of Water Conservancy and Hydropower, Hohai University, Nanjing 210098, China

Correspondence should be addressed to Xiaoyu Wang; forever\_zard@163.com

Received 2 June 2016; Accepted 21 November 2016; Published 3 January 2017

Academic Editor: Ziran Wu

Copyright © 2017 Xiaoyu Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Displacement is an important physical quantity of hydraulic structures deformation monitoring, and its prediction accuracy is the premise of ensuring the safe operation. Most existing metaheuristic methods have three problems: (1) falling into local minimum easily, (2) slowing convergence, and (3) the initial value's sensitivity. Resolving these three problems and improving the prediction accuracy necessitate the application of genetic algorithm-based backpropagation (GA-BP) neural network and multiple population genetic algorithm (MPGA). A hybrid multiple population genetic algorithm backpropagation (MPGA-BP) neural network algorithm is put forward to optimize deformation prediction from periodic monitoring surveys of hydraulic structures. This hybrid model is employed for analyzing the displacement of a gravity dam in China. The results show the proposed model is superior to an ordinary BP neural network and statistical regression model in the aspect of global search, convergence speed, and prediction accuracy.

## 1. Introduction

Dam cracks and displacement monitoring that reflect the structural aging and disease are widely used in various forms of dams (e.g., Xianghongdian Dam [1], Chencun Dam [2], and Dokan Dam [3]). Dam deformation is generally caused by three primary factors: temperature variation, chemical reactions, and live loads [4]. Many ideas have been proposed to monitor dam deformation with statistical regression analysis or mechanical calculation. Tonini [5] proposed that dam displacement may come from three causative influences: water pressure, temperature variation, and aging. Other scholars further researched these influences and proposed various models.

Statistical regression models [6] have been proposed to analyze and describe dam deformation data quantitatively. Improved models used the average temperature, certain interval number of days before the observation, to analyze Castelo Arch Dam utilizing hysteresis of air temperature in the dam [7]. Rocha used the power-polynomial of the reservoir water level value to express hydrostatic pressure factor [6]. It was not until the failure of Malpasset Arch Dam in France in 1959 and Arch Dam's reservoir-bank landslide that we began

to realize the importance of dam safety and value safety monitoring work. Thus, statistical regression model was used through the finite-element method to calculate influence factor of the dam deformation [8].

Researchers further studied and proposed deterministic models [9]. Deterministic model can be used to analyze quantitatively and qualitatively dam time observation sequence [10]. Multiple linear regression approaches, which are simple and require no prior knowledge of the structure material properties, became popular in analyzing the relationship between environment quantity and effect size [11]. Hybrid models, a combination of the deterministic models and purely statistical (regression) models, can be used to analyze dam displacement through the finite-element theory.

In the last 10 years, artificial intelligence algorithms, such as the grey system, the fuzzy mathematic theory, the time series, the wavelet theory, and bionics algorithm, were also gaining popularity. The grey system has been proposed and applied to the dam stress grey forecasting model [12, 13]. The fuzzy mathematic theory is used to analyze gravity dam instability due to interval risk [14]. The method based on the wavelet theory describes the effect of dam monitoring data of quantity separating into effect size and environment quantity

[15]. Recent studies such as artificial neural network [16] and artificial bee colony algorithm (ABC) [17] provided excellent fitting precision to assess feasibility and practicability of dam safety monitoring model.

Artificial neural network, which possesses strong ability of nonlinear function approximation and self-organizing and self-adaptive function, has been applied to the data of dams safety monitoring analysis and forecasting to remove data irregularity. Backpropagation neural networks have been proposed to monitor and predict the dam deformation while based on the actual values of a concrete gravity dam's horizontal displacement [16]. The prediction accuracy was greatly improved from previous statistical model. In addition, space displacement analysis [18] and the forward-inversion analysis method [19] were also proposed to supervise dam deformation. That BP network model's shortcomings were optimizing the structure of the network model and establishing the dam deformation forecasting model whose BP neural network was improved [20]. Recent studies [21, 22] have revealed that the artificial neural network can be applied to monitor deformation and monitor seepage of earth-rock dams.

Many studies [23, 24] demonstrated that BP neural network has some defects, such as slow learning convergence speed, local extremism, and the inconsistency and unpredictability of the structure. Therefore, many scholars combined BP neural network and other theories to improve prediction accuracy, such as fuzzy mathematics theory [25, 26]. Such combination provides better results than regression models. The combination with wavelet decomposition on the function approximation improved the fitting and prediction accuracy of dam deformation monitoring [27]. The combination with particle swarm also received good forecasting results [28]. A hybrid wavelet neural network methodology shows the superiority in improving prediction precision of time-varying behavior of engineering structures [29].

As a new type of search methods, genetic algorithm (GA) has many advantages, such as simple general, high global searching capability, strong robustness, and wide application range. Many studies have shown that these advantages can be used to optimize BP neural network's structure, the weight, threshold value, and parameters and improve the prediction accuracy. Fu et al. [30] combined genetic algorithm backpropagation neural network prediction and finite-element model simulation to improve the process of multiple-step incremental air-bending forming of sheet metal. However, little research has been conducted in dam deformation analysis. Yin et al. [31] proposed the use of the GA-BP to optimize the injection molding process parameters. Chen et al. [32] suggested that GA-based BP neural network could be a promising approach for anticipating MMP in CO<sub>2</sub>-EOR process. However, the results [30–32] may have been more beneficial if the phenomenon of premature convergence in the GA was considered; because all individuals in the population tended to have the same status and stopped evolution, the algorithm fails to give a satisfactory solution. Furthermore, when using normal GA to solve practical problems, we could have been puzzled by setting control parameters and designing the genetic operator. They tend to be given based on the actual problem tentatively. As aforementioned, the

inappropriate setting will largely influence the performance of the algorithm. Thus further studies are required for exploring the applicability and reliability of GA-BP in dam deformation and seeking better methods of improvement.

The objective of this study is to analyze the feasibility of GA-BP in dam deformation, to explore the usefulness of proposed multiple population genetic algorithm backpropagation (hybrid MPGA-BP model), and to compare it with statistical regression model and conventional BP network model with the same parameter for dam deformation analysis. The rest of the paper is arranged as follows. Section 2 points out the pertinence between the loads and the dam behavior and then presents a brief review of BP neural network and MPGA and introduces proposed MPGA-BP model. Section 3 provides a case study of a gravity dam, which includes model setting-up, the results of model analysis, and evaluation. Several figures and tables are presented to illustrate the comparison among the statistical regression model, BP neural network model, and MPGA-BP model. Section 4 presents concluding remarks.

## 2. Material and Methods

*2.1. Statistical Relation between the Loads and the Dam Behavior.* Dam structure is influenced by hydraulic, environmental, and geomechanical factors. Therefore, the situation requires us to study the variables that affect the dam behavior before applying the improved artificial neural network approaches. Based on the results of the study of [33, 34], the approved formulation for deformation  $y$  of an observing point located on the dam can be generated as follows:

$$y = y_H + y_T + y_\theta, \quad (1)$$

where  $y_H$ ,  $y_T$ , and  $y_\theta$  are the displacements contributed by hydrostatic pressure, temperature variation, and aging, respectively.

As an important factor of deformation, hydrostatic pressure can be expressed as a polynomial function for the reservoir water level  $H$  above the foundation as follows:

$$y_H = a_0 + \sum_{i=1}^n a_i H^i, \quad (2)$$

where  $a_0$  and  $a_i$  are determined by regression analysis;  $n$  is the number of water pressure factors ( $n = 3$  for the concrete gravity dam).

The displacement contributed by temperature variation can be modeled in two ways. If temperature measurements within the dam body and foundation are adequate and available, then

$$y_T = b_0 + \sum_{i=1}^n b_i T_i, \quad (3)$$

where  $T_i$  is the temperature measured values at temperature measuring point at point  $i$ ;  $b_0$  and  $b_i$  are determined by regression analysis;  $n$  is the number of temperature factors.

