

Research Article

A Novel Adjustment Method for Shearer Traction Speed through Integration of T-S Cloud Inference Network and Improved PSO

Lei Si,¹ Zhongbin Wang,¹ Xinhua Liu,^{1,2} Yinwei Yang,¹ and Lin Zhang¹

¹ School of Mechatronic Engineering, China University of Mining & Technology, Xuzhou 221116, China

² Xuyi Mine Equipment and Materials R&D Center, China University of Mining & Technology, Huai'an 211700, China

Correspondence should be addressed to Zhongbin Wang; wangzbpaper@126.com and Yinwei Yang; yywei-163@163.com

Received 10 July 2014; Accepted 10 November 2014; Published 23 November 2014

Academic Editor: Karim G. Oweiss

Copyright © 2014 Lei Si 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.

In order to efficiently and accurately adjust the shearer traction speed, a novel approach based on Takagi-Sugeno (T-S) cloud inference network (CIN) and improved particle swarm optimization (IPSO) is proposed. The T-S CIN is built through the combination of cloud model and T-S fuzzy neural network. Moreover, the IPSO algorithm employs parameter automation adjustment strategy and velocity resetting to significantly improve the performance of basic PSO algorithm in global search and fine-tuning of the solutions, and the flowchart of proposed approach is designed. Furthermore, some simulation examples are carried out and comparison results indicate that the proposed method is feasible, efficient, and is outperforming others. Finally, an industrial application example of coal mining face is demonstrated to specify the effect of proposed system.

1. Introduction

Currently, the cooperative control of coal mining machines (shearer, scraper conveyers, and hydraulic supports) is becoming a development trend in fully mechanized mining face. As a key factor of cooperative control, the traction speed of shearer has a great influence on the mining efficiency and the working states of other coal mining machines. Therefore, the traction speed should be precisely and reasonably adjusted in a reliable way. However, due to the poor working conditions of coal mining such as narrow space, high coal dust, low visibility, and large noise, shearer operator does not have accurate and timely manner to adjust the traction speed only depending on the vibration noise of shearer mining and manual visualization [1, 2]. This phenomenon cannot ensure shearer work in cooperation with other coal mining machines reasonably and may lead to the problem of low mining efficiency. Moreover, many safe accidents in collieries occurred increasingly frequently. Under this kind of background, the researches on adjustment methods for shearer traction speed have become a challenging and significant research subject.

Due to the randomness and complexity of underground geological conditions, adjusting traction speed would present the characteristics of complexity, fuzziness, uncertainty, and high risk, and this may affect the coal production or even endanger the operator's life. Therefore, it is necessary to handle the speed accurately and effectively. In the real mining condition, some key index parameters have a strong relationship with shearer traction speed and the relationship is highly nonlinear in nature so that it is hard to develop a comprehensive mathematic model. To deal with this kind of problem, the commonly used methods are fuzzy theory and neural networks [3–5]. Fuzzy neural network (FNN) can combine the advantages of both fuzzy logic in processing vague information and neural network in good learning abilities [6]. It can also handle imprecise information through linguistic expressions. For several decades, FNN has attracted much attention and has been applied in many domains [7].

As a typical type of FNN, Takagi-Sugeno (T-S) type model [8, 9] has been commonly and widely used. However, FNN adopts traditional membership functions (such as trigonometric function, trapezoidal function, and normal function) to describe the subordinate relations [10]. It is

difficult to completely reflect the uncertainty only through the precise membership functions. Li et al. proposed a new model, namely, the cloud model on the basis of random mathematics and fuzzy mathematics [11]. Cloud model uses linguistic values to represent the transformation between qualitative concepts and their quantitative representation. Thus, the cloud model can be introduced to replace the membership functions in conventional T-S FNN and then the T-S cloud inference network (T-S CIN) is constituted [12]. However, there are many drawbacks of T-S CIN by common back propagation (BP) algorithm with gradient descent for training, such as being easy to trap into local minimum point and poor ability on global search [13, 14]. In addition, the performance of BP training depends on the initial values of the system parameters, and for different network topologies it has to derive new mathematical expressions for each network layer. Based on the past work on artificial intelligent optimization algorithms, this paper tries to tackle the problem.

Bearing the above observation in mind, we propose an integrated approach based on T-S CIN and improved particle swarm optimization (IPSO) to adjust the shearer traction speed. The remainder of this paper is organized as follows. In Section 2, some related works are outlined based on literatures. Section 3 describes the integrated approach based on T-S CIN and IPSO algorithm and designs the flowchart of proposed algorithm. Section 4 provides some simulation examples and carries out the comparison with other methods to verify the feasibility, efficiency, and outperforming of others. An industrial example of mine automation production based on proposed system is demonstrated to specify the application effect in Section 5. Our conclusions are summarized in Section 6.

2. Literature Review

Recent publications relevant to this paper are mainly concerned with the streams of learning algorithms for T-S models. In this section, we try to summarize the relevant literatures.

