

Research Article

Integrated Modeling and Intelligent Control Methods of Grinding Process

Jie-sheng Wang,^{1,2} Na-na Shen,¹ and Shi-feng Sun¹

¹ School of Electronic and Information Engineering, University of Science & Technology Liaoning, Anshan 114044, China

² National Financial Security and System Equipment Engineering Research Center, University of Science & Technology Liaoning, Anshan 114044, China

Correspondence should be addressed to Jie-sheng Wang; wang.jiesheng@126.com

Received 17 April 2013; Accepted 3 September 2013

Academic Editor: Jianming Zhan

Copyright © 2013 Jie-sheng 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.

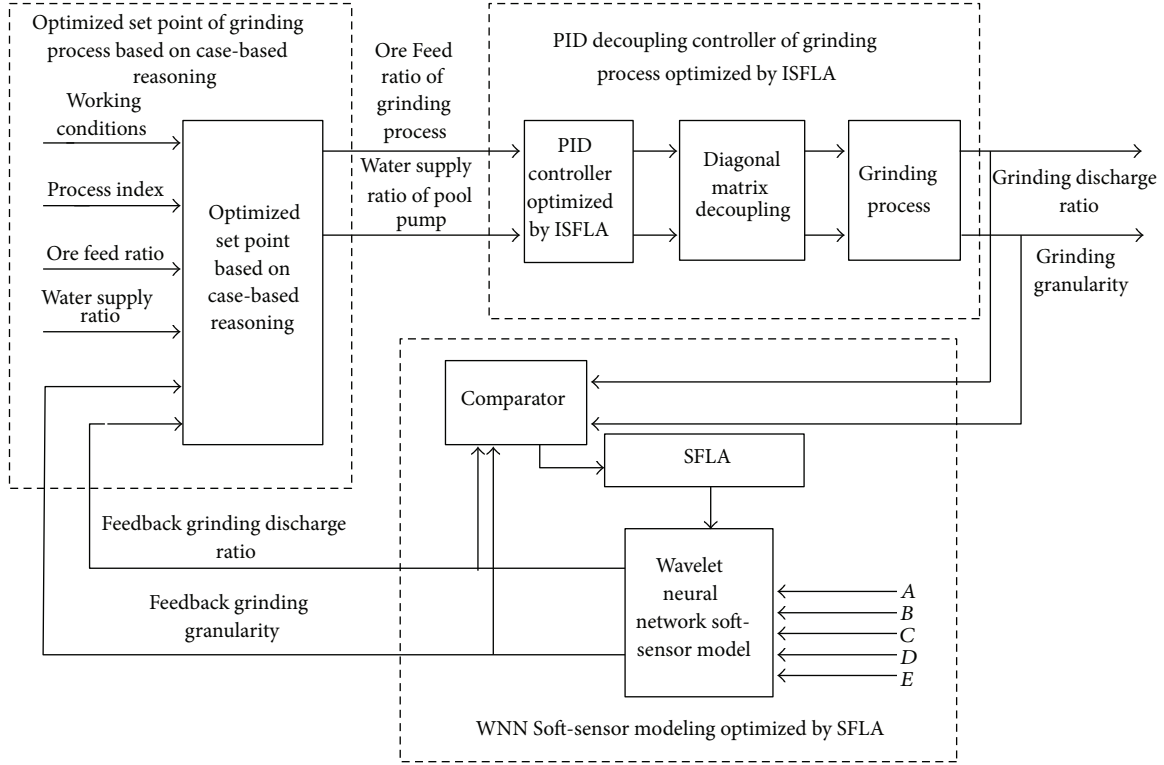
The grinding process is a typical complex nonlinear multivariable process with strongly coupling and large time delays. Based on the data-driven modeling theory, the integrated modeling and intelligent control method of grinding process is carried out in the paper, which includes the soft-sensor model of economic and technique indexes, the optimized set-point model utilizing case-based reasoning, and the self-tuning PID decoupling controller. For forecasting the key technology indicators (grinding granularity and mill discharge rate of grinding process), an adaptive soft-sensor modeling method based on wavelet neural network optimized by the improved shuffled frog leaping algorithm (ISFLA) is proposed. Then, a set point optimization control strategy of grinding process based on case-based reasoning (CBR) method is adopted to obtain the optimized velocity set-point of ore feed and pump water feed in the grinding process controlled loops. Finally, a self-tuning PID decoupling controller optimized is used to control the grinding process. Simulation results and industrial application experiments clearly show the feasibility and effectiveness of control methods and satisfy the real-time control requirements of the grinding process.

1. Introduction

Grinding process has complex production technique and many influencing factors, such as the characteristics of the ore fed into the circuit (ore hardness, particle size distribution, mineral composition, or flow velocity), the flow velocity of water fed into the loops, and the changes of the cyclone feed ore. Grinding process is a serious nonlinear, strong coupling, and large time delay industrial production process. Obtaining the optimal control results by the traditional control method is difficult. Scholars at home and abroad have carried out many advanced control strategies for the grinding process, such as fuzzy control [1–3], neural network control [4], soft sensor modeling [5–8], and other advanced control technology [9–12]. Reference [3] proposed a multivariable fuzzy supervisory control method composed by the fuzzy supervisor, loop precedent set-point model, and the particle size soft-sensor model. Reference [4] studied the grinding process with non-linear, multivariable, time varying parameters, boundary conditions, and fluctuations complex features

and proposed an integrated intelligent model for dynamic simulating, of the grinding and classification process.

Because of the limitations of the industrial field conditions and a lack of mature detectors, the internal parameters (particle size and grinding mills discharging rate) of the grinding process is difficult to obtain the real-time quality closed-loop control directly. The soft-sensing technology can effectively solve the predictive problem of the online measurement of the quality indices. Therefore, the soft-sensor model according to the auxiliary variables can be set up in order to achieve the particle size and grinding mills discharging rate for the real-time forecasting and monitoring, which has great significance on improving the grinding process stability and energy conservation. Domestic scholars have proposed many soft-sensor models, such as neural network model [5–7] and the case-based reasoning technology [8]. Combining the actual working conditions of the grinding classification process of [5] proposed a RBFNN-based particle size soft-sensor model. Reference [6] introduced a grinding size neural network soft-sensor model and



A: Water supply
 B: New ore feed
 C: Divide overflow concentration
 D: Grinding current
 E: Grader current

FIGURE 2: System configuration of the integrated modeling and intelligent control methods of grinding process.

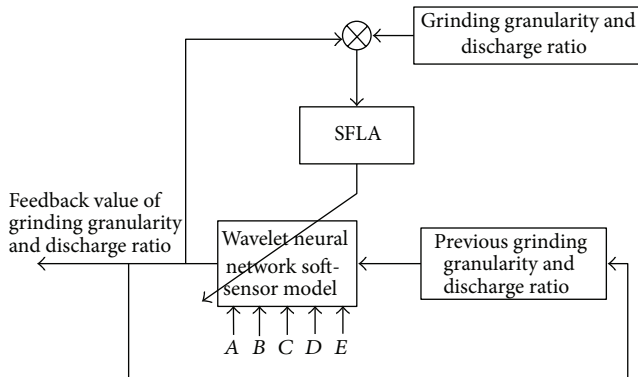


FIGURE 3: Soft-sensor model structure of grinding process.

ore feed, C is the concentration of sub-overflow, D is the milling current and E is the grading machine power. For the key process indicators of grinding process (feedback grinding granularity and the discharge rate), the two multi-input single-output wavelet neural network soft-sensor model is set up. (1) Input variables are A , B , C , D , E , and the previous moment of grinding granularity. Grinding granularity is

output for the feedback. (2) Input variables are A , B , C , D , E , and the previous moment milling discharge ratio. The discharge ratio is output for the feedback. The differences between the predictive values and the actual values are used to optimize the parameters of wavelet neural network through the improved shuffled frog leaping algorithm.