If temperature measurements are inadequate and unavailable, the form of measured temperature cannot be used to

describe temperature field's variation. When dam temperature field closes to the quasi-stable temperature field, we can describe approximatively the changes of temperature field within dam body through the changes of temperature outside. But there is a lag effect on the dam body internal temperature variation which was influenced by air temperature changes. So, the influence of air temperature variation on monitoring effect-quantity also causes a lag effect. As a consequence, average temperature of several days before the day of monitoring effect-quantity was served as temperature factors. Therefore the form of temperature temporal loadings can be described as follows:

$$y_T = b_0 + \sum_{i=1}^n b_i T_{i(l-m)}, \quad (4)$$

where  $T_{i(l-m)}$  is the average value of the temperature factors  $i$  in the day  $l$  to the day  $m$  before observational days;  $b_0$  and  $b_i$  are determined by regression analysis;  $n$  is the number of temperature factors. The influence contributed by aging can be modeled as follows:

$$y_\theta(t) = c_0 + \sum_{i=1}^n c_i I_i(t), \quad (5)$$

where  $I_i$  is the relation function of the aging factor;  $t$  is order value of observational days;  $t_0$  is the order value of base day;  $n$  is the number of temperature factors.  $c_0$  is the regression constant;  $c_i$  is the regression coefficient;  $c_0$  and  $c_i$  are determined by regression analysis.

**2.2. Structure of BP Neural Network.** W. McCulloch and W. Pitts opened an era of neuroscience research in 1943 since they created the mathematical model of neuron formation and imitation of biological neuron activity function. As a mathematical analogue of the biological system, it can be used to highlight processes, to deal with fuzzy information, or to display chaotic properties. BP neural network is by far the most widely used neural network, whose training method is based on the error backpropagation (BP) to the multilayer neural network. The topology of the BP neural network structure is as in Figure 1. In the figure, a three-layer structure of BP neural network divides into input layer, one hidden layer, and output layer. The individual neurons connected between two layers; there is only a connection weight between adjacent two layers. When a set of data enter the network structure from the input layer, the value of each hidden layer node can be obtained, combining with the weights between input layer and hidden layer and related algorithm. Then the value of each output node will arise by combining with the weights between hidden layer and output layer and the related algorithm. Nevertheless, the expected output and actual output had certain error; the model would utilize predetermined error values to adjust the weights between layers. This process will repeat until the predetermined error values present.

Assuming the input value of neural network is  $[x_1, x_2, \dots, x_m, \dots, x_M]^T$ , the output value is  $[y_1, y_2, \dots, y_j, \dots, y_J]^T$  and the target value is  $O_{ij}$ ; as the hidden layer excitation function,

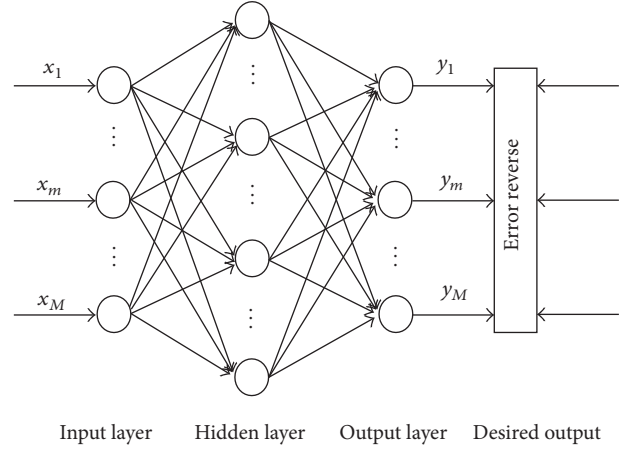


FIGURE 1: BP neural network topology structure.

$f_1$  adopts the Sigmoid function. Sigmoid function can control the arbitrary input within the scope of  $(0, 1)$ . Logarithmic Sigmoid function relation can be modeled as follows:

$$f(x) = \frac{1}{1 + \exp(-x)}. \quad (6)$$

As the output layer excitation function,  $f_2$  adopts the linear type function that can be modeled as follows:

$$f(x) = kx + c; \quad (7)$$

$w_{mk}$  is connection weight between the  $m$ th input layer neuron and the  $k$ th hidden layer neuron;  $w_{kj}$  is connection weight between the  $k$ th hidden layer neuron and  $j$ th output layer neuron;  $\theta_k$  is the threshold value of  $k$ th hidden layer neuron;  $\theta_j$  is threshold value of  $j$ th hidden layer neuron;  $\eta$  is the learning rate of the network;  $E$  is error between the output value and target value.

The BP neural network training process is mainly divided into the following two steps.

**Step 1 (forward propagation).** Obtain the input values of the  $k$ th hidden layer neuron as follows:

$$\text{net}_k = \sum_{m=1}^M w_{mk} x_{mk}. \quad (8)$$

Because the threshold value of the  $k$ th hidden layer neuron is  $\theta_k$ , we can get the  $k$ th hidden layer neuron activation value  $\text{net}_k - \theta_k$ . Therefore, the  $k$ th hidden layer output value can be expressed as  $v_k = f_1(\text{net}_k - \theta_k)$ ,  $k = 1, 2, \dots, K$ . The same can be deduced such that the input value of the  $j$ th output layer neuron is  $\text{net}_j = \sum_{k=1}^K w_{kj} v_k$ , the threshold value of the  $j$ th output layer neuron is  $\theta_j$ , the  $j$ th output layer neuron activation value is  $\text{net}_j - \theta_j$ , and the  $j$ th output layer output value is as follows:

$$y_j = f_2(\text{net}_j - \theta_j), \quad j = 1, 2, \dots, J. \quad (9)$$

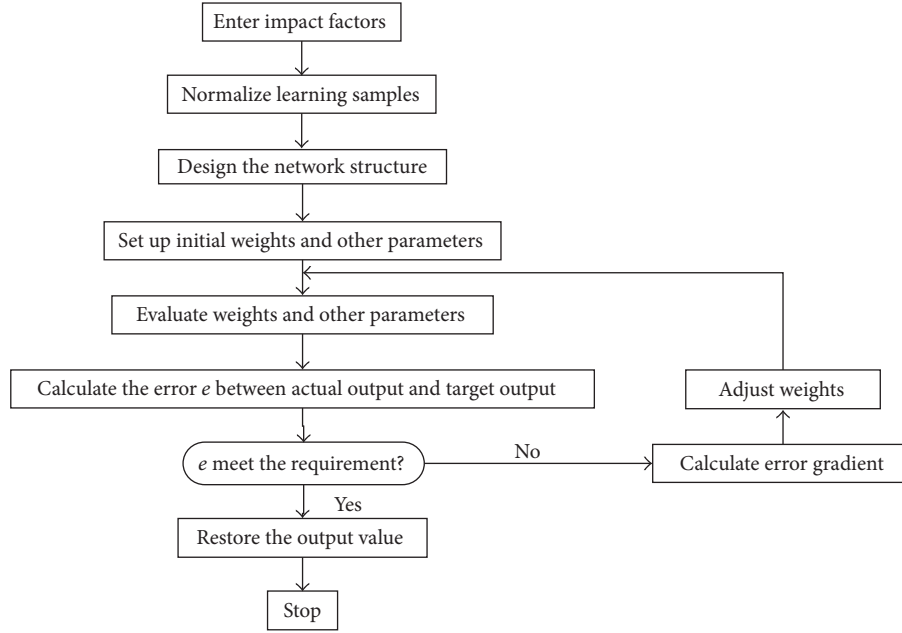


FIGURE 2: Calculation flow chart of BP neural network.