In recent years, many researches have used genetic algorithms (GAs) for the learning of T-S models and attain better performance than BP algorithm [15]. In [16], a hybrid algorithm, combining the advantages of genetic algorithm's strong search capacity and Kalman filter's fast convergence merit, was proposed to construct a "parsimonious" fuzzy model with high generalization ability. Wang et al. proposed a new scheme based on multiobjective hierarchical genetic algorithm extract interpretable rule-based knowledge from data and this method was derived from the use of multiple objective genetic algorithms [17]. In [18], a hybrid system combining a fuzzy inference system and genetic algorithms was proposed to tune the parameters in the Takagi-Sugeno-Kang fuzzy neural network. Lin and Xu proposed a self-adaptive neural fuzzy network with group-based symbiotic evolution method and genetic algorithms were used to adjust the parameters for the desired outputs [19]. In [20], a fuzzy controller design method was proposed based on genetic algorithm to find the membership functions and the rule sets

simultaneously. Juang proposed a TSK-type recurrent fuzzy network with a genetic algorithm for control problems [21].

Recently, as a new branch in evolutionary algorithms, particle swarm optimization (PSO) has attracted many researchers' interests [22]. Compared with GA, the PSO has some attractive characteristics, such as simple concept, easy implementation, robustness to control parameters, and computation efficiency when compared with other heuristic optimization techniques. Successful applications of PSO in some optimization problems, such as function optimization and neural network optimization, have demonstrated its potential [23, 24]. The combined method of fuzzy model and PSO algorithm was proposed in [25, 26] and the authors found that PSO algorithm could generate better results for identifying the fuzzy model than GA with the same complex problem. Although PSO algorithm has been developing rapidly, it is relatively inefficient in local search and easy to result in premature convergence. Therefore, some improved approaches and variants of PSO have been reported. Du et al. proposed a novel hybrid learning algorithm based on random cooperative decomposing particle swarm optimization algorithm and discrete binary version of PSO algorithm, and the optimal structure and parameters of T-S FNNs were achieved simultaneously [27, 28]. In [29], a prediction algorithm for traffic flow of T-S fuzzy neural network and improved particle swarm optimization was proposed, and the improved strategy was used to make the algorithm jump out of local convergence by using t distribution. Lin proposed a new learning algorithm based on the immune-based symbiotic particle swarm optimization for use in TSK-type neurofuzzy networks to avoid trapping in a local optimal solution and to ensure the search capability of a near global optimal solution [30].

In addition, a cooperative particle swarm optimization (CPSO) algorithm has been proposed based on the notion of coevolution and proven to be more effective than the traditional PSO in most optimization problems [31]. In [32], a powerful cooperative evolutionary particle swarm optimization algorithm based on two swarms with different behaviors to improve the global performance of PSO was proposed. In [33], a novel adaptive cooperative PSO with adaptive search was presented, and the proposed approach combined cooperative learning and PSO to combat curse of dimensionality and control the balance of exploration and exploitation in all the smaller-dimensional subswarms.

According to above analysis, although many improved strategies for PSO have been proposed, they have some common shortcomings summarized as follows. Firstly, most improved IPSO algorithms are hard to get a good tradeoff between global convergence and convergent efficiency. Secondly, it cost long computation time and there is a weak ability in high dimension optimization problems. Finally, there is lack of the effective judge tool to determine whether the particles have gotten into local optimal value or not.

In this paper, an improved PSO algorithm is proposed by employing parameters automation strategy and velocity resetting, and the integrated method based on IPSO learning algorithm and T-S CIN is generated to adjust the shearer traction speed. Some simulation examples and comparison

with other methods are carried out, and the proposed approach is proved feasible and efficient.

3. The Proposed Method

3.1. Cloud Model. The cloud is a model using the linguistic value to represent the uncertainty conversion between a qualitative concept and its quantitative representation. Suppose U is a quantitative domain expressed in precise values and A is a qualitative concept in U . If a quantitative value $x \in U$ is a random realization of the qualitative concept A and the membership of x to A , $\mu(x) \in [0, 1]$, is a random number with a stable tendency: $\mu: U \rightarrow [0, 1], \forall x \in U, x \rightarrow \mu(x)$, then the distribution of x on domain U is called the cloud and each x is called a cloud droplet.

Normal cloud is widely used as a cloud model. We suppose that $R(E_1, E_2)$ denotes a one-dimensional normal distribution random function, where E_1 is the expected value and E_2 is the standard deviation. If $x (x \in U)$ and $\mu(x)$ satisfy the equations, which can be expressed as follows:

$$\begin{aligned} x &= R(\text{Ex}, \text{En}), \\ p &= R(\text{En}, \text{He}), \\ \mu &= \exp\left(-\frac{(x - \text{Ex})^2}{2p^2}\right) \end{aligned} \quad (1)$$

then the distribution of x on domain U is called the normal cloud. In (1), Ex , En , and He denote the expectation, entropy, and hyper entropy, respectively, which are used to describe the numerical characteristics of cloud. Ex is the expectation of cloud droplets in the distribution of the domain and is the most typical point that represents this qualitative concept. En is the uncertain measurement of the qualitative concept and reflects the relevance of fuzziness and randomness. He is the uncertain measurement of entropy and is determined by the fuzziness and randomness.

A possible form of normal cloud and membership function, whose linguistic values are close to zero, can be shown as Figure 1. Obviously, membership function is a specific curve. Once the membership function represents the property of fuzziness, it is no longer vague. However, normal cloud is composed of some cloud droplets, which can reflect the fuzziness. The membership is a group of random values with a stable tendency, rather than fixed values. Cloud model is not described through certain functions, therefore, to enhance the processing capacity for uncertainty.