Considering a multi-input single-output (MISO) system, the training sample set can be expressed as $D = \{Y, X_i \mid i = 1, 2, \dots, m\}$. Y is the output variable. X_i represents the i th input vector and can be expressed as $X_i = [x_{1i}, x_{2i}, \dots, x_{ni}]'$ (n is the number of samples in the training set and m is the number of input variables). Soft-sensing modeling requires a datum set from the normal conditions as the modeling data. Assume that the system has m process variable and n data vectors composing the test sample datum matrix $X \in R^{n \times m}$. In order to avoid the different dimensions of the process variables affecting the results and obtain the easy mathematical treatment, it is necessary to normalize the datum. Set μ is the mean vector of X , and σ is the standard deviation vector of X . So, the normalized process variable is expressed as follows:

$$\hat{X} = \frac{X - \mu}{\sigma}. \quad (1)$$

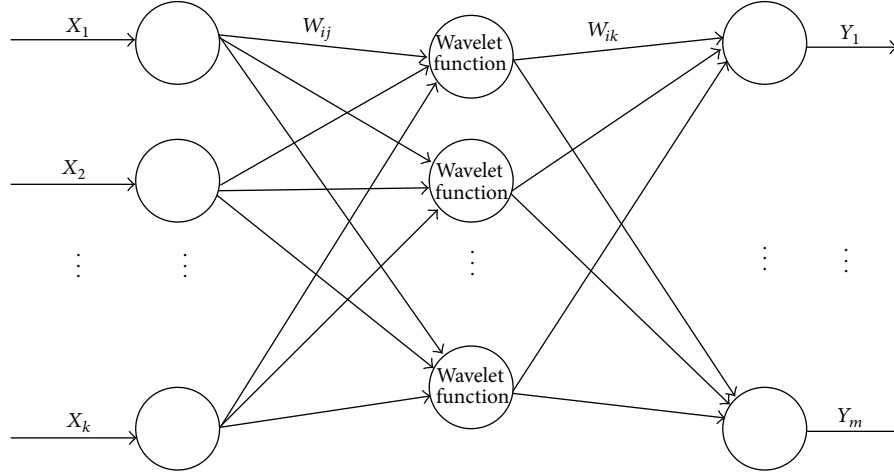


FIGURE 4: Structure of wavelet neural network.

The input vector \hat{X} of the training samples is fed into the wavelet neural network to predict the output \hat{Y} . The root mean square error (RMSE) is selected as the fitness of the WNN soft-sensor model:

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^n (\hat{Y}_k - Y_k^*)^2}{n}}. \quad (2)$$

3.2. Wavelet Neural Network. Wavelet neural network (WNN) is similar to BP neural network with the same topology, which adopts the wavelet base function as the transfer function of hidden layer nodes [14]. Its structure is shown in Figure 4.

In Figure 4, x_1, x_2, \dots, x_k is the input parameters of the wavelet neural network, Y_1, Y_2, \dots, Y_m is the prediction output of the wavelet neural network, and ω_{ij} and ω_{jk} are the weights of the wavelet neural network. When the input signal sequence is x_i ($i = 1, 2, \dots, k$), the output of the hidden layer is calculated as follows:

$$h(j) = h_j \left[\frac{\sum_{i=1}^k \omega_{ij} x_i - b_j}{a_j} \right], \quad j = 1, 2, \dots, l, \quad (3)$$

where $h(j)$ is the j th node output of the hidden layer, ω_{ij} is the connection weights between input layers and hidden layers, b_j is the translation factor of the wavelet base function h_j , a_j is the stretching factor of the wavelet basis function h_j , and h_j is the wavelet function. The morlet wavelet function is adopted in this paper, which is represented as follows:

$$y = \cos(1.75x) e^{-x^2/2}. \quad (4)$$

The parameters of output layers of the wavelet neural network are calculated as

$$y(k) = \sum_{i=1}^l \omega_{ik} h(i), \quad k = 1, 2, \dots, d, \quad (5)$$

where ω_{ik} is weight for the hidden layer to output layer, $h(i)$ is the i th output in the hidden layer, l is the number of

the hidden layer nodes, and d is the number of the input layer nodes.

Standard wavelet neural network uses the gradient descent method to train the structural parameters. But the inherent characteristics of gradient descent method make the WNN training process convergence slow, easy to fall into local minimum, and easily lead to oscillation effect [15]. Therefore, the paper adopts the improved shuffled frog-leap algorithm to train WNN.

3.3. Improved Shuffled Frog Leaping Algorithm. Shuffled frog leap algorithm [16] (SFLA) is a population-based heuristic cooperative swarm intelligent search algorithm. SFLA adopts the metaheuristic algorithm based on swarm intelligence to solve the combinatorial optimization problems, which is based on the meme evolution of the individuals in the population and global information exchange of the memes. SFLA combines the advantages of the genetic-based memetic algorithm (MA) and particle swarm optimization (PSO) with foraging behaviors of the population, such as simple concept, few parameters, quick calculation speed, global optimization capability, easy to implement features [17]. SFLA has been successfully applied in many fields, such as water network optimization problems [16], placement sequence optimization [18, 19], flow shop scheduling problem [20], clustering [21], and so forth.

SFLA is an evolutionary computation algorithm combining deterministic method and stochastic method. Deterministic algorithm can make effective use of strategic information to guide the search response and the random element to ensure the flexibility and robustness of the algorithm searching patterns. The SFLA is described in detail as follows. First, an initial population of N frogs $P = \{X_1, X_2, \dots, X_N\}$ is created randomly. For S -dimensional problems (S variables), the position of a frog i in the search space is represented as $X_i = [x_{i1}, x_{i2}, \dots, x_{iS}]$. After the initial population is created, the individuals are sorted in a descending order according to their fitness. Then, the entire population is divided into m memplexes, each containing n frogs (i.e., $N = m \times n$), in

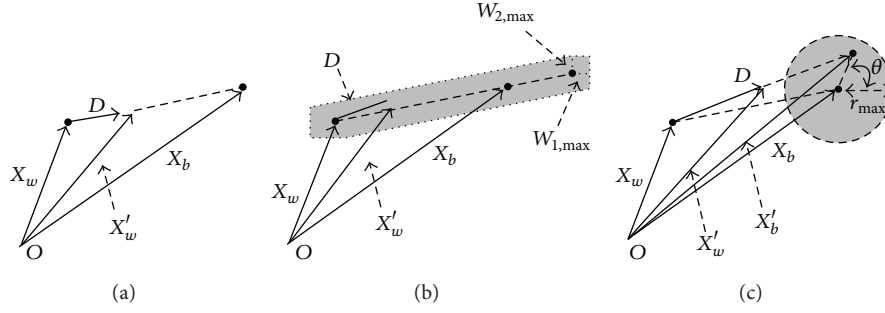


FIGURE 5: Frog leaping rules.

such a way that the first frog belongs to the first memplex, the second frog goes to the second memplex, the m th frog goes to the m th memplex, and the $(m + 1)$ th frog goes back to the first memplex, so forth. Let M^k is the set of frogs in the k th memplex, this dividing process can be described by the following expression:

$$M^k = \{X_{k+m(l-1)} \in P \mid 1 \leq l \leq n\}, \quad (1 \leq k \leq m). \quad (6)$$

In each memplex, the frogs with the best fitness and worst fitness are identified as X_b and X_w . The frog with the global best fitness in the population is identified as X_g . Then, the local searching is carried out in each memplex; that is to say, the worst frog X_w leaps towards the best frog X_b according to the original frog leaping rules (shown in the Figure 5(a)) described as follows:

$$\begin{aligned} D &= r \cdot (X_b(t) - X_w(t)), \\ X'_w(t) &= X_w(t) + D, \quad (\|D\| \leq D_{\max}), \end{aligned} \quad (7)$$

where r is a random number between 0 and 1 and D_{\max} is the maximum allowed change of frog's position in one jump. If the new frog X'_w is better than the original frog X_w , it replaces the worst frog. Otherwise, X_b is replaced by X_g and the local search is carried out again according to formula (7). If no improvement is achieved in this case, the worst frog is deleted and a new frog is randomly generated to replace the worst frog X_w . The local search continues for a predefined number of memetic evolutionary steps L_{\max} within each memplex, and then the whole population is mixed together in the shuffling process. The local evolution and global shuffling continue until convergence iteration number G_{\max} arrives.