Step 2 (reverse propagation). If the difference value between output value of forward propagation and that we expect is bigger than set value, the corresponding weights need to be adjusted backwards constantly. The error function of the neural network can be expressed as follows:

$$E = \frac{1}{2} \sum_{j=1}^J (y_j - O_{ij})^2. \quad (10)$$

Based on the gradient descent method, the incremental formula of connection weights between  $k$ th hidden layer neuron and  $j$ th output layer neuron can be described as follows:

$$\begin{aligned} \Delta w_{kj} &= -\eta \frac{\partial E}{\partial w_{kj}} = -\eta \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial w_{kj}} \\ &= \eta (O_{ij} - y_j) \cdot f_2' \cdot v_k = \eta \cdot \delta_{ij} \cdot v_k, \end{aligned} \quad (11)$$

where  $\delta_{ij} = (O_{ij} - y_j) \cdot f_2'$ ,  $f_2'$  represents excitation function  $f_2(u)$  derivation of the independent variable  $u$ , and  $u$  is the activation value of this layer. In the same way, we can get threshold value increment of the output layer as follows:

$$\begin{aligned} \Delta \theta_j &= -\eta \frac{\partial E}{\partial \theta_j} = -\eta \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \theta_j} = -\eta (O_{ij} - y_j) \cdot f_2' \\ &= -\eta \cdot \delta_{ij}. \end{aligned} \quad (12)$$

Then, based on the same gradient method, the incremental formula of connection weights between  $m$ th output layer

neuron and  $k$ th hidden layer neuron can be described as follows:

$$\begin{aligned} \Delta w_{mk} &= -\eta \frac{\partial E}{\partial w_{mk}} = -\eta \sum_{j=1}^J \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial v_k} \cdot \frac{\partial v_k}{\partial w_{mk}} \\ &= \eta \sum_{j=1}^J (O_{ij} - y_j) \cdot f_2' \cdot w_{kj} \cdot f_1' \cdot x_m \\ &= \eta \cdot x_m \cdot \sum_{j=1}^J \sigma_{ijk}, \end{aligned} \quad (13)$$

where  $\sigma_{ijk} = \delta_{ij} w_{kj} f_1'$  represents excitation function  $f_1(u)$  derivation of the independent variable  $u$ . Therefore, threshold value increment of the output layer can be deduced as follows:

$$\Delta \theta_k = -\eta \sum_{j=1}^J (O_{ij} - y_j) \cdot f_2' \cdot w_{kj} \cdot f_1' = -\eta \sum_{j=1}^J \sigma_{ijk}. \quad (14)$$

Through the update incremental formula above, neurons connection weights and thresholds can be iterated and updated for the next network learning and training. Subsequently, circuit training the above steps many times, the neural network connection weights and thresholds are continually updated. Meanwhile, output error will also tend to be minimum. When the minimum reaches set value, the loop is completed; if not reached, the cycle training continues for many times. The BP neural network calculation flow diagram was shown in Figure 2.

2.3. Multiple Population Genetic Algorithm (MPGA). Although the genetic algorithm shows excellent characteristics

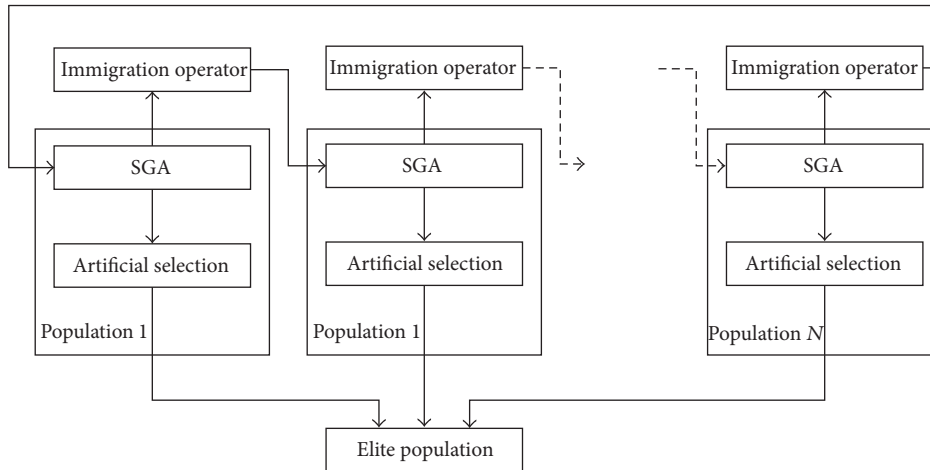


FIGURE 3: MPGA structure diagram.

of its global efficiency, it also has its disadvantages in practical application. One of the important is premature convergence which is considered the common phenomenon in the GA. It has much effect on the solution of the optimal value, whose main characteristic is that all individuals in the population present a trend and terminate evolution. Thus, the satisfied solution cannot be obtained. The multiple population structure has been introduced in order to solve the problem of premature convergence of the GA. This paper uses multiple population genetic algorithm replacing conventional standard genetic algorithm. The structure diagram was shown in Figure 3.

MPGA is mainly done through the following optimization based on SGA:

- (1) MPGA introduces multiple populations to optimize searching simultaneously, while SGA has only a single population. Different population can achieve different searching purposes with control different parameters.
- (2) Through the migration operator contacting various populations, the multiple population coevolution can be realized. It will eventually obtain the optimal solution.
- (3) Artificial selection operator will save the best individual of each population in evolution, which is regarded as criterion of the algorithm convergence.

In MPGA, each population selects different control parameters. Meanwhile, the value of crossover probability and mutation probability decided the algorithm's global search and local search ability. Although many articles and scholars advised choosing a larger crossover probability  $P_c$  (0.7~0.9) and smaller mutation probability  $P_m$  (0.001~0.05), approaches to  $P_c$  and  $P_m$  are still not sure. Many kinds of possibility lead to the final optimization results causing great difference. MPGA overcomes this shortcoming of SGA. The multiple cooperation in a kind of coevolution with the population of random control parameters takes into account

the global search and local search ability in SGA. But while all populations in the MPGA are independent of each other, they can contact by immigration operator (immigrant). That immigrant moves the best individual in kinds of population in the process of evolution into other populations every certain number of generations for exchanging information between populations. As aforementioned, the immigrant in the MPGA is crucial. If the immigrant does not exist, there is no contact between kinds of population and MPGA is equivalent to using different control parameters for the SGA calculation for many times. Then, the problem of premature convergence is not solved. Every evolution generation should select the best individual of other populations through artificial selection operator (EliteIndividual) and put it in elite population to save. Unlike other populations, elite population is without related genetic operation, to ensure the integrity of the best individual. At the same time, elite population is termination criterion of the algorithm, where one uses the best individual keeping the number of generations as termination criterion. This method is more reasonable than the SGA, whose biggest number of generations is seen as the termination criterion.

**2.4. Multiple Population Genetic Algorithm with Backpropagation (MPGA-BP).** As aforesaid, the BP neural network and the standard genetic algorithm (SGA) have their own shortcomings. This study suggests using multiple population genetic algorithm (MPGA) to solve the problem of premature convergence in SGA and to optimize the BP neural network weights and threshold for speeding up the learning speed of network. MPGA-BP model includes the advantages of the BP neural work, such as nonlinear mapping ability, self-learning and adaptive ability, and strong fault-tolerant ability. It uses genetic algorithm to overcome the BP neural network easy falling into local minimum value, slow convergence speed, initial threshold and weight values difficult to determine, and other shortcomings. Meanwhile, it uses concept of multiple population to solve the problem of premature convergence of genetic algorithm itself retaining the global search ability

of genetic algorithm. The modeling steps of MPGA-BP generally include the following.

*Step 1.* Determine the topological structure of BP network (determine the number of weights and thresholds to be optimized).

*Step 2.* Genetic operation is based on each population's initialized code (put the norm of prediction error matrix as objective function; design the fitness function).

*Step 3.* Set crossover and mutation parameters for each population (introduce multiple population concept to genetic algorithm).

*Step 4.* Collect the best individual of various populations to form an elite population through immigration artificial selection operation (realize best individual with the optimal weights and thresholds of the network).

*Step 5.* Assign the best individual weights and threshold to the BP neural network.

*Step 6.* Get the final result through the network training, testing, and verifying.