3.2. Structure of T-S Cloud Inference Network. For a multiple-input and single-output (MISO) system, the T-S model can be given as follows: let $X = [x_1, x_2, \dots, x_n]$ denote an input vector, where each variable x_i is a fuzzy linguistic variable. The set of linguistic variables for x_i is represented by $T(x_i) = \{A_i^1, A_i^2, \dots, A_i^m\}$ ($i = 1, 2, \dots, n$), where A_i^j ($j = 1, 2, \dots, m$) is the j th linguistic value of the input x_i . The membership of fuzzy set defined on domain of x_i is μ_{ij} ($i = 1, 2, \dots, n, j = 1, 2, \dots, m$). According to [8], the T-S CIN is composed

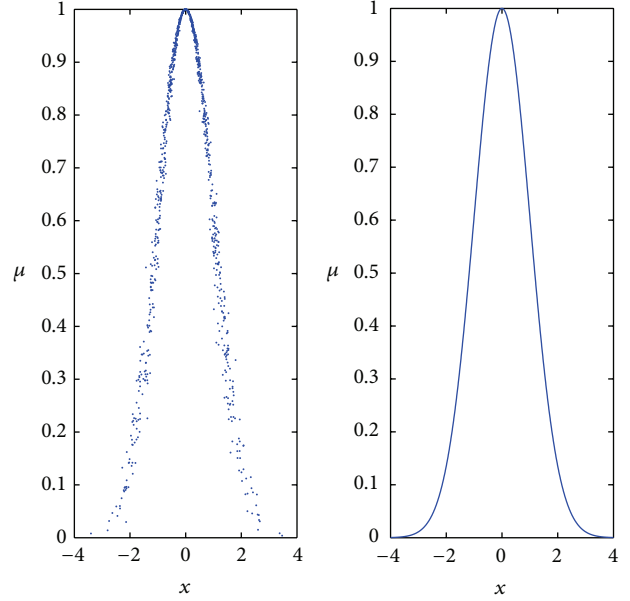


FIGURE 1: Normal cloud and membership function.

of four layers, which can be divided into two networks: antecedent network and consequent network. The first three layers of this T-S CIN correspond to the antecedent network and the fourth layer is output layer. The structure of T-S CIN can be described as Figure 2.

In Figure 2, the purpose and meaning of each layer can be defined as follows.

First Layer. This layer is the input layer of antecedent network and no function is performed in this layer. The nodes are only used to transmit the input values to the second layer.

Second Layer. This layer is the fuzzification layer by the use of cloud model. Nodes in this layer correspond to one linguistic label of the input variables in the first layer. Each node represents a cloud model, which is used to realize the cloud of input variables. In this study, the number of partitions for the cloud is set as m and the total number of second layers is $n \times m$. The degree of membership cloud for input variable x_i can be calculated through the following equations:

$$\begin{aligned} \mu_{ij}(x_i) &= \exp\left[-\frac{(x_i - \text{Ex}_{ij})^2}{2p_{ij}^2}\right], \\ (i &= 1, 2, \dots, n; j = 1, 2, \dots, m), \end{aligned} \quad (2)$$

where $p_{ij} = R(\text{En}_{ij}, \text{He}_{ij})$.

Third Layer. This layer is the cloud inference layer (cloud rule layer). Firing strength of every rule is calculated. Each node describes one cloud rule and is used to match the input vector. The degree that the input vector X matches rule Rule_j can be computed through the following equation:

$$\lambda_j = \mu_{1j} \cdot \mu_{2j} \cdots \mu_{nj} = \prod_{i=1}^n \mu_{ij} \quad (j = 1, 2, \dots, m), \quad (3)$$

where λ_j is called the firing strength of rule Rule_j .

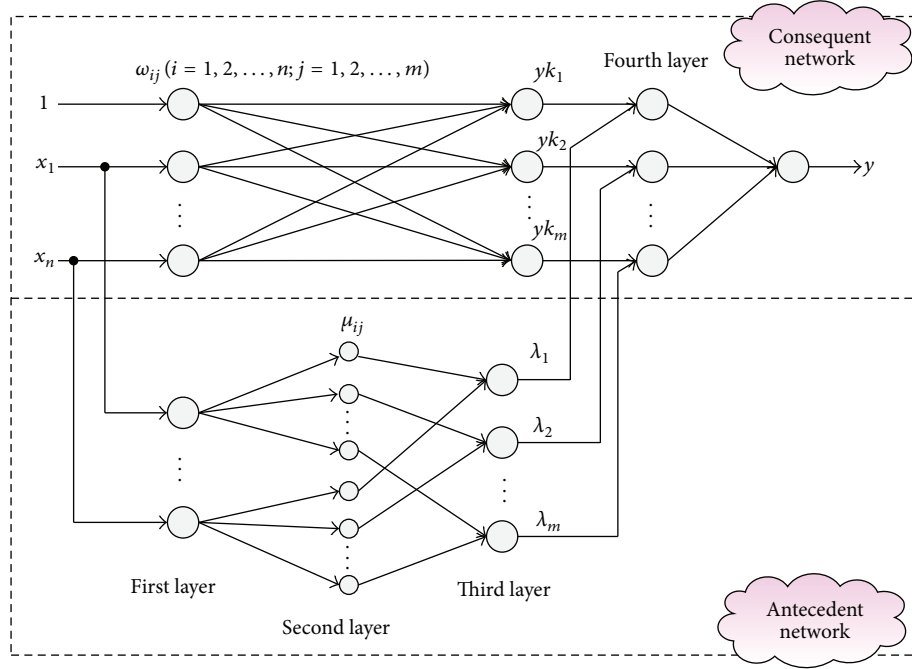


FIGURE 2: Structure of T-S cloud inference network for the MISO system.