3.4. Improved Frog Leaping Rule. During the natural memetic evolution of the frogs, the worse frog is affected by the better frog to leap for the better one in order to get more food. According to the above description of the initial frog leaping rule (shown in the Figure 5(a)), the likely position of the worst frog is limited to the line segment between the current value and the position of the best frog. So this frog-leaping rule limits the search scope of memetic evolution which not only reduces the convergence velocity but also easily leads to the premature convergence. A modified shuffled frog leaping

algorithm [22] based on a new frog leaping rule (shown in Figure 5(b)) can be expressed as follows:

$$D = r \cdot c \cdot (X_b - X_w) + W, \quad (8)$$

$$W = [r_1 w_{1,\max}, r_2 w_{2,\max}, \dots, r_S w_{S,\max}]^T,$$

$$X'_w = \begin{cases} X_w + D, & \text{if } \|D\| \leq D_{\max}, \\ X_w + \frac{D}{\sqrt{D^T D}} D_{\max}, & \text{if } \|D\| > D_{\max}, \end{cases} \quad (9)$$

where r is a random number of $[0, 1]$, c is a constant of $[1, 2]$, r_i ($1 \leq i \leq S$) is a random number of $[-1, 1]$, and $w_{i,\max}$ ($1 \leq i \leq S$) is the maximum perceptual and the movement uncertainty of the i th search space.

This frog leaping rule increases the algorithm search scope in a certain degree. Combined with the characteristics of SFLA, the paper puts forward a new frog leaping rule (shown in the Figure 5(c)) described as follows:

$$\begin{aligned} X'_b &= X_b + r_1 \cdot r_{\max} e^{j\theta}, \\ D &= r_2 \cdot (X'_b - X_w), \end{aligned} \quad (10)$$

where θ is a random angle of $[0, 360^\circ]$, r_{\max} is the maximum local search radius, r_1 and r_2 are random numbers of $[0, 1]$, and r_{\max} defines the maximum perceptual surrounding the local optimization value of the frog memetic groups. θ , r_1 , and r_2 decide the uncertainty of frog leaping. The position vector is still updated by using formula (9).

3.5. Algorithm Procedure of Optimization of WNN Soft-Sensor Model Based on ISFLA. Two wavelet neural network soft-sensor models optimized by the improved SFLA are set up in the paper for predicting the grinding granularity and grinding discharge ratio. The algorithm procedure of ISFLA-based WNN soft-sensor model is shown in Figure 6.

Combined with the proposed new frog leaping rule, the algorithmic procedure of the ISFLA-based wavelet neural network training is described as follows.

Step 1 (initialize the SFLA parameters). Initialize the frog population size N , the search space dimension S , the number of meme groups is m (each meme group contains n frogs) ($N = m \times n$), the allowed frog leaping maximum step D_{\max} ,

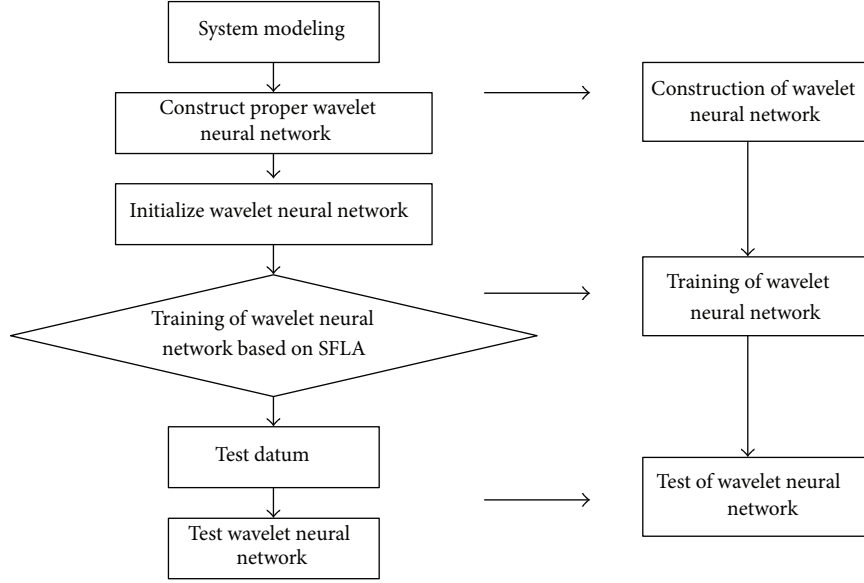


FIGURE 6: Algorithm procedure of optimization of WNN soft-sensor model based on ISFLA.

the local search number L_{\max} and the global hybrid iteration number G_{\max} , and maximum local search radius r_{\max} .

Step 2 (frog population creation). Randomly initial the population of N frogs $P = \{X_1(t), \dots, X_k(t), \dots, X_N(t)\}$ ($k = 1, \dots, N$). Set the iteration counter $t = 0$. Each frog $X_k(t)$ is set as the structure parameters of the wavelet neural network soft-sensor model (wavelet stretch factor a_k , translation factor b_k , and the network connection weights ω_{ij} and ω_{jk} , $i = 1, \dots, k$, $j = 1, \dots, l$, $k = 1, \dots, d$). Then, the training sample datum is fed into the wavelet neural network to carry out the precedent calculation according to the formula (3)–(5). Each individual's fitness value $F_k(t) = F(X_k(t))$ is calculated according to the formula (2) after the simulation. Finally, the frogs are sorted in a descending order according to their fitness. The outcome is stored with the style $U_k(t) = \{X_k(t), F_k(t)\}$. The global best frog in the frog population is identified as $X_g(t) = U_1(t)$.

Step 3 (memeplex creation). The U is divided into the m memeplex $Y_1(t), Y_2(t), \dots, Y_m(t)$ ($j = 1, \dots, m$) according to formula (6). Each memeplex includes n frogs. The frogs with the best fitness and worst fitness in the memeplex are identified as $X_b^j(t)$ and $X_w^j(t)$.

Step 4 (memeplex evolution). The frog in the memeplex $Y_j(t)$ ($j = 1, \dots, m$) is assigned a probability value $P_{jk} = 2(n + 1 - k)/(n(n + 1))$, ($k = 1, \dots, n$). Set the random value $r \in [P_{j1}, P_{jn}]$. If $P_{jk} < r$, the k th frog in the j th memeplex evolves in accordance with formula (10), and the objective function value of the new frog is calculated. If $P_{jk} > r$, the evolution will be given up.

If the frog doesn't achieve the meme evolution, $X_{jb}(t)$ is substituted by $X_g(t)$ to carry out the local search again. If no improvement is achieved, a new frog is created randomly to substitute the $X_w^j(t)$.

Step 5 (local depth search). The frog in the meme group $Y_j(t)$ ($j = 1, \dots, m$) carries out Step 4 for i times to get the meme group $Y_1(t)', Y_2(t)', \dots, Y_m(t)'$.

Step 6 (memeplex shuffled). The frogs in the iterated memeplex $Y_1(t)', Y_2(t)', \dots, Y_m(t)'$ are mixed together in the shuffling process and identified as $(t + 1) = \{Y_1(t)', Y_2(t)', \dots, Y_m(t)'\}$. In $U(t + 1)$, the frog in the objective function value according to ascending sort will be recorded as the best group of frogs $X_g(t + 1) = U_1(t + 1)$.

Step 7 (test the algorithm termination condition). $t = t + 1$; if $t < G_{\max}$, go to Step 3. Otherwise output the best frog.

3.6. Adaptive Revision of Soft-Sensor Model Based on Model Migration Method. The model migration method [23, 24] based on process similarity is shown in the Figure 7, which is based on the well-established model to develop a new model of the similar process by adopting few datum.

Due to the fluctuations in ore grade and other working conditions of the grinding process, the current soft-sensor results are no longer accurate so that the soft-sensor model must be adaptive corrected. At the moment, a small amount of datum may be adopted to set up a new soft-sensor model based on the model migration (linear correction and planning) from the original soft-sensor model to be adapted to the new working conditions. In this paper, the migration modeling method based on the input-output correction programming method is adopted, whose basic principle is shown in Figure 8.

Assume the original soft-sensor model:

$$Y_{\text{base}} = f(X_{\text{base}}), \quad (11)$$

where X_{base} and Y_{base} are the input and output of the original model.

TABLE 1: Input data set of forecasted grinding granularity.

Number	Water of ore feed (m ³ /h)	New ore feed (T/h)	Divide overflow concentration (%)	Grinding current (A)	Grader current (A)	Grinding granularity (%)
1	20.45	95.13	80.83	55.3	5.1	80.24
2	23.23	110.14	82.08	56.9	5.3	82.97
3	24.27	104.67	83.98	54.3	5.8	86.69
4	25.34	102.87	87.32	55.9	5.2	84.36
⋮	⋮	⋮	⋮	⋮	⋮	⋮
300	23.87	103.09	86.23	55.4	5.6	81.56

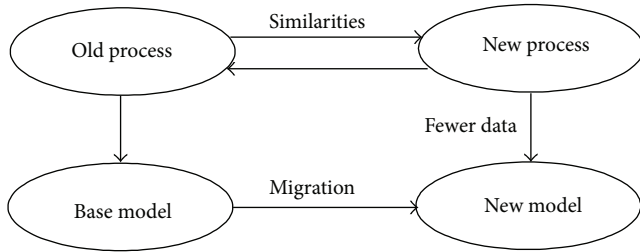


FIGURE 7: Basic principles block diagram of PMBPS.