*2.4.1. Set Parameters.* Set the size of the population as  $N$ . Then, an initial population can be generated randomly as follows:

$$W = (W_1, W_2, \dots, W_N)^T. \quad (15)$$

Each individual population needs encoding because solutions that need to be optimized in initial populations are BP network's weights and thresholds. This study adopts binary encoding method, which encodes every individual into a binary string. Here, the size of the individual consists of four parts; there are connection weights of input layer and hidden layer, thresholds of implicit layer, connection weights of hidden layer and connection layer, and thresholds of output. Each individual's weights and thresholds use  $S$  bit binary code. Connecting all codes of weights and thresholds forms an individual code string. Assume that the input layer contains  $M$  nodes, the hidden layer contains  $L$  nodes, and the output layer contains  $J$  nodes; then the length of individual code string is  $K = (M \cdot L + L + L \cdot J + J) \cdot S$ . Therefore each individual code string can be expressed as follows:

$$W_i = (w_1, w_2, \dots, w_k), \quad i = 1, 2, \dots, N. \quad (16)$$

*2.4.2. Design Fitness Function.* In the genetic algorithm, the solution of issue is shown as the values of population individual and survival of the fittest through fitness function. In this paper, we are going to get the optimal weight and threshold of BP network, so the individual values are expressed as weights and thresholds of BP network and the error norm of the matrix can be chosen as target function. There will be a positive and negative value of the target function, so we need to design a fitness function. We need to ensure that fitness value is nonnegative based on the relationship between

fitness function and target function. At the same time, the optimization direction of target function is the increased direction of fitness.

For the optimization problem of target function's minimum, we need to add a minus sign to transform it into the optimization problem of target function's maximum in theory as follows:

$$F(x) = -\min f(x). \quad (17)$$

For the problem of target function's maximum, we can directly set up fitness function as equal to the target function, when the target function is always positive as follows:

$$F(x) = \max f(x). \quad (18)$$

This paper mainly involves the target function's minimum, so we choose the fitness function as formula (17).

*2.4.3. Design Genetic Operators and Multiple Populations.* MPGA includes SGA genetic operation: selection, crossover, and mutation. Selection operation is mainly to choose excellent individual with a certain probability from old population to form new population in order to reach the goal of breeding the next generation individuals. Crossover operation is to select two individuals at random from the population and to pass parents' excellent genetics to offspring through exchanging and combination of two chromosomes for generating new excellent individuals. The main purpose of the mutation operation is to maintain the diversity of population.

In MPGA model, different population has different crossover probability  $P_c$  and smaller mutation probability  $P_m$ . Crossover probability  $P_c$  generates in  $[0.7, 0.9]$  at random while mutation probability  $P_m$  generates in  $[0.001, 0.5]$  at random. Then selection of each population uses roulette wheel to sample. The crossover operator uses simple single-point crossover operator while mutation operator adopts the method of discrete variable. By introducing the migration operator (immigrant) and artificial selection operator (EliteIndividual), the best individual of various populations can form elite population. The best individual finally got is best solution of the error function, which is the optimal weights and thresholds of the neural network. Multiple population genetic neural network calculation processes were shown in Figure 4.

### 3. Case Study

*3.1. Data Set.* A series of observations including the water level, temperature, and aging of a gravity dam in China were used building the MPGA-BP model. The crest length of the dam is 1080 m, the crest elevation is approximately 91.7 m, the upstream slope is 95%, the downstream slope is 78%, the profile is approximately the triangle, and the total reservoir capacity is 11.5 billion cubic meters. Observation wire system has been set within crest cable corridor. In this study, we examined the recorded data of the monitoring point No. 16, which is located at the centre of impervious reinforced concrete face of the overflow dam section as shown

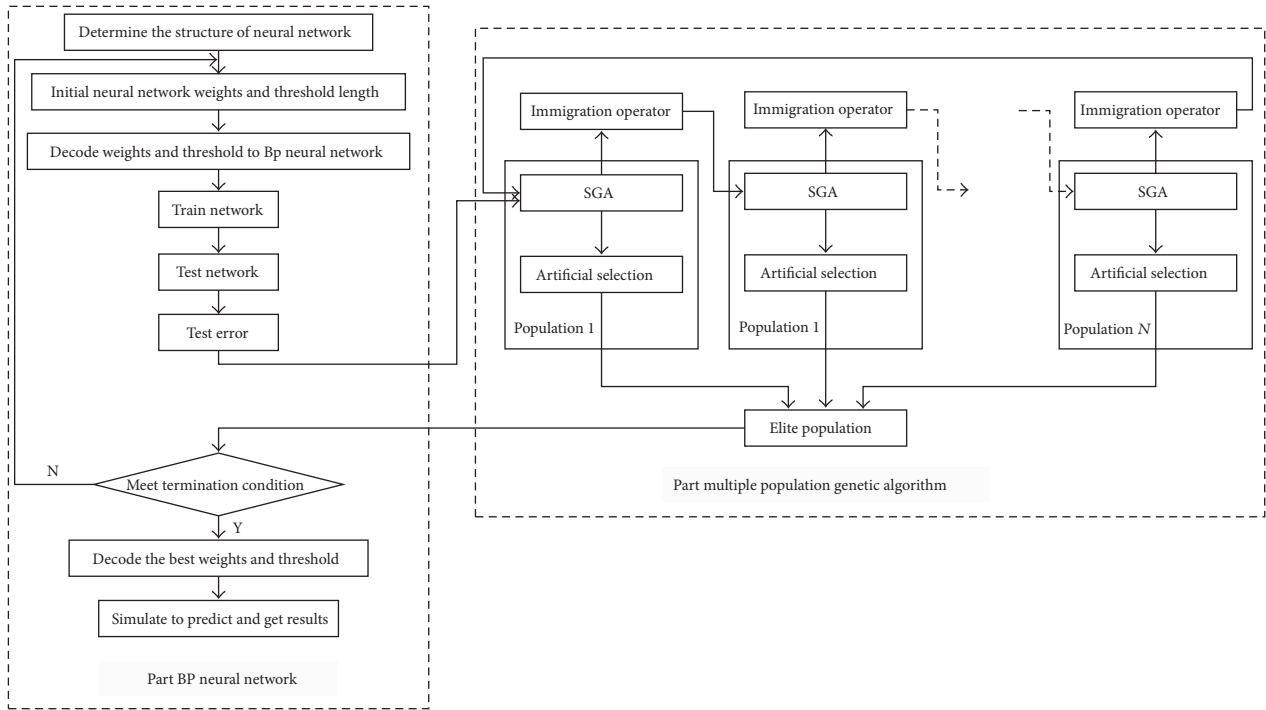


FIGURE 4: Multiple population genetic neural network calculation flow diagram.

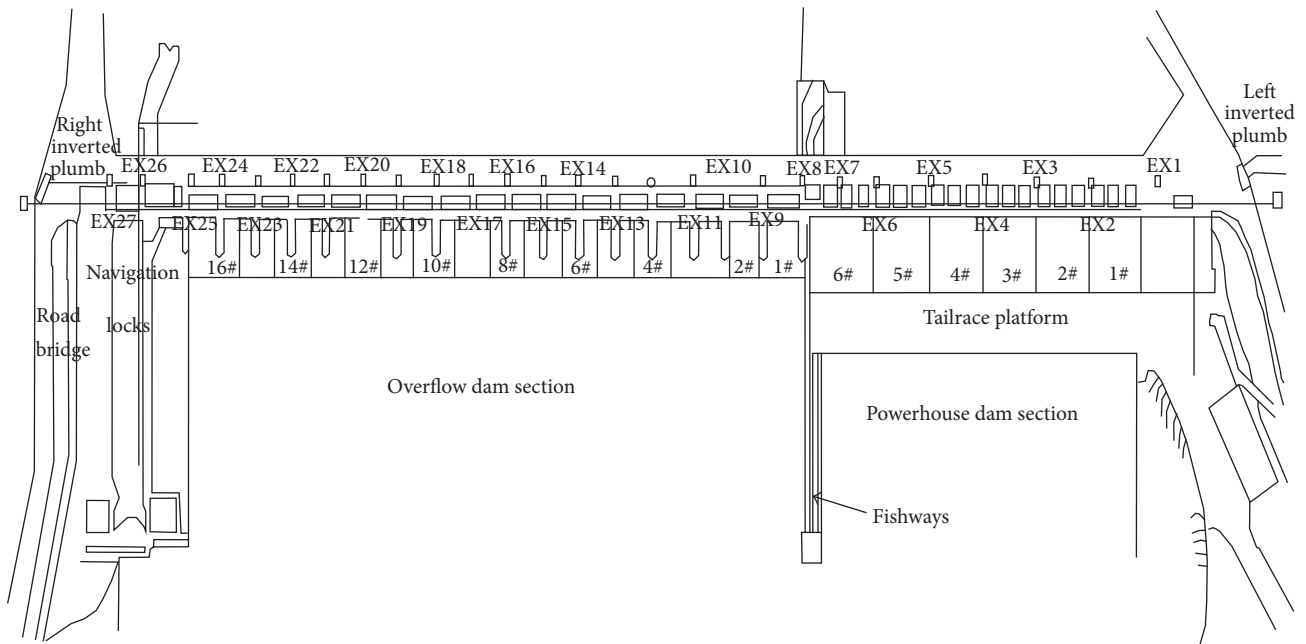


FIGURE 5: Arrangement of monitoring points.