Fourth Layer. In consequent network, it is a linear relationship between the layers. The hidden layer output of this network can be given through the following equation:

$$yk_j = \sum_{i=0}^n x_i \cdot \omega_{ij} \quad (x_0 = 1; j = 1, 2, \dots, m), \quad (4)$$

where ω_{ij} is the coefficient of the network.

The output layer sums up all the activated values from the cloud inference rules to generate the overall output y , which can be calculated by

$$y = \frac{\sum_{j=1}^m \lambda_j \cdot yk_j}{\sum_{j=1}^m \lambda_j}. \quad (5)$$

3.3. Learning Algorithm for T-S CIN. According to the principle of T-S CIN, the structure and parameters mainly include the expectation Ex_{ij} , entropy En_{ij} , hyper entropy He_{ij} of cloud model, and coefficient ω_{ij} of consequent network. Conventional learning algorithm for T-S CIN is the gradient descent method. However, the initial values of gradient descent method have a great influence on the learning effect of network and this method is easy to fall into local minimum. In this paper an improved particle swarm optimization algorithm (IPSO) is proposed as the learning algorithm to optimize the structure and parameters of T-S CIN.

The basic particle swarm optimization algorithm (PSO) is that a swarm of particles are initialized randomly in the solution space and each particle motions in a certain rule to explore the optimal solution after several iterations. It has two attributes of position and velocity. The position of the i th particle is X_i and the velocity can be denoted by V_i . In T-S

CIN, the parameter of hyper entropy He_{ij} is the uncertain measurement of entropy and depends on the actual situation. In this paper, He_{ij} is set as $He_{ij} = En_{ij}/10$. Thus, other parameters should be optimized through PSO. The location of a particle X_i corresponding to T-S CIN can be encoded as Figure 3.

Therefore, the position and the velocity of the i th particle can be given as

$$X^i = \begin{bmatrix} x_{11}^i, x_{12}^i, \dots, x_{1,3n+1}^i \\ x_{21}^i, x_{22}^i, \dots, x_{2,3n+1}^i \\ \vdots \\ x_{m1}^i, x_{m2}^i, \dots, x_{m,3n+1}^i \end{bmatrix}, \quad (6)$$

$$V^i = \begin{bmatrix} v_{11}^i, v_{12}^i, \dots, v_{1,3n+1}^i \\ v_{21}^i, v_{22}^i, \dots, v_{2,3n+1}^i \\ \vdots \\ v_{m1}^i, v_{m2}^i, \dots, v_{m,3n+1}^i \end{bmatrix}.$$

Particles are updated through tracking two “extremums” in each iteration. One is the individual optimal solution $P^i = [p_{jl}^i]_{m \times (3n+1)}$ found by the particle itself and another is the global optimal solution $P^g = [p_{jl}^g]_{m \times (3n+1)}$ found by the particle population. The specific iteration formulas can be expressed as follows:

$$v_{jl}^i(k+1) = w_k v_{jl}^i(k) + c_1 r_1 [p_{jl}^i(k) - x_{jl}^i(k)] + c_2 r_2 [p_{jl}^g(k) - x_{jl}^i(k)], \quad (7)$$

$$x_{jl}^i(k+1) = x_{jl}^i(k) + v_{jl}^i(k+1),$$

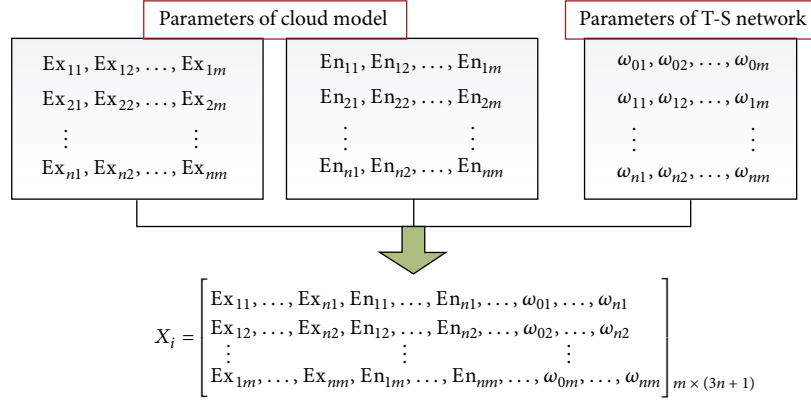


FIGURE 3: Encoding of a particle location.

where k is the current iteration times; $i = 1, 2, \dots, M$, M is the number of particles; $j = 1, 2, \dots, m$ and $l = 1, 2, \dots, 3n + 1$; c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are uniformly distributed random numbers in the range $(0, 1)$. The velocity vector V is limited to the range $[-V_{\max}, V_{\max}]$ to reduce the likelihood of the particle leaving the search space and the position vector X is clamped to the range $[X_{\min}, X_{\max}]$, which can be determined according to practical problem and V_{\max} is usually chosen to be $\alpha \times X_{\max}$, with $\alpha \in [0.1, 1.0]$; ω_k is the current inertia weight.