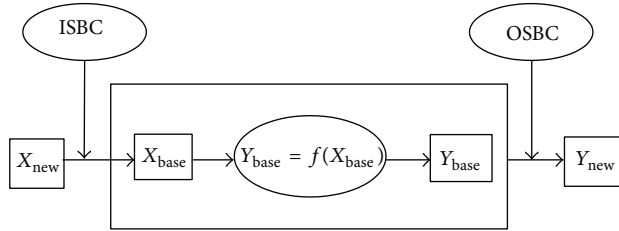


FIGURE 8: IOSBC migration modeling principles chart.

Through the output space migration and plan a new model is obtained as follows

$$Y_{\text{new}} = S_0 f(S_I X_{\text{new}} + B_I) + B_0, \quad (12)$$

where S_0 and B_0 is the scale factor and offset parameters of output space in the original model.

Then, the input space is shifted and revised. The input X_{new} of the new model can be obtained by the input bias correction of the original model input X_{base} , which is described as follows

$$X_{\text{base}} = S_I X_{\text{new}} + B_I, \quad (13)$$

where S_I and B_I are the scale factor and offset parameters of the input space, respectively.

Therefore, the new model is obtained by the input-output offset correction of the original model, which is described as follows:

$$Y_{\text{new}} = S_0 f(S_I X_{\text{new}} + B_I) + B_0. \quad (14)$$

New sample datum can be used to train the correct parameters:

$$\begin{aligned} \text{Min } J(S_0, B_0, S_I, B_I) &= ee^T \\ \text{st. } e_i &= y_i - [S_0 f(S_I X_{\text{new},i} + B_I) + B_0], \end{aligned} \quad (15)$$

where y_i is the i th observation data of the new process, e represents the prediction error between the measurement and the prediction value of the new model. The dimension of the identified amendments and planning parameters is determined by the input-output space dimension.

3.7. Simulation Results. Aiming at the grinding and classification process, the grinding granularity and grinding discharge ratio soft-sensor model is set up based on the wavelet neural network. Firstly, the input-output data set is shown in Table 1 in order to train and test the ISFLA-based WNN soft-sensor model. The precedent 260 group data comes from the same working condition. The later 40 group data comes from another dynamic working condition due to the variation of the ore feed grade in order to verify the adaptive performance of the oft-sensor model. The first 200-group data was used to train the wavelet neural network by the ISFLA and gradient descent method. The later 100-group data was adopted to carry out the soft-sensor model validation. The predictive results of the validation data by the proposed soft-sensor model illustrated in Figures 9 and 10.

Usually the average relative variance (ARV) [1] is adopted to measure the difference between the predicted value and the measured value, which is defined as follows:

$$\text{ARV} = \frac{\sum_{i=1}^N [x(i) - \hat{x}(i)]^2}{\sum_{i=1}^N [x(i) - \bar{x}]^2}, \quad (16)$$

where N is the number of comparative data, $x(i)$ is the measurement value, \bar{x} is the average of the measurement values, and $\hat{x}(i)$ is the predictive value. Obviously, the smaller the average relative variance, the better the predictive performance. $\text{ARV} = 0$ means that the model has an ideal prediction. $\text{ARV} = 1$ indicates that the model only obtains the average prediction results. The contrast results of the AVR values under the WNN soft-sensor model and the ISFLA-based WNN soft-sensor model are listed in Table 2.

As seen from Figures 9 and 10 and Table 2, the WNN adaptive soft-sensor model optimized by the improved shuffled frog-leaping algorithm (ISFLA) of the grinding process

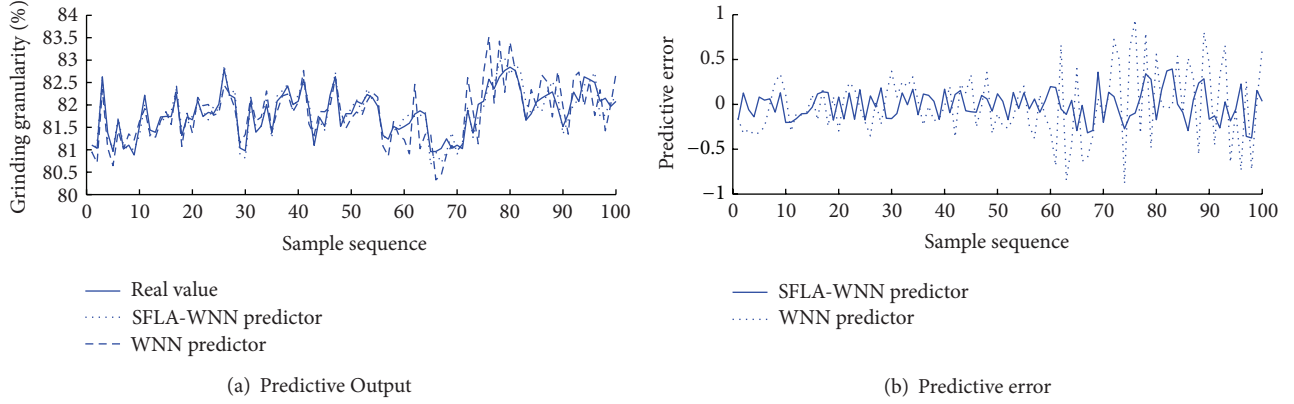


FIGURE 9: Predictive output of grinding granularity.

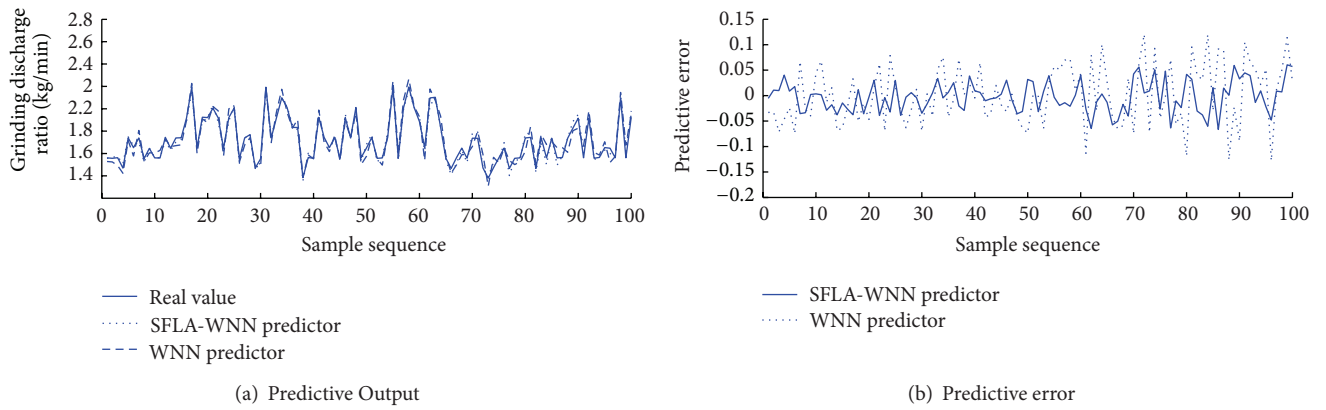


FIGURE 10: Predictive output of mill discharge rate.

TABLE 2: Predictive AVR.

	SFLA-WNN	WNN
AVR of grinding granularity	0.1183	0.5700
AVR of grinding discharge ratio	0.0212	0.0790

for predicting the key technique indicators (grinding granularity and milling discharging ratio) has higher prediction accuracy and generalization ability than those of the standard wavelet neural network soft-sensor model. The proposed ISFLA can effectively adjust the structure parameters of the WNN soft-sensor model. On the other hand, when the working condition of the grinding process changes, the soft-sensor model can be corrected adaptively based on the model migration strategy, which results in the more accurate predictions.

4. Set-Point Optimization of Grinding Process Based on Case Reasoning

4.1. Basic Flowchart of Case-Based Reasoning. The general procedure of the case-based reasoning process includes

retrieve-reuse-revise-retain. In the CBR process, the case retrieval is the core of CBR technology, which directly determines the speed and accuracy of decision making. The basic procedure of the case-based reasoning technology [25, 26] is shown in Figure 11.