in Figure 5. The test data set, built up over a period of ten years, consists of 489 pairs. These data reflect the change of displacement, water level, temperature, and aging. In order to avoid raising the overfitting problem in BP network training, we use half data for training, a quarter for validating and a quarter for predicting. By considering the dam characteristics in Section 2.1 and the available observations, we selected

the ten variables listed in Table 1 as influential factors for forecasting dam deformation.

3.2. *Determinate Hidden Layer Nodes.* There exists a Kolmogorov theorem, namely, the continuous function representation theorem, ensuring that any continuous function or mapping can be used in a three-layer neural network to

TABLE 1: Variables used to forecast dam deformation.

Description	Variable	Calculation
Water level	Var. 1	$H$
	Var. 2	$H^2$
	Var. 3	$H^3$
Temperature	Var. 4	$T_0$
	Var. 5	$T_5$
	Var. 6	$T_{15}$
	Var. 7	$T_{30}$
	Var. 8	$T_{60}$
Aging	Var. 9	$\theta$
	Var. 10	$\ln \theta$

realize. The network includes input layer, output layer, and hidden layer. Then, the number of hidden layer nodes has a great impact on the generalization and the training speed. If the number of nodes were too small, network nonlinear mapping ability would be low, prediction accuracy would be not high, and the network would be thin. If the number of nodes were too large, the learning time would be too long and the training speed would be slow. So the appropriate number of nodes is crucial to a strong network. At present, there is no clear theoretical guidance to selection method of the number of hidden layer nodes. The empirical formula commonly used in the application is as follows:

$$l = \sqrt{m + n} + a, \quad (19)$$

where  $m$ ,  $n$ , and  $l$  are the number of input layer nodes, the number of output layer nodes, and the number of hidden layer nodes.  $a$  is the constant of 1~10.

In addition, the number of nodes can be found out through the test algorithm. We eventually get the appropriate value through either training from smaller number of nodes and increasing gradually the number of nodes based on the change of network output error or training from larger number of nodes and reducing gradually the number of nodes. This method consumes more time and energy for a large number of calculations. And not only that, it is able to quickly find suitable number of hidden layer nodes on account of strong randomness.

This study uses the method of combining the empirical formula and test algorithm to determine the number of nodes in the hidden layer. In the case,  $m$  and  $n$  are 10 and 1. The values range of  $l$  is 5~14 based on formula (19). Then, we utilize the method of gradual growth to start the trial and compare mean square error values (MSE) of network verification results under the condition of different number of nodes. Do test 10 times, respectively, on different number of nodes (5~14), and then take their average MSE to compare. The test results were listed in Table 2:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2, \quad (20)$$

where  $a_i$  and  $t_i$  are network output value and desired output of validation sample.

Through observing the relationship between the number of hidden layer nodes and the MSE value we concluded that if the number of hidden layer nodes were 13, the MSE value of network would be smaller. So we determine that the number of hidden layer nodes is 13 and the network structure is 10-13-1.

**3.3. Sample Data Normalization.** On account of the difference of units and dimension within sample data (displacement, water pressure, temperature, and aging), we need to carry out normalization processing with sample values of input factors and output factors. Retaining original nature of sample data makes the values in  $[-1, 1]$ . Normalization formula is as follows:

$$x_{\text{nor}} = 2 \times \frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1, \quad (21)$$

where  $x$ ,  $x_{\max}$ , and  $x_{\min}$  are the original sample data, the maximum of sample data, and the minimum of sample data.  $x_{\text{nor}}$  is the normalized sample data. The *premnmx* function can be used to achieve this process in the MATLAB. For network output value, normalized deformation on the above formula can be used to obtain sample data.

**3.4. Optimization Process of MPGA-BP.** Common encoding includes binary code, grey code coding, and real coding. The binary coding is one of the most common kinds of coding way. It not only facilitates implementation of the genetic operations such as crossover and mutation, but also causes encoding and decoding operation to be easy. This study uses a binary code, which is set up of binary notation  $\{0, 1\}$ . Code string consists of connection weights of input layer and hidden layer, hidden layer threshold, connection weights of hidden layer and output layer, and output layer threshold. As aforementioned, the length of individual coding is  $K = (M \cdot L + L + L \cdot J + J) \cdot S$ , where  $S$  are binary digits, which have been set as  $S = 10$ . So the length of individual coding is  $K = (10 \times 13 + 13 + 13 \times 1 + 1) \times 10 = 1570$ . Then,  $N$  individuals like these form a group, set  $N = 100$ . The scope of the initial weights and threshold has been set in  $(-1, 1)$ . In the MPGA, the number of the populations is more than one, which has been set as  $MP = 10$ . There is mutual relationship between multiple populations.

Fitness function is a method to judge whether individual population is good or bad. We decode the code representing weights and threshold on the code string and substitute into the BP neural network. By using training samples to train the network and using testing samples to test the network the test error  $e = \sum |y_i - o_i|$  can be obtained, where  $y_i$  is the expected output of  $i$ th node and  $o_i$  is the predicted output of  $i$ th node. In order to make the error of predicted values and expectations as small as possible in the BP network prediction, we may choose error matrix norm as objective function. Considering the objective function that the model final seeks is minimum, we can simply add a minus sign. Therefore, the fitness function can be used as follows:

$$f = -e(x). \quad (22)$$



TABLE 2: Relationships between the number of hidden layer nodes and the MSE value.

$n$	5	6	7	8	9	10	11	12	13	14
MSE	0.00286	0.00247	0.0023	0.00254	0.00242	0.00251	0.00241	0.00243	0.00228	0.00231

TABLE 3: Parameters used in implementing MPGA-BP variable selection.

Hidden layer transfer function	tansig
Output layer transfer function	purelin
Training function	trainlm
Times of training	2000
Training goal	0.001
Learning rate	0.1
Number of populations	10
Population size	100
Generation gap	0.9
Binary digits	10

In MATLAB, Sheffield genetic algorithm toolbox has a fitness distribution function named ranking based on the sort to achieve the goal as follows:  $\text{Fitn} V = \text{ranking}(-\text{obj})$ , where  $\text{obj}$  is the output of the objective function.

The optimization process of MPGA certainly contains the inherent optimization operation of standard genetic algorithm (SGA), which has selection, crossover, and mutation operation. Selection operation adopts roulette selection operator (RWS) in the Sheffield genetic algorithm toolbox. Crossover operation adopts single-point crossover operator (xovsp). Crossover probability  $P_c$  arises at random in the range of  $[0.7, 0.9]$ ; then they are endowed corresponding population that generated randomly. Mutation operation adopts discrete mutation operator (mut). Just like crossover probability, mutation probability arises at random in the range of  $[0.001, 0.05]$ ; then they are endowed corresponding population that generated randomly. The parameters were set to default value (see Table 3) so as to obtain the best exploitation. Not only does MPGA optimization contain the SGA optimization, but it also contains the migration operator and artificial selection operator. Various groups are relatively independent, but they contact with each other by immigration operator (immigrant). It is able to take the best individual arising from genetic operation into other populations every certain evolution population. This process allows the exchange of information between populations. The best individual in other populations can be selected by artificial selection operator (EliteIndividual) to preserve within the elite population. Seeing the minimum preservation population of the best individual as termination substratum, the elite population no longer does genetic operation. The above-mentioned operation effectively solves the premature problem in SGA.