Shi and Eberhart [34] proposed a linearly varying inertia weight (w_k) over the course of generations, which significantly improves the performance of PSO and can be updated by the following equation:

$$w_k = (w_{\max} - w_{\min}) \frac{T - k}{T} + w_{\min}, \quad (8)$$

where w_{\max} and w_{\min} are the maximum and minimum of inertia weight; T is the maximum number of allowable iterations. The empirical studies in [34] indicated that the optimal solution can be improved by varying the value of w_k from 0.9 at the beginning of the evolutionary process to 0.4 at the end of the evolutionary process for most problems.

Although the version of PSO based on the time-varying inertia weight is capable of locating a good solution with a significantly faster velocity, the ability to fine-tune the optimum solution is comparatively weak, mainly due to the lack of diversity at the end of the evolutionary process. Observed from (7), the particles tend to the optimal solution through two stochastic components: one is the cognitive component and another is the social component. Thus, proper control of the two components is urgently needed and effective for searching for the optimum solution. In this paper, a version of PSO based on time-varying acceleration coefficients is presented to adjust the components by decreasing c_1 and increasing c_2 with time. Based on empirical studies, Ratnaweera et al. [35] have observed that the optimal solutions on most of the benchmarks can be improved by decreasing c_1 from 2.5 to 0.5 and increasing c_2 from 0.5 to

2.5 over the full range of the search. Therefore, the varying scheme of c_1 and c_2 can be given as follows:

$$c_1 = 2.5 - (2.5 - 0.5) \cdot \frac{k}{T}, \quad (9)$$

$$c_2 = (2.5 - 0.5) \cdot \frac{k}{T} + 0.5.$$

At the beginning of the search, a large cognitive component and a small social component are assigned to guarantee the particles' moving around the search space. On the other hand, a small cognitive component and a large social component allow the particles to converge to the global optimum in the latter of the search.

PSO can quickly find a good local solution but it sometimes suffers from stagnation without an improvement and then traps in the local optimal solution. In this study, the fitness variance is adopted to measure whether PSO gets into local optimum, which can be calculated as follows:

$$\sigma^2 = \frac{1}{M} \sum_{i=1}^M \left[\frac{1}{f_{\Delta}} \left(f_i - \frac{1}{M} \sum_{i=1}^M f_i \right) \right]^2, \quad (10)$$

where f_i denotes the fitness of the i th particle; f_{Δ} denotes the normalized factor. The fitness function and f_{Δ} can be calculated as follows:

$$f_i = \frac{1}{q} \sum_{s=1}^q (y_s - Y_s)^2, \quad (11)$$

$$f_{\Delta} = \max \left\{ 1, \max \left\{ \left| f_i - \frac{1}{M} \sum_{i=1}^M f_i \right| \right\} \right\},$$

where q is the total number of training samples; y_s is the network output of the s th training sample; Y_s is the expected output. Thus, f_i is the normalized mean squared error (MSE) of the individual i on the training set.

The fitness variance σ^2 is the symbol of particles convergence degree. When σ^2 is smaller than a specified value σ_{\min}^2 , the algorithm is considered as falling into precocity.

Therefore, to avoid this drawback of basic PSO, a mutation mechanism based on resetting the velocity is proposed to enable particles to have a new momentum. Under this new strategy, when $\sigma^2 < \sigma_{\min}^2$, each particle i will be selected by a predefined probability from the population, and then a random perturbation is added to each dimension v_{jl}^i (selected by a predefined probability) of velocity vector V^i of the selected particle i . The pseudocode of resetting velocity can be given as in Pseudocode 1, where $p_{.1}$, $p_{.2}$, and $p_{.3}$ are separately generated and uniformly distributed random numbers in range (0, 1).

3.4. *Flowchart of Proposed Method.* With above specific treatment, structure and parameters of the T-S CIN evolution can be implemented by IPSO. According to above description about the learning algorithm for T-S CIN, the proposed approach is an iterative algorithm and can be coded easily on the computer, and the flowchart can be summarized as shown in Figure 4.

4. Simulation Examples

In this section, an example on the adjustment of shearer traction speed is provided to validate the proposed method. The aim of this study is to improve the accuracy and efficiency of identification for traction speed. Furthermore, the example can be divided into three main stages. Firstly, according to the working principle of shearer, the level of traction speed and mainly evaluation indexes can be determined and the sample can be established reasonably. Secondly, according to the obtained evaluation indexes, the T-S CIN model can be constructed. Thirdly, the constructed standard T-S CIN, the T-S CIN with PSO, IPSO, and traditional T-S FNN, the T-S FNN with PSO, IPSO optimization are, respectively, tested with the same training and test samples to compare the accuracy and efficiency in adjustment of shearer traction speed.