The case-based reasoning process is mainly divided into four basic steps [27]: (1) case retrieval: by a series of searching and similarity calculation, the most similar case with the current problem is found in the case database. (2) Case reuse: compare the differences between the source case and the target case. The solution case recognized by the user will be submitted to the user, and the effect of its application will be observed. (3) Case revision: the solution strategy of the retrieval case is adjusted by combining the effect of case reuse and the current issue in order to fit the current problem. (4) Case storage: the current issue is resolved and stored in the case database for the future use.

4.2. Set-Point Optimization Strategy of Grinding Process. Grinding process is a complex nonlinear industrial controlled object. Combining the real problems that exist in grinding process control with the theory of case-based reasoning, the basic procedure of the set point optimization strategy is shown in Figure 12 [25]. By carrying out a comprehensive

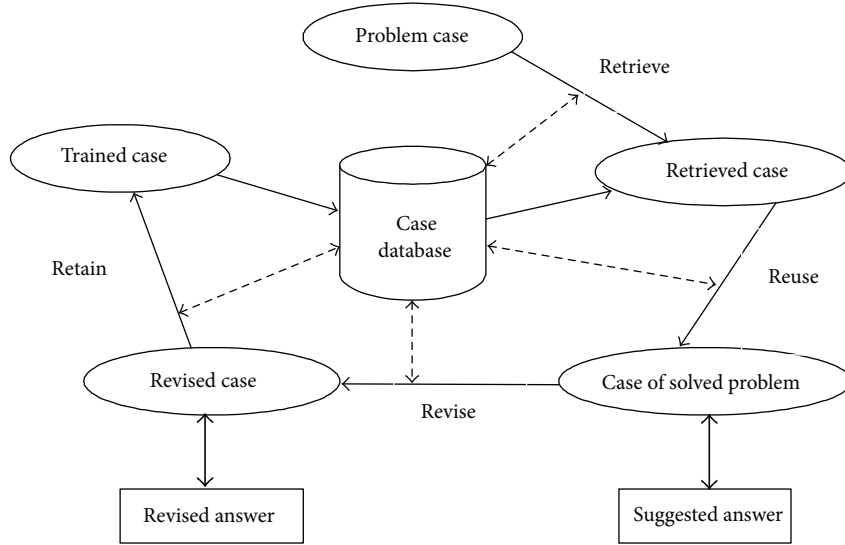


FIGURE 11: Basic flowchart of case-based reasoning.

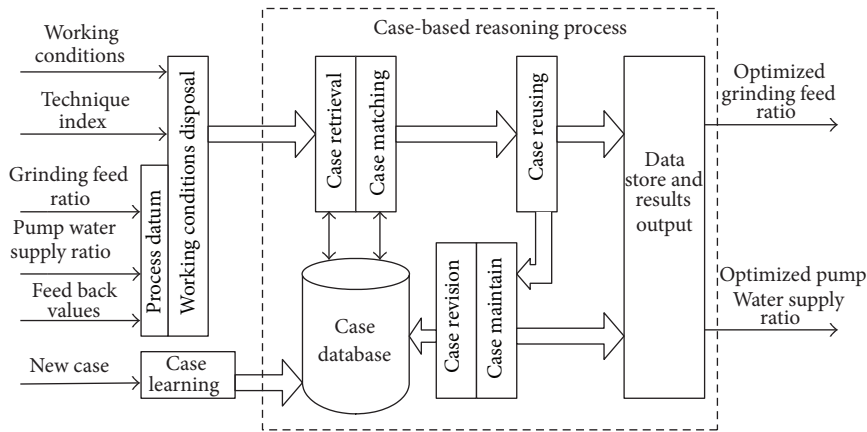


FIGURE 12: Diagram of the grinding intelligent set-point control based on case-based reasoning.

analysis and case-based reasoning for the complex process, the intelligent set-point of the grinding feed ratio and pump water supply ratio are obtained in an optimized manner.

The basic procedure is described as follows. Firstly, the working conditions, the process indicators, and the process datum are dealt with for the case reasoning. Then, the case retrieval and case matching are carried out for obtaining the matched case. If the matched case is not obtained, the new case will appear and be studied and stored into the database. Thirdly, the matched case will be reused and corrected. Finally, maintain the case database, output the results, and store the datum.

4.3. Case Description. The most commonly used knowledge representation methods have production rules, semantic networks, frames, decision trees, predicate logic and fuzzy relations, and so forth. In theory, the form that knowledge is represented by in the case is not a new knowledge representation method, but it is an abstract knowledge representation

based on the past ones, which means that the case is a logical concept. The case must be based on the existing variety knowledge representation methods. That is to say that almost all the existing knowledge representation methods can be used as the implementation of the case representation. A typical case generating process is essentially refinement process of case databases. It represents a large number of similar cases and experiences in common and can reduce not only the retrieval process in the selected set of objects but also other parts of the analog process the workload.

The case model in the CBR process is described as $C_k = [T_k, F_k, J_k]$. C_k means there are k cases in total, $k = (1, 2, \dots, n)$. T_k represents the time at which the case occurs. $F_k = (f_{1k}, f_{2k}, \dots, f_{5k})$ expresses the characteristics of what C_k describes. f_{1k} is the working conditions of industrial process, f_{2k} is the process indicators, f_{3k} is the grinding ore feed ratio, f_{4k} is the pump water feed velocity, and f_{5k} denotes the feedback amount. $J_k = (j_1, j_2, \dots, j_n)$ expresses the characteristics solutions of the case C_k .

4.4. Case Retrieval and Matching. Case matching and case retrieval are important steps in the case-based reasoning process and the key of the information extraction from the case databases. In general, the case matching strategy includes the serial and parallel search methods. In the serial search process, the cases are organized with the hierarchical manner. The top-down refinement layer by layer retrieval approach is adopted, which means the more down the layer, the higher the similarity. The parallel searching strategy weakens the level features among the cases. The retrieve method is to return to the most similar case by retrieving many cases. The commonly used search strategies have nearest neighbor strategy, inductive reasoning strategy, and knowledge guidance strategy.

If the current working condition is N , the similarity degree between the description features f_i ($i = 1, \dots, n$) of N , and the description features f_{ik} of the case is described as follows:

$$\text{sim}(f_i, f_{ik}) = \begin{cases} 1, & f_i = f_{ik}, \\ 0, & f_i \neq f_{ik}. \end{cases} \quad (17)$$

The similarity function between N and C_k is described as follows:

$$\text{sim}(N, C_k) = \sum_{i=1}^n c_i \text{sim}(f_i, f_{ik}), \quad (18)$$

where c_i is the feature weight.

So the static similarity threshold adopted in the paper is described as follows:

$$\text{sim}_{kj} = \begin{cases} X_{kj}, & \text{sim}_{\max} \geq X_{kj}, \\ \text{sim}_{\max}, & \text{sim}_{\max} < X_{kj}, \end{cases} \quad (19)$$

where X_{kj} is 0.9 and sim_{kj} is similarity threshold.

4.5. Case Reuse. In the actual production process, because the case library does not have a case fully matching with the current work under normal circumstances, the retrieved solution parameters matching the working conditions are not directly selected as the control parameters of the current conditions. Therefore, the similar case solution retrieved must be reused. That is to say that the CBR system will adjust the retrieved case solution according to the specific circumstances of the new case to get the solution of the input new case. The adjustment strategy uses the existing process knowledge to obtain the current working parameters based on the differences between the input case working conditions and the retrieval case working conditions. The SSTD-based case reuse strategy is described as follows:

$$Q = \begin{cases} Q', & (\text{sim}_{\max} = 1) \vee (\text{sim}_{kj} = \text{sim}_{\max}), \\ \frac{\sum_{k=1}^H (\text{sim}_k \times J_k)}{\sum_{k=1}^H \text{sim}_k}, & \text{Other}, \end{cases} \quad (20)$$

where sim_k is the similarity property value of the case and Q' is the solution of the maximum sim_{\max} .

4.6. Case Correction. In order to prove the effectiveness of case reuse, the case must be amended in the new implementation process in order to form the effective feedback. Usually the case model obtained by the case-based reasoning can be directly applied to the current working conditions. However, due to some differences between the current working conditions and the retrieved cases in some characteristics, the retrieved cases cannot be directly used in the current conditions. The retrieved case must be amended to adapt to the current issue.