**3.5. Modeling and Forecasting.** In this paper, we established MPGA-BP model compiled by MATLAB software to train the sample data and forecast dam deformation. From the error

changing curve of the inducing generation (see Figure 6), we decided to demonstrate the relationship between error changes and genetic generations. The figure shows a trend of decrease of network error as a whole with the increasing of generation. After the genetic generation reached 21 generations, fitness is stable and gradually converges to 0.3781. Moreover, the optimal initial weights and threshold were substituted into BP neural network for training. In Figure 7, the blue line represents the training process while black line represents the target error. The figure shows the termination of network training after 8 times of training. The training error of sample is 0.000864965 which is less than the target 0.001.

On the other hand, stepwise regression and ordinary BP have the same topological structure 10-13-1 and were also carried out in this study for comparison with MPGA-BP model. The same data has been applied to train and forecast. Details of the investigation are listed in Table 4.

Through the contrast in Table 4 it can be seen that the error of statistical regression is higher than that of MPGA-BP model and BP model. The reason is that the limited data for linear regression model is hard to fitting nonlinear relationship accurately. The prediction error range of MPGA-BP network model is in  $[-1, 1]$  while that of BP network model is in  $[-1.8, 2.6]$ . By contrast, MPGA-BP model is better than BP model on the overall accuracy. By calculating, we can find that the standard deviation of prediction error is 0.4778 and 0.7545, respectively. What is more, the average relative error is 2.35% and 8.04%, respectively. Statistics indicate that MPGA-BP model is much more stable than BP model. And not only that, but the number of network training times is 8 and 14 times, respectively, which means MPGA-BP model is superior to the BP network model in prediction accuracy and the convergence. In order to better reflect the superiority of the three models, test sample error matrix norm of the predicted and measured values can be used to compare. Norm results came from the MATLAB and SPSS running in Table 5.

By contrast, it is not difficult to find that error matrix norms of MPGA-BP model are smaller than other two models in the process of the training, validation, and prediction. This reflects that MPGA-BP model is superior to other two models from another aspect. From Figure 8 we can see the forecasting process line of three models compared with the measured values curve. Likewise, from Figure 9 we can see the comparison between the three kinds of model prediction error curve.

Figures 10 and 11 are the simulation rendering of MPGA-BP model and its error distribution curve, respectively. Figure 10 illustrates that the transformation trend of the predicted and the measured values is consistent on the whole. The error distribution interval is in the range of  $[-1, 1]$ . It is only a few points in this interval that fitting precision is low

TABLE 4: Comparison of measured value under the conditions and predicted values for three models.

Serial number	Measured value	MPGA-BP network		BP network		Statistical regression model	
		Predicted value	Absolute error	Predicted value	Absolute error	Predicted value	Absolute error
1	3.42	3.219	-0.201	2.9499	-0.4701	3.4946	0.0746
2	2.92	3.099	0.179	2.6533	-0.2667	1.1312	-1.7888
3	2.40	2.2092	-0.1908	2.6708	0.2708	1.0325	-1.3675
4	1.61	0.9662	-0.6438	0.9569	-0.6531	-1.6487	-3.2587
5	0.28	-0.11	-0.39	0.617	0.337	-1.8689	-2.1489
6	-0.66	-0.4886	0.1714	-1.8839	-1.2239	-0.6495	0.0105
7	-2.01	-5.757	-0.0809	-2.4074	-0.3974	-1.8241	0.1859
8	-2.05	-2.1287	-0.0787	-2.2925	-0.2425	-0.6052	1.4448
9	-1.05	-0.7926	0.2574	-0.4213	0.6287	0.102	1.152
10	-0.31	-0.3099	0.0001	-0.7973	-0.4873	-0.1388	0.1712
11	1.54	1.5925	0.0525	0.9992	-0.5408	-0.3177	-1.8577
12	2.45	2.7487	0.2987	2.8884	0.4384	0.3896	-2.0604
13	3.57	3.5094	-0.0606	3.3034	-0.2666	5.3142	1.7442
14	4.03	3.6277	-0.4023	5.1262	1.0962	5.4135	1.3835
15	2.86	3.4951	0.6351	3.739	0.879	6.2595	3.3995
16	1.75	1.7595	0.0095	1.2968	-0.4532	3.3688	1.6188
17	0.25	0.133	-0.117	0.3085	0.0585	0.0966	-0.1534
18	-0.57	-0.8113	-0.2413	-1.2609	-0.6909	-2.2653	-1.6953
19	-1.34	-1.6984	-0.3584	-0.8202	0.5198	-1.2542	0.0858
20	-1.71	-1.7945	-0.0845	-2.1513	-0.4413	-0.1046	1.6054
21	-1.29	-1.0311	0.2589	-1.137	0.153	-0.166	1.124
22	-0.41	-0.4861	-0.0761	-0.6547	-0.2447	0.8821	1.2921
23	1	0.9514	-0.0486	0.9505	-0.0495	1.8992	0.8992
24	2.58	2.6243	0.0443	2.2951	-0.2849	2.8736	0.2936
25	3.44	3.2802	-0.1598	2.2069	-1.2331	2.0372	-1.4028
26	2.78	3.0696	0.2896	2.2219	-0.5581	0.1712	-2.6088
27	2.54	2.9736	0.4336	2.8318	0.2918	0.3636	-2.1764
28	1.61	0.9371	-0.6729	0.3634	-1.2466	-2.4768	-4.0868
29	0.34	-0.0086	-0.3486	0.8106	0.4706	3.3434	3.0034
30	-0.53	-0.7678	-0.2378	-1.0756	-0.5456	0.106	0.636
31	-1.63	-1.6412	-0.0112	-1.3885	0.2415	0.4952	2.1252
32	-1.81	-1.6554	0.1546	-1.5228	0.2872	-0.7823	1.0277
33	-1.45	-1.4973	-0.0473	-2.3524	-0.9024	-3.3323	-1.8823
34	-0.13	-0.4999	-0.3699	-0.2023	-0.0723	-1.7023	-1.5723
35	1.44	1.6243	0.1843	0.8773	-0.5627	1.2163	-0.2237
36	2.65	2.9985	0.3485	2.9386	0.2886	4.2088	1.5588
37	3.3	2.9541	-0.3459	3.5507	0.2507	3.9739	0.6739
38	3.05	3.5091	0.4591	3.7402	0.6902	4.7142	1.6642
39	2.55	3.0312	0.4812	3.2089	0.6589	3.7868	1.2368
40	2.07	2.4561	0.3861	2.5193	0.4493	2.9521	0.8821
41	0.75	0.64996	-0.1004	-0.9744	-1.7244	-0.7985	-1.5485
42	-0.36	-0.1018	0.2582	-0.7845	-0.4245	-3.5358	-3.1758
43	-1.29	-1.4917	-0.2017	-1.9291	-0.6391	-1.8964	-0.6064
44	-2.08	-2.2348	-0.1578	-3.1187	-1.0387	-2.698	-0.618
45	-0.87	-0.9948	-0.1248	-0.7079	0.1621	2.0642	2.9342
46	0.75	0.7598	0.0098	1.1541	0.4041	3.0046	2.2546
47	1.22	1.0021	-0.2179	2.2424	1.0224	6.9321	5.7121
48	2.45	1.7558	-0.6942	1.897	-0.553	1.4844	-0.9656

TABLE 4: Continued.