4.1. *Sample Preparation.* In a fully mechanized coal mining face, the adjustment of shearer traction speed should consider the coordination with other coal mining equipment (scraper conveyor and hydraulic support). After the analysis of shearer working principle, the evaluation indexes of traction speed mainly consist of cutting motor current (CMC), cutting motor temperature (CMT), traction motor current (TMC), traction motor temperature (TMT), scraper conveyor current (SCC), and scraper conveyor speed (SCS). For a fixed shearer of MG 300/730-WD, the adjusting range of traction speed is 0~9.0 m/min. The levels of traction speed are reasonably partitioned and can be applied in controlling shearer. Based on the empirical studies, the speed levels can be divided as 0~2.0 m/min (Class I), 2.0~3.5 m/min (Class II), 3.5~4.5 m/min (Class III), 4.5~6.0 m/min (Class IV), 6.0~7.5 m/min (Class V), and 7.5~9.0 m/min (Class VI). However, as the information in the database is collected after the workers operate the coal mining equipment, the information maybe not very ideal and practical. Therefore, a threshold of 0.2 is introduced to express the subjective factors, and the traction speed levels from the database can be processed and described as Figure 5.

```

For i = 1 to M
  If p_{.1} > 0.5
    Particle i is selected for velocity resetting
    For j = 1 to m
      For l = 1 to (3n + 1)
        If p_{.2} > 0.5
          v_{jl}^i = v_{jl}^i + (2 * p_{.3} - 1) * V_{max}
        End if
      End for
    End for
  End if
End for

```

PSEUDOCODE 1

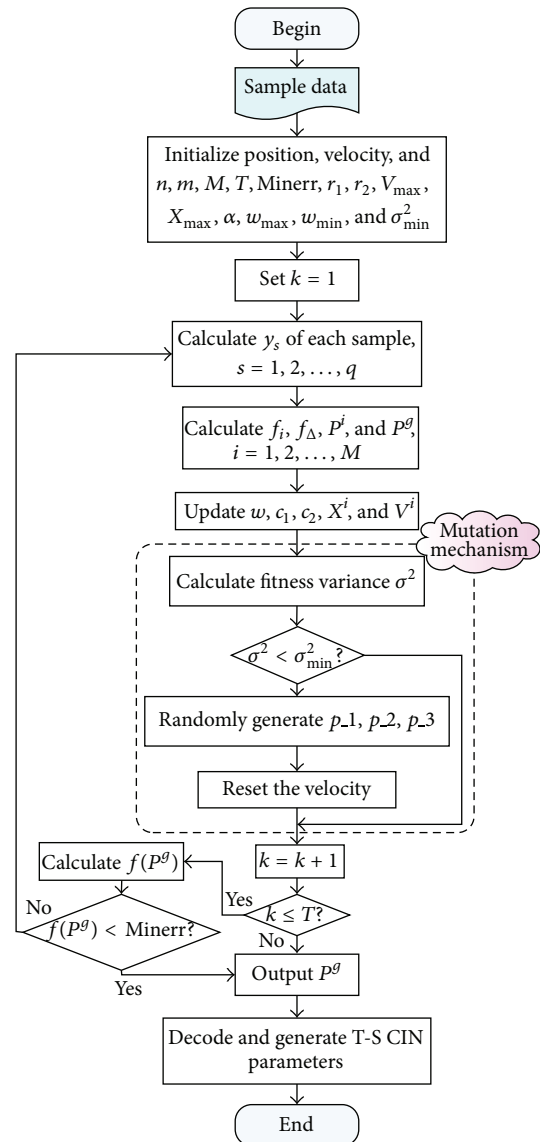


FIGURE 4: Flowchart of proposed method.

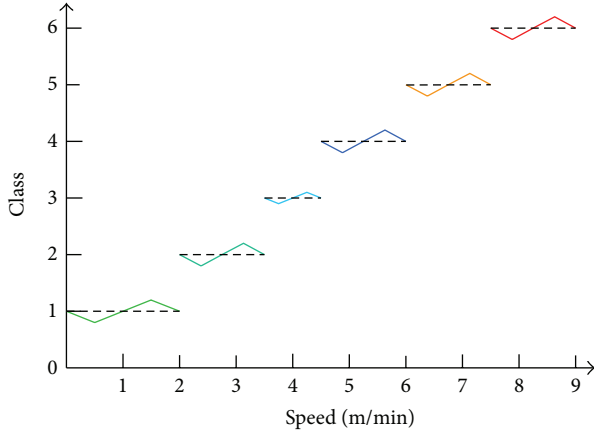


FIGURE 5: Redefined levels of traction speed.

Taken Class 1 (Class I) as an example, the level of speed 0~2 m/min can be redefined as follows:

$$\text{Class}(Sp)_1^{\text{New}} = \begin{cases} -0.4Sp + 1, & 0 < Sp \leq 0.5, \\ 0.4Sp + 0.6, & 0.5 < Sp \leq 1.5, \\ -0.4Sp + 1.8, & 1.5 < Sp \leq 2, \end{cases} \quad (12)$$

where Sp is the current traction speed of shearer. In the same way, the redefined functions for other speeds can also be obtained easily.

According to the information database acquired from the 2215 coal face in Changcun Coal Mine of Yima Coal Industry Group Co., 400 groups of samples are randomly extracted and rearranged as shown in Figure 6.