The main contents of the case correction mainly include the amendments of the case features and structures. If $\text{sim}_{\max} < E_i$, E_i is identified by the experts, the case $C_k' = [T_k', F_k', J_k']$ is added into the new case database in order to make the case database be constantly updated.

4.7. Case Maintenance. Case storage strategy is to store the new cases and their solutions into case database according to a certain strategy. The case storage is the base of the case library. By doing so, the case database keeps growing and expanding to increase the searching scope of the case database. At the same time, the maintenance of case database has become an essential work. In the case of storage, the new case $C_k' = [T_k', F_k', J_k']$ very similar to the case in the database, that is to say the similarity is 1, is not stored. This is to simplify the case database and reduce the maintenance time.

5. PID Decoupling Controller Based on ISFLA

The paper mainly studies the relationship between the input variables (grinding ore feed ratio and pump water feed velocity) and output variables (grinding granularity and grinding discharge ratio). Through experiments, the dynamic process model of the grinding circuit includes the ball milling mechanistic model, based on material balance, the empirical model of hydrocyclones, the pump pool hybrid model based on the mechanistic model and empirical model. Through the step response of the grinding process, the system transfer function model [22] is described in formula (21):

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{-0.425e^{-1.52s}}{11.7s + 1} & \frac{0.1052(47.1s + 1)}{11.5s + 1} \\ \frac{2.977}{5.5s + 1} & \frac{1.063e^{-2.26s}}{2.5s + 1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (21)$$

The mathematical model of the grinding process described in formula (21) is decoupled by the diagonal matrix decoupling method. The two control variables are the grinding ore feed ratio U_1 and pump water feed velocity U_2 . The two controlled variables are the overflow mass fraction Y_1 and the grinding discharge ratio Y_2 . The structure of the parameters self-tuning multivariable PID decoupling controller optimized by the ISFLA is shown in Figure 13 [28], which is composed of PID controller and decoupling compensator based on diagonal decoupling method. The parameters of the PID controller are optimized by the improved shuffled frog leaping algorithm.

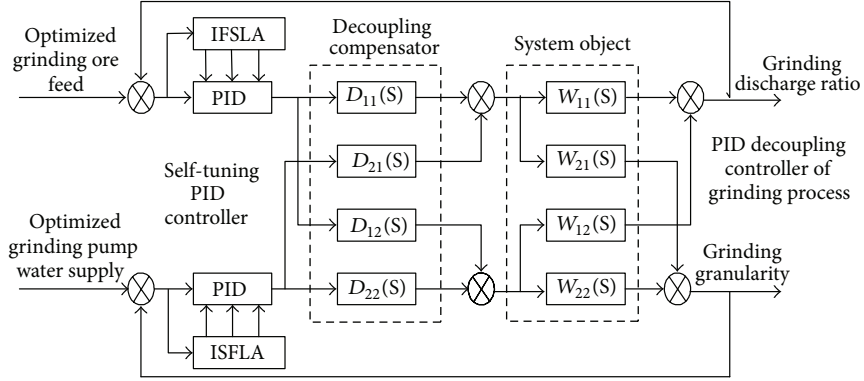


FIGURE 13: Self-tuning PID Decoupling controller.

5.1. *Diagonal Matrix Decoupling.* Diagonal matrix decoupling procedure is described as follows [29]:

$$\begin{aligned} D_{12}(s)W_{11}(s) + D_{22}(s)W_{12}(s) &= 0, \\ D_{11}(s)W_{21}(s) + D_{21}(s)W_{22}(s) &= 0. \end{aligned} \quad (22)$$

Additionally, $M_{c2} \neq 0$ and $M_{c1} \neq 0$. The system may be achieved by decoupling. By using the diagonal matrix

decoupling strategy to obtain the two single-input single-output (SISO) systems,

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \begin{pmatrix} W'_{11} & 0 \\ 0 & W'_{22} \end{pmatrix} = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \quad (23)$$

must be satisfied.

For meeting $W'_{21} = W'_{12} = 0$, the system must satisfy

$$\begin{aligned} \begin{pmatrix} W'_{11} & 0 \\ 0 & W'_{22} \end{pmatrix} &= \begin{pmatrix} W_{11}(s) & W_{12}(s) \\ W_{21}(s) & W_{22}(s) \end{pmatrix} \begin{pmatrix} D_{11}(s) & D_{12}(s) \\ D_{21}(s) & D_{22}(s) \end{pmatrix} \\ &= \begin{pmatrix} W_{11}(s)D_{11}(s) + W_{12}(s)D_{21}(s) & W_{11}(s)D_{12}(s) + W_{12}(s)D_{22}(s) \\ W_{21}(s)D_{11}(s) + W_{22}(s)D_{21}(s) & W_{21}(s)D_{12}(s) + W_{22}(s)D_{22}(s) \end{pmatrix}. \end{aligned} \quad (24)$$

So

$$W'_{11} = W_{11}D_{11} + W_{12}D_{21}, \quad (25)$$

$$W'_{22} = W_{21}D_{12} + W_{22}D_{22}, \quad (26)$$

$$W'_{11} = \frac{0.425}{11.7s+1} \cdot \frac{1.063}{2.5s+1} e^{-3.78s} + \frac{4.95s+0.1052}{11.5s+1} \cdot \frac{2.977}{5.5s+1}, \quad (27)$$

$$W'_{22} = \frac{1.063}{2.5s+1} \cdot \frac{1}{11.7s+1} e^{-3.78s} + \frac{2.977}{5.5s+1} \cdot \frac{4.95s+0.105}{4.89s+0.43}. \quad (28)$$

So obtain two SISO systems: $Y_1 = W'_{11}U_1$ and $Y_2 = W'_{22}U_2$. By decoupling, two separate single-input single-output channels are created to make the direct control of the milling ore feed ratio to milling discharge ratio and the pump water feed velocity to the grinding granularity.

5.2. *PID Controller Optimization Based on ISFLA.* PID control is a regulator based on the linear combination of

the proportion, differential, and integral of the bias, whose transfer function is described as follows [30]:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + \frac{T_d de(t)}{dt} \right], \quad (29)$$

where $K_i = K_p/T_i$, $K_d = K_p T_d$, and $e(t)$ are the feedback bias.

Firstly, the Z-N method is adopted to determine the initial parameters of PID controller. So the two groups of PID controller parameters (K_p , K_i , and K_d) are (6, 35, and 0.1) and (4, 23.2, and 1), respectively. Then with the PID controller parameters achieved by Z-N method as the center, the searching space of the PID parameters is expanded, which is described as follows:

$$\begin{aligned} (1-\alpha)K'_p &\leq K_p \leq (1+\alpha)K'_p, \\ (1-\alpha)K'_i &\leq K_i \leq (1+\alpha)K'_i, \\ (1-\alpha)K'_d &\leq K_d \leq (1+\alpha)K'_d. \end{aligned} \quad (30)$$

Four performance indicators IAE, ITAE, ISE, and ITSE are described in (31). The best performance index is selected through simulation comparison to determine the PID parameters. The target of selecting the performance index is to

obtain the best control performance, such as increasing response speed, decreasing overshoot of the PID controller, and so forth

$$\begin{aligned}
 J_1 &= \text{IAE} = \int_0^{\infty} |e(t)| dt, \\
 J_2 &= \text{ITAE} = \int_0^{\infty} t |e(t)| dt, \\
 J_3 &= \text{ISE} = \int_0^{\infty} e(t)^2 dt, \\
 J_4 &= \text{ITSE} = \int_0^{\infty} t^2 e(t)^2 dt.
 \end{aligned} \tag{31}$$

The original two-input-two-output (TITO) system is decoupled into two independent single-input-single-output (SISO) channels by adopting diagonal matrix decoupling methods. The parameters of the PID controller are optimized by the proposed ISFLA.

5.3. Simulation Research and Industrial Application Experiments. The simulation parameters are selected as follows: frog population size $N = 50$, search space dimension $S = 3$, the number of memplex $m = 5$, the number of frogs in each memplex $n = 10$, the allowed leaping maximum distance $D_{\max} = 0.01$, the number of local search $L_{\max} = 10$, and overall hybrid iteration number $G_{\max} = 100$. Aiming at two SISO models (denoted as M1 and M2) described in Formula (27)-(28), the parameters of the PID controller are tuned by the ZN method and ISFLA under four fitness functions (ISE, IAE, ITAE, and ISTE). By comparing the simulation results under four different fitness functions, the optimization effects can be obtained. The output response curves of M1 and M2 are shown in Figures 14 and 15.