Serial number	Measured value	MPGA-BP network		BP network		Statistical regression model	
		Predicted value	Absolute error	Predicted value	Absolute error	Predicted value	Absolute error
49	4.09	3.8346	-0.2554	3.5129	-0.5771	3.2155	-0.8745
50	3.68	3.7379	0.0579	4.338	0.658	4.2639	0.5839
51	2.65	2.1345	-0.5155	2.0135	-0.6365	5.1232	2.4732
52	1.63	1.4641	-0.1659	1.9165	0.2865	4.2448	2.6148
53	0.51	-0.1674	-0.6774	-0.1089	-0.6189	0.1948	-0.3152
54	-0.28	-0.4346	-0.1546	0.4133	0.6933	-2.697	-2.417
55	-2.26	-1.3331	0.9269	0.3333	2.5933	-2.428	-0.168
56	-2.74	-2.3767	0.3633	-1.9026	0.8374	-3.7954	-1.0554
57	-2.13	-1.4265	0.7035	-2.2508	-0.1208	-4.662	-2.532
58	-0.87	-1.3164	-0.4464	-1.0228	-0.1528	-1.1613	-0.2913
59	0.19	0.1876	-0.0024	-0.0096	-0.1096	1.849	1.659
60	1.78	1.9028	0.1228	1.1716	-0.6084	0.4988	-1.2812

TABLE 5: The related parameters for three models.

Model	Prediction error norm	Prediction error standard deviation	Prediction relative error	Training error norm	Validation error norm
MPGA-BP network	0.4581	0.4778	2.35%	0.4379	0.4769
BP network	0.6049	0.7545	8.04%	0.4855	0.5071
Statistical regression model	1.6821	1.8539	18.38%	1.5478	1.7456

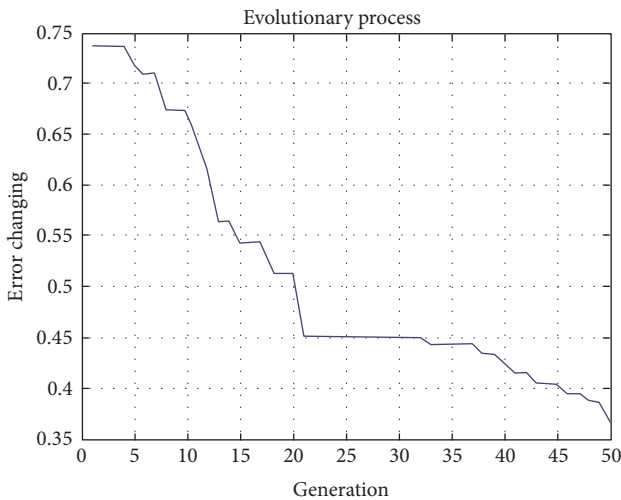


FIGURE 6: Error evolution process curve of MPGA-BP network.

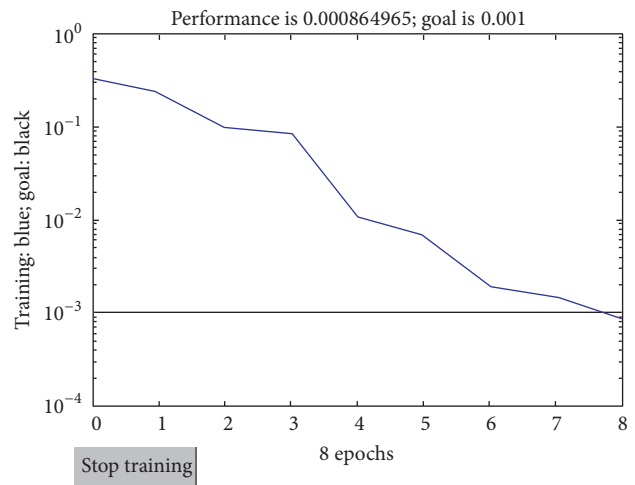


FIGURE 7: Training process curve of MPGA-BP network.

and error is big. The individual point has certain deviation because BP network reflects the common sense rather than a single individual.

To sum up, we suggest that MPGA-BP model is far superior to statistical regression model and BP network model. It can be proved from three angles of prediction accuracy, convergence speed, and error matrix norm. It means that MPGA-BP model that has been built is scientific and effective.

It is feasible that predicting gravity dam deformation uses this model.

#### 4. Conclusions

This study extends the methods of genetic algorithm with backpropagation neural network (GA-BP) and multiple population genetic algorithm (MPGA) to gravity dam deformation analysis. A hybrid multiple population genetic algorithm

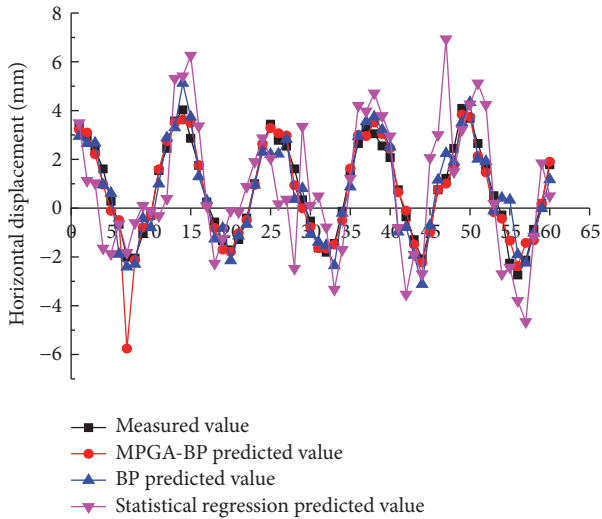


FIGURE 8: Predicted values of three models compared with the measured values.

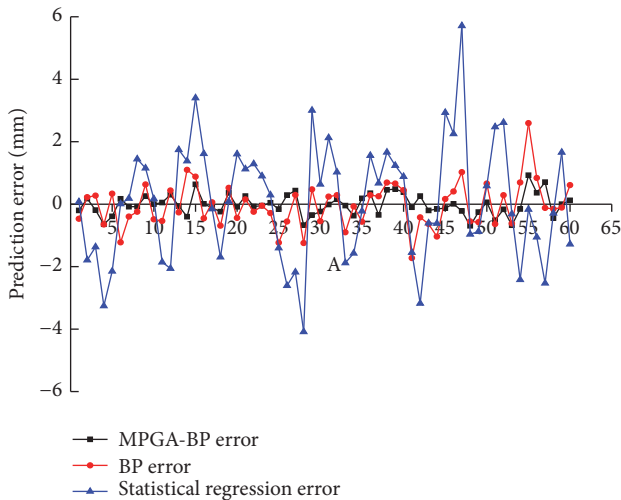


FIGURE 9: Prediction error curve of three models.

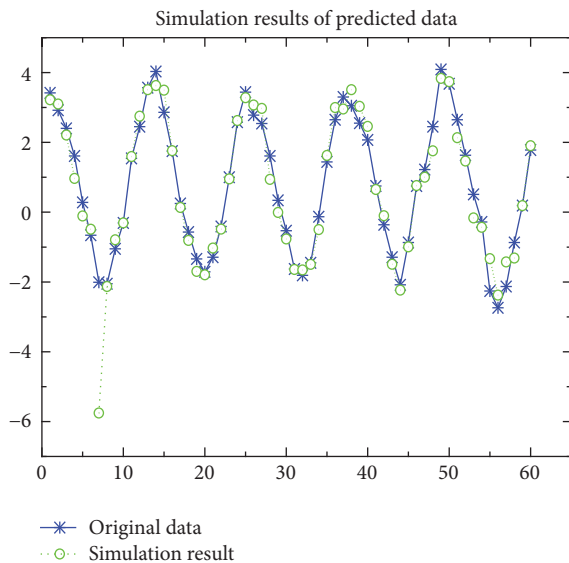


FIGURE 10: Comparison of predicted and measured values of horizontal displacement (unit: mm).

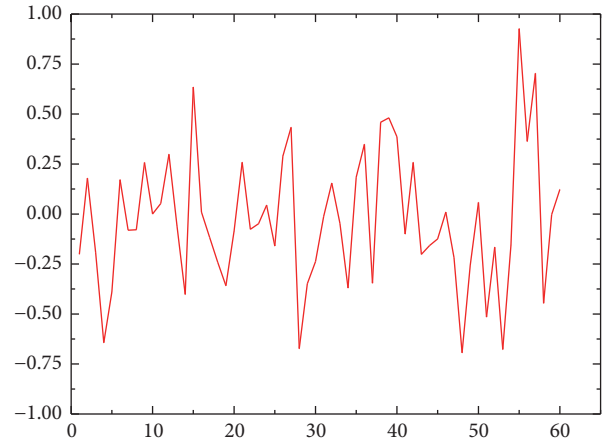


FIGURE 11: Error distribution curve (unit: mm).

backpropagation (MPGA-BP) neural network algorithm has been proposed to cope with premature convergence problem and local extremum problem. A case study using the observations of a gravity dam in China has been presented and discussed to examine the performance of the proposed model. Compared with the statistical regression model and the traditional neural network algorithm, this model prediction accuracy is improved greatly. The applicability of MPGA-BP for the prediction of gravity dam deformation has been highlighted. The results show that the proposed model can be used as an excellent tool instead of that commonly used in gravity dam deformation analysis and prediction.