4.2. Parameters Selection for Proposed Method. There are some parameters in IPSO which need to be specified by the user. However, it is unnecessary to tune all these parameters for the sample data because IPSO is not very sensitive to them. Therefore, these parameters are set as the number of particles M (50); the maximum number of allowable iterations T (500); the position and velocity range of particles $([-1, 1])$; the initial acceleration coefficients c_1 and c_2 of IPSO (2.5 and 0.5); the inertia weights w_{\max} and w_{\min} of IPSO (0.9 and 0.4); the termination error Minerr (0.0001); the minimum fitness variance for mutation σ_{\min}^2 (0.001).

The structure of T-S CIN is determined by the sample data. In this simulation example, the input data of T-S CIN is 6-dimensional and output data is 1-dimensional. Thus, $n = 6$ and m can be set as 12. Other parameters including expectation Ex_{ij} , entropy En_{ij} , hyper entropy He_{ij} , and coefficient ω_{ij} can be optimized through IPSO.

4.3. Simulation Results. The sample data in Figure 6 should be normalized firstly and are randomly split into a training data set containing 350 samples and a testing data set containing the remaining 50 samples, which is only used to verify the accuracy and the effectiveness of the trained T-S CIN model.

The relevant parameters are given as Section 4.3 described. The proposed method runs 10 times and the mean

values are regarded as the final results. The performance criterion of T-S CIN can be measured by the mean squared absolute error (MSE) and the mean absolute error (MAE) between the predicted outcome and the actual outcome. The learning curves with MSE and MAE of T-S CIN model based on IPSO can be shown in Figure 7.

As shown in Figure 7, after the IPSO-based T-S CIN model is trained for 500 times, MSE of the training samples can reach 0.00065 and MAE can reach 0.00987. Actually, the values of MSE and MAE basically keep stable at the times of 280, which can show good convergence performance of proposed method.

After the training phase, a T-S CIN model can be obtained. In order to verify the accuracy of the model, the remaining 50 samples are utilized to test its performance. The prediction errors and deviation comparison diagrams of the network output and actual output are given as Figure 8. As shown in Figure 8, the MSE and MAE of testing samples are 0.006118 and 0.0346, respectively, showing good generalization performance. Furthermore, the mean relative error and maximum relative error are 1.23% and 5.78%, which satisfies the accuracy requirement.

4.4. Comparison with Other Methods. In order to indicate the meliority of T-S CIN integrating IPSO, the T-S CINs based on the basic PSO (bPSO), CPSO, and IPSO are provided to solve the problem of above example. The training samples and testing samples are the same. The configurations of simulation environment for three algorithms are uniform and the relevant parameters are in common with above example. The compared learning curves with MSE and MAE of T-S CIN models based on bPSO, CPSO, and IPSO can be shown in Figure 9 and some performance criterions are listed in Table 1, where 50_MSE and 50_MAE are the values of MSE and MAE in the stage of 50 iterations. Furthermore, MRE and MaxRE denote the mean relative error and maximum relative error of the network output and actual output.

Seen from Figure 9 and Table 1, the declining velocity of the error of CPSO and IPSO is faster than that of bPSO during the training phase. The MAE of IPSO-based T-S CIN gets to <0.05 for 30 iterations and the MSE of training phase reaches a stable phase for 300 iterations. However, the training errors of MAE with the bPSO, CPSO-based T-S CIN model are still 0.05026 and 0.1293 for 30 iterations. In the testing phase, the test sample error of bPSO, CPSO-based T-S CIN is much larger than the same input conditions of proposed method. By analysis, the criterions of CPSO-based T-S CIN are more excellent than these of other methods both in the training stage and in the testing stage, which proves the effectiveness and feasibility of proposed method.

In order to verify the superiority of T-S CIN (T-S NN coupling cloud model), the sample data in Figure 6 are used to test the performance of T-S CIN and conventional T-S FNN, and the proposed IPSO is also integrated with the two networks. Thus, four algorithms are developed, marked as T-S FNN, T-S CIN, T-S FNN_IPSO, and T-S CIN_IPSO. The configurations of simulation environment for four algorithms are uniform and the parameters are in common with above