For model M1 and M2, the parameters of the PID controller tuned by a variety of methods are shown in Tables 3 and 4. The performance indexes of the step response, such as overshoot, rise time, and settling time, are adopted to carry out comparison with the different PID controllers, which are illustrated in Tables 5 and 6. As seen from the above simulation results, the PID controllers optimized by the ISFLA under four fitness functions (ISE, IAE, ITAE, and ISTE) achieve better performances than the conventional ZN method, while the PID controller under the fitness function ITAE obtains more superior performances.

Under the premise that the other process variables remain unchanged, the grinding ore feed ratio and pump water feed velocity before and after optimization have a different influence on the performance indexes of the grinding process. Because the ultimate impact factors on the economic efficiency of grinding process are the concentrate grade and tailings grade. So the industrial application experiments are carried out under the proposed data-driven integrated modeling and intelligent control method in the grinding process. The technique indexes controlled scopes are described as follows. Concentrate grade J and tailings grade W are $66\% \leq J \leq 71.5\%$ and $W \leq 28\%$, respectively. The target is to increase the concentrate grade and reduce the tailing grade as

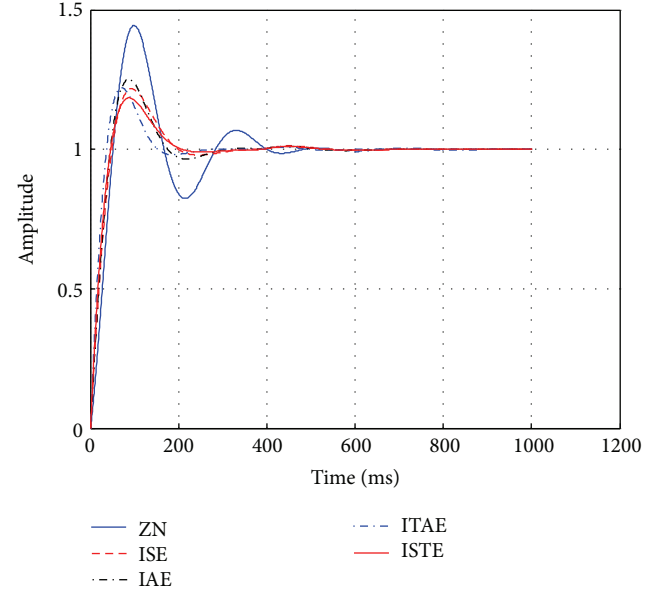


FIGURE 14: Output response curves of M1.

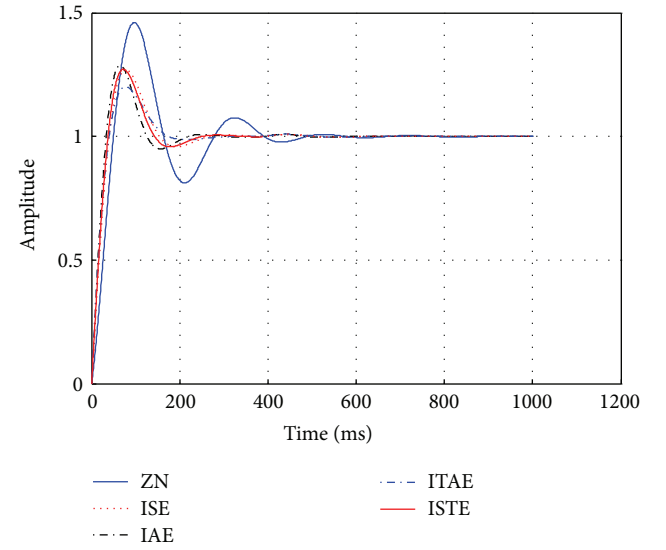


FIGURE 15: Output response curves of M2.

TABLE 3: PID controller parameters of M1.

PID parameters	ZN	ISE	IAE	ITAE	ISTE
K_p	6	13.5	7.8	12.6	12.44
K_i	35	26.3	9.29	14.52	28.84
K_d	0.1	0.2	0.38	0.38	0.38

much as possible. As the concentrate grade and tailings grade can directly determine the effect of optimized controller, the experiments results are shown in Figures 16–19, which includes 200 groups datum (concentrate grades and tailings grades before and after optimization controller).

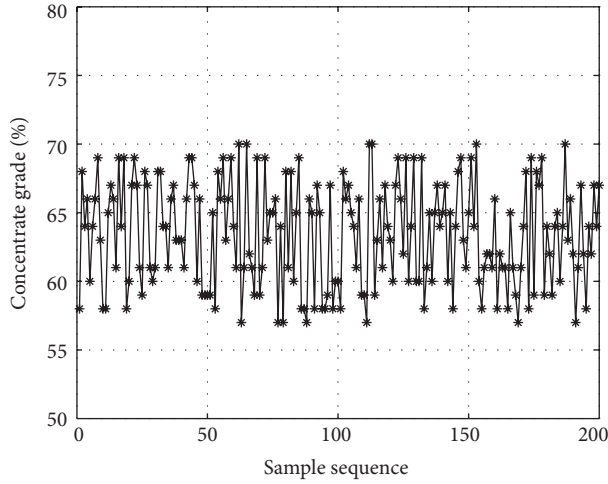


FIGURE 16: Concentrate grade before intelligent optimization.

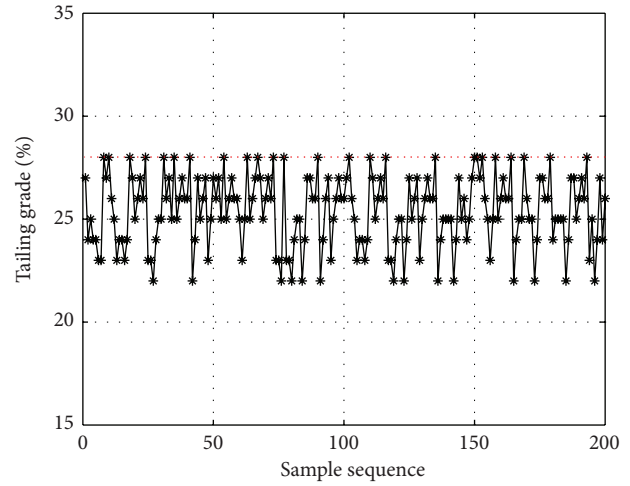


FIGURE 19: Tailings of grade after intelligent optimization.

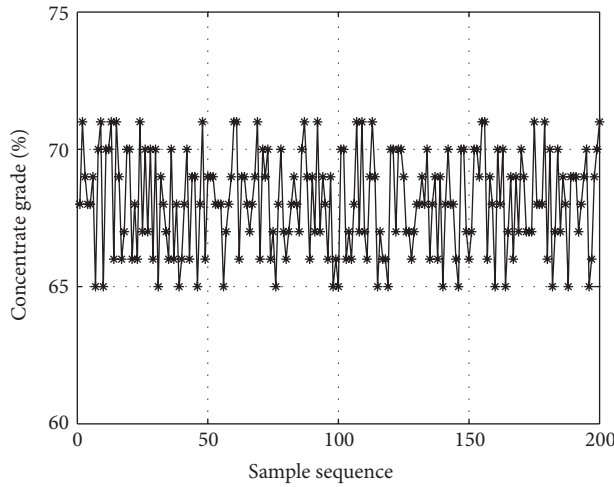


FIGURE 17: Concentrate grade after intelligent optimization.

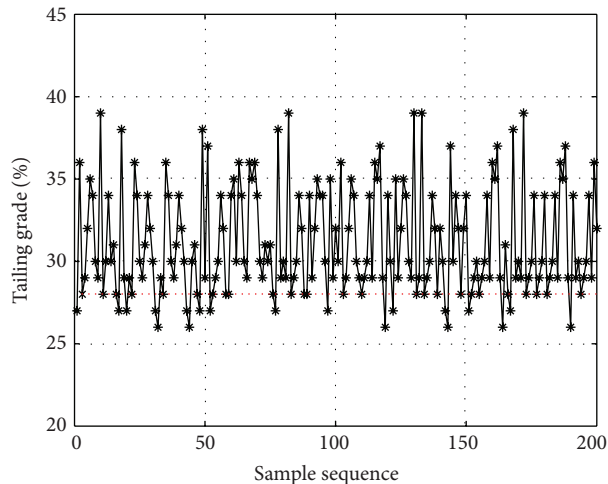


FIGURE 18: Tailings grade before intelligent optimization.