### Competing Interests

The authors declare that they have no competing interests.

### Acknowledgments

This study was partly supported by the National Basic Research Program of China (973 Program) (2012CB417006) and National Science Support Plan Project of China (2009BAC56B03). The authors would like to thank Professor Shen for his generous sharing of the deformation-observation data of a gravity dam. Special thanks are due to Professor Wan (University of Houston Victoria, USA) for his constructive comments, which improved the manuscript substantially.

### References

- [1] Y. Wang, "Crack reason analysis and handling suggestions for xianghongdian dam," *Jianghuai Water Resources Science and Technology*, vol. 6, pp. 9–10, 2006 (Chinese).
- [2] H. Xu and X. Li, "Inferring rules for adverse load combinations to crack in concrete dam from monitoring data using adaptive neuro-fuzzy inference system," *Science China Technological Sciences*, vol. 55, no. 1, pp. 136–141, 2012.
- [3] E. Dafny, K. J. Tawfeeq, and K. Ghabraie, "Evaluating temporal changes in hydraulic conductivities near karst-terrain dams: Dokan Dam (Kurdistan-Iraq)," *Journal of Hydrology*, vol. 529, no. 1, pp. 265–275, 2015.

- [4] E. Koroneos, G. Poulakos, and G. Fivos Sargentis, "DAMS-Materials-Environmental Impacts," National Technical University of Athens. Laboratory of Materials of Construction (online), 2004.
- [5] D. Tonini, "Observed behavior of several Italian arch dams," *Journal of the Power Division*, vol. 82, no. 6, pp. 1–26, 1956.
- [6] M. Rocha, "A quantitative method for the interpretation of the results of the observation of dams," in *Proceedings of 6th Congress on Large Dams*, New York, NY, USA, 1985.
- [7] A. C. Xerez and J. F. Lamas, "Methods of analysis of arch dam behavior," in *Proceedings of 6th Congress on Large Dams*, vol. 2 of *Congress 6, question 21, report 39*, pp. 407–431, New York, NY, USA, 1958.
- [8] P. Marazio, "Behavior of Enel's large dams," Enel's report, Roma, 1980.
- [9] P. Bonaldi, M. Fanelli, and G. Giusepitti, "Displacement forecasting for concrete dams via deterministic mathematical models," *Water Power & Dam Construction*, vol. 29, pp. 74–78, 1980.
- [10] J. O. Pedro, "Stress evaluation in concrete dams, the example of verosa dam," in *Proceedings of the International Conference on Safety of Dams*, Coimbra, Portugal, 1984.
- [11] Q. M. Guedes and P. S. Coelho, "Statistical behavior model of dams," in *Proceedings of the 15th ICOLD Congress*, Q.56, R.16, Lausanne, Switzerland, 1985.
- [12] J.-J. Deng, "Control problems of grey systems," *Systems & Control Letters*, vol. 1, no. 5, pp. 288–294, 1982.
- [13] Z. Wu, C. Gu, Z. Shen et al., "Theory and application of dam safety synthetical analysis and assessment," *Advances in Science and Technology of Water Resources*, vol. 18, no. 3, pp. 5–9, 1998 (Chinese).
- [14] H. Su and Z. Wen, "Interval risk analysis for gravity dam instability," *Engineering Failure Analysis*, vol. 33, pp. 83–96, 2013.
- [15] Y. Cui, L. Zhang, and J. Zuo, "Abstracting response component of dam observation data based on wavelet analysis," *Water Resources and Power*, vol. 27, no. 5, pp. 80–82, 2009 (Chinese).
- [16] W. Hong, "Study on prediction model of gravity dam displacement monitoring based on artificial neural network," *Journal of Inner Mongolia Industrial University Natural Science Edition*, vol. 18, no. 4, pp. 264–270, 1999 (Chinese).
- [17] Z. Mei, Q. Xu, and F. Kang, "Dam deformation monitoring based on stepwise regression model with artificial bee colony algorithm," *Journal of Disaster Prevention and Mitigation Engineering*, vol. 33, no. 6, pp. 651–670, 2013 (Chinese).
- [18] N. Deng, F. Qiu, and H. Xu, "Application of BP model to data analysis of earth-rock," *Engineering Journal of Wuhan University*, vol. 34, no. 4, pp. 17–20, 2001.
- [19] H. Zhou, *Research on Analysis and Back-Analysis of Deformation Monitoring for Dam Based on Artificial Neural Network*, Wuhan University, Wuhan, China, 2004.
- [20] H. Xu and G. Li, "Application of BP neural network to observed data processing for dam based on Matlab," *Journal of Wuhan University of Hydraulic and Electric Engineering*, vol. 38, no. 3, pp. 50–53, 2005 (Chinese).
- [21] C. Y. Kao and C. H. Loh, "Monitoring of long-term static deformation data of Fei-Tsui arch dam using artificial neural network-based approaches," *Structural Control and Health Monitoring*, vol. 20, no. 3, pp. 282–303, 2013.
- [22] J. Qiu, D. Zheng, and K. Zhu, "Seepage monitoring models study of earth-rock dams influenced by rainstorms," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1656738, 11 pages, 2016.
- [23] Y. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1331–1343, 1998.
- [24] Y. Li, A. B. Rad, and W. Peng, "An enhanced training algorithm for multilayer neural networks based on reference output of hidden layer," *Neural Computing and Applications*, vol. 8, no. 3, pp. 218–225, 1999.
- [25] J. Chen and X. Lu, "Application of fuzzy artificial neural network to observation data analysis of earth dam monitoring," *Journal of Hydraulic Engineering*, no. 1, pp. 19–22, 2000 (Chinese).
- [26] S. Tian and J. Chen, "An evaluation model for tailings storage facilities using improved neural networks and fuzzy mathematics," *Journal of Applied Mathematics*, vol. 2014, Article ID 328902, 9 pages, 2014.
- [27] P. Gao and G. Xue, "Deformation monitoring model for dam based on wave-net and its application in forecasting," *Journal of Hydraulic Engineering*, no. 7, pp. 107–110, 2003 (Chinese).
- [28] F. Zhang, *Research on the Application of BP Neureal Network Based on Partical Swarm Optimization in Dam Displacement Prediction*, Liaoning Technical University, 2010.
- [29] M. Cao, P. Qiao, and Q. Ren, "Improved hybrid wavelet neural network methodology for time-varying behavior prediction of engineering structures," *Neural Computing and Applications*, vol. 18, no. 7, pp. 821–832, 2009.
- [30] Z. Fu and J. Mo, "Multiple-step incremental air-bending forming of high-strength sheet metal based on simulation analysis," *Materials and Manufacturing Processes*, vol. 25, no. 8, pp. 808–816, 2010.
- [31] F. Yin, H. Mao, and L. Hua, "A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters," *Materials and Design*, vol. 32, no. 6, pp. 3457–3464, 2011.
- [32] G. Chen, K. Fu, Z. Liang et al., "The genetic algorithm based back propagation neural network for MMP prediction in CO<sub>2</sub>-EOR process," *Fuel*, vol. 126, pp. 202–212, 2014.
- [33] Y. Q. Chen and A. Chrzanowski, "An overview of the physical interpretation of deformation measurements," in *Proceedings of the Deformation Measurements Workshop*, pp. 207–220, MIT, Cambridge, Mass, USA, 1986.
- [34] Z. Wu, *Thery of Safety Monitoring of Hydraulic Constructions and Its Applications*, Publishing House of Hohai University, Nanjing, China, 2003.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