TABLE 4: PID controller parameters of M2.

PID parameters	ZN	ISE	IAE	ITAE	ISTE
K_p	4	6.89	6.85	8.64	4.95
K_i	23.2	14.15	12.22	4.64	10.37
K_d	1	0.67	0.74	1.05	0.73

TABLE 5: Output response performance index of M1.

Performance Index	ZN	ISE	IAE	ITAE	ISTE
Overshoot	48.1	25.1	27.8	25.1	23.2
Rising time	0.78	0.69	0.71	0.73	0.72
Regulated time	4.2	2.5	2.7	2.4	2.5

TABLE 6: Output response performance index of M2.

Performance index	ZN	ISE	IAE	ITAE	ISTE
Overshoot	48.9	25.3	26.2	20.1	25.1
Rising time	0.79	0.74	0.75	0.72	0.73
Regulated time	4.5	2.5	2.6	2.3	2.6

As seen from Figures 16 and 17, the performances under the intelligent optimized controller are better than those under the conventional controller, such as the lower fluctuate of the concentrate grade and the higher concentrate grade, which indicates that the proposed data-driven integrated modeling and intelligent control strategy helps to increase the product quality. As seen from Figures 18 and 19, the performances under the intelligent optimized controller are better than those under the conventional controller, such as the lower fluctuate of the tailings grade and the lower tailings grade, which indicates that the proposed data-driven integrated modeling and intelligent control strategy helps to increase the resources utilization rate.

6. Conclusions

For the grinding process, a complex industrial controlled object, an integrated automation, and control system are researched in detail, which includes the economic and technical indicators of soft-sensor model, the set-point optimized model based on the case-based reasoning methods, and the self-tuning PID decoupling controller. Simulation and industrial experimental results show that the proposed data-driven integrated modeling and intelligent control methods have a better feasibility and effectiveness to meet the real-time control requirements of the grinding production process.

Acknowledgments

This work is partially supported by the Program for China Postdoctoral Science Foundation (Grant no. 20110491510), the Program for Liaoning Excellent Talents in University (Grant no. LJQ2011027), and the Program for Research Special Foundation of University of Science and Technology of Liaoning (Grant no. 2011zx10).

References

- [1] X.-G. Wu, M.-Z. Yuan, and H.-B. Yu, "Product flow rate control in ball mill grinding process using fuzzy logic controller," in *Proceedings of the IEEE International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 761–764, July 2009.
- [2] J. Q. Yu, A. M. Xi, and J. H. Fu, "The application of fuzzy adaptive learning control (FALCON) in milling-classification operation system," *Journal of Xi'an University of Architecture & Technology*, vol. 32, no. 2, pp. 175–178, 2000.
- [3] P. Zhou, H. Yue, X.-P. Zheng, and T.-Y. Chai, "Multivariable fuzzy supervisory control for mineral grinding process," *Control and Decision*, vol. 23, no. 6, pp. 685–688, 2008.
- [4] M. Tie, H. Yue, and T.-Y. Chai, "Hybrid intelligent modeling and simulation for ore grinding and classification process," *Journal of Northeastern University*, vol. 28, no. 5, pp. 609–612, 2007.
- [5] X. Zhang, W. Wang, and X. Wang, "Research of the particle size neural network soft sensor for concentration process," *Control Theory and Applications*, vol. 19, no. 1, pp. 85–88, 2002.
- [6] J. Ding, H. Yue, Y. Qi, T. Chai, and X. Zheng, "NN soft-sensor for particle size of grinding circuit based GA," *Chinese Journal of Scientific Instrument*, vol. 27, no. 9, pp. 981–984, 2006.
- [7] G. C. He, Y. P. Mao, and W. Ni, "Grinding size soft sensor model based on neural network," *Metal Mines*, vol. 344, no. 2, pp. 47–49, 2005.
- [8] P. Zhou, H. Yue, D.-Y. Zhao, and T.-Y. Chai, "Soft-sensor approach with case-based reasoning and its application in grinding process," *Control and Decision*, vol. 21, no. 6, pp. 646–655, 2006.
- [9] X. Li and Y. Rong, "Framework of grinding process modeling and simulation based on microscopic interaction analysis," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 2, pp. 471–478, 2011.
- [10] L. E. Olivier, B. Huang, and I. K. Craig, "Dual particle filters for state and parameter estimation with application to a run-of-mine ore mill," *Journal of Process Control*, vol. 22, no. 4, pp. 710–717, 2012.
- [11] J. S. Wang, X. W. Gao, and S. F. Sun, "Data-driven integrated modeling and intelligent control methods of grinding process," in *Advances in Neural Networks*, J. Wang, G. G. Yen, and M. M. Polycarpou, Eds., vol. 7368 of *Lecture Notes in Computer Science*, pp. 396–405, Springer, Berlin, Germany, 2012.
- [12] A. Cortinovis, M. Mercangöz, T. Mathur et al., "Nonlinear coal mill modeling and its application to model predictive control," *Control Engineering Practice*, vol. 21, no. 3, pp. 308–320, 2013.
- [13] J. S. Wang, Y. Yang, and S. F. Sun, "SFLA-WNN soft-sensor modeling and reconfiguration of grinding process based on model migration," *Journal of Shanghai Jiaotong University*, vol. 46, no. 12, pp. 1951–1955, 2012.
- [14] S. Suhartono and S. Subanar, "Development of model building procedures in wavelet neural networks for forecasting non-stationary time series," *European Journal of Scientific Research*, vol. 34, no. 3, pp. 416–427, 2009.
- [15] K. K. Minu, M. C. Lineesh, and C. Jessy John, "Wavelet neural networks for nonlinear time series analysis," *Applied Mathematical Sciences*, vol. 4, no. 49–52, pp. 2485–2495, 2010.
- [16] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.
- [17] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [18] A. Rahimi-Vahed and A. H. Mirzaei, "A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem," *Computers and Industrial Engineering*, vol. 53, no. 4, pp. 642–666, 2007.
- [19] G.-Y. Zhu, "Meme triangular probability distribution shuffled frog-leaping algorithm," *Computer Integrated Manufacturing Systems*, vol. 15, no. 10, pp. 1979–1985, 2009.
- [20] A. Rahimi-Vahed, M. Dangchi, H. Rafiei, and E. Salimi, "A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 41, no. 11–12, pp. 1227–1239, 2009.
- [21] B. Amiri, M. Fathian, and A. Maroosi, "Application of shuffled frog-leaping algorithm on clustering," *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1–2, pp. 199–209, 2009.
- [22] T.-H. Huynh, "A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '08)*, pp. 1–6, April 2008.
- [23] J. Lu and F. Gao, "Process modeling based on process similarity," *Industrial and Engineering Chemistry Research*, vol. 47, no. 6, pp. 1967–1974, 2008.
- [24] J. Lu, K. Yao, and F. Gao, "Process similarity and developing new process models through migration," *AIChE Journal*, vol. 55, no. 9, pp. 2318–2328, 2009.
- [25] J. S. Wang and S. F. Sun, "Optimized set-point model of grinding process based on case-based reasoning method," in *Proceedings of the IEEE International Conference on System Science and Engineering*, pp. 107–111, 2012.
- [26] S. Craw, N. Wiratunga, and R. C. Rowe, "Learning adaptation knowledge to improve case-based reasoning," *Artificial Intelligence*, vol. 170, no. 16–17, pp. 1175–1192, 2006.
- [27] M. Salamó and E. Golobardes, "Rough sets reduction techniques for case-based reasoning," in *Case-Based Reasoning Research and Development*, D. W. Aha and I. Watson, Eds., vol. 2080 of *Lecture Notes in Computer Science*, pp. 467–482, Springer, Berlin, Germany, 2001.

- [28] J.-S. Wang and S.-F. Sun, "Design of self-tuning PID decoupling controller of grinding process," *Journal of System Simulation*, vol. 23, no. 10, pp. 2220–2223, 2011.
- [29] M. I. Menhas, L. Wang, M. Fei, and H. Pan, "Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4390–4401, 2012.
- [30] J.-S. Wang, Y. Zhang, and W. Wang, "Optimal design of PI/PD controller for non-minimum phase system," *Transactions of the Institute of Measurement and Control*, vol. 28, no. 1, pp. 27–35, 2006.

Copyright of Mathematical Problems in Engineering is the property of Hindawi Publishing Corporation and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.