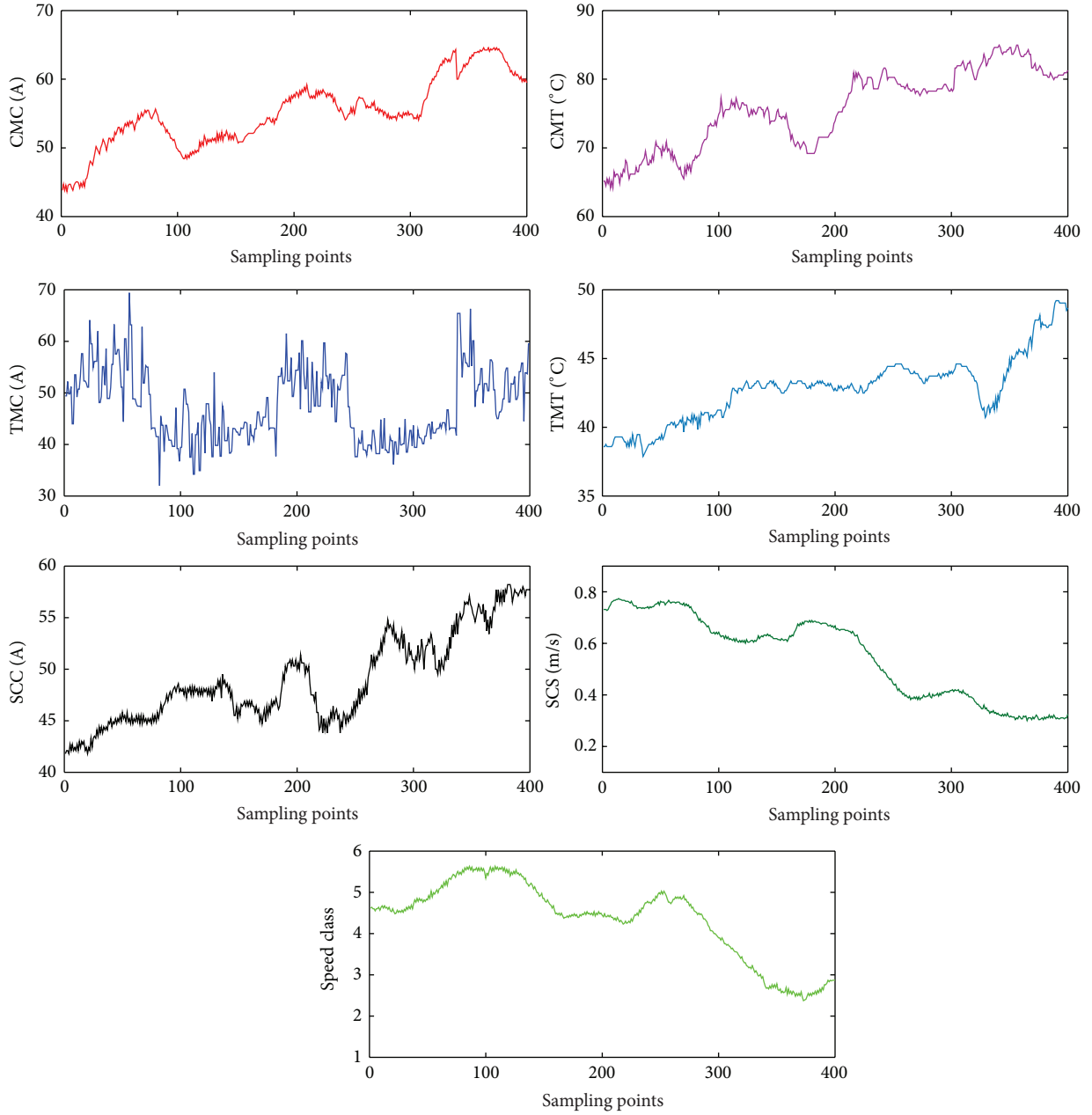


FIGURE 6: Sample data of this example.

TABLE 1: The compared criterions of T-S CIN based on bPSO, CPSO, and IPSO.

Method	Training phase				Testing phase			
	MSE	MAE	50_MSE	50_MAE	MSE	MAE	MRE	MaxRE
bPSO	0.04024	0.05396	0.07361	0.11573	0.09726	0.2324	9.86%	17.36%
CPSO	0.00626	0.02698	0.03657	0.04956	0.01853	0.0954	4.65%	9.57%
IPSO	0.00065	0.00987	0.02196	0.03779	0.006118	0.0346	1.23%	5.78%

simulation example. The training samples and testing samples of these algorithms should keep consistent. In order to avoid the random error, each algorithm runs 10 times and calculated the average values. The comparison diagram of different testing results is shown in Figure 10.

As Figure 10 illustrated, the prediction errors of T-S CIN are obviously smaller than these of T-S CIN. Through the application of cloud model replacing the membership function in T-S model, the processing capacity for the uncertainty of the problem can be enhanced and the T-SCIN performs

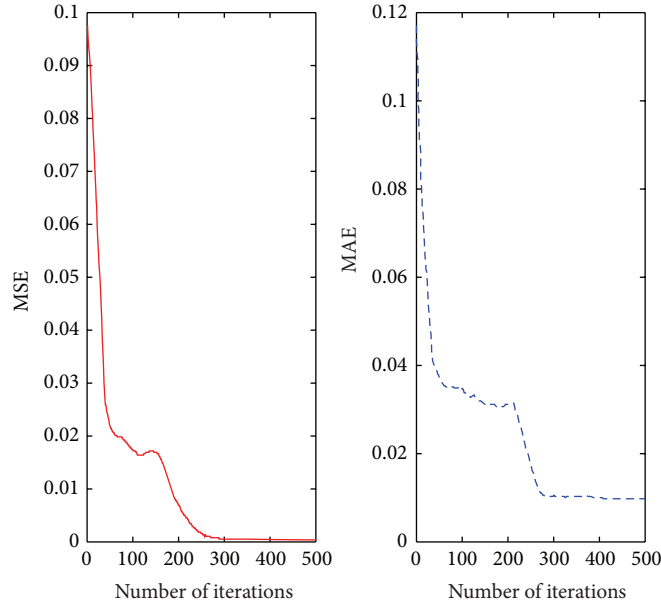


FIGURE 7: The learning curves of T-S CIN model based on IPSO.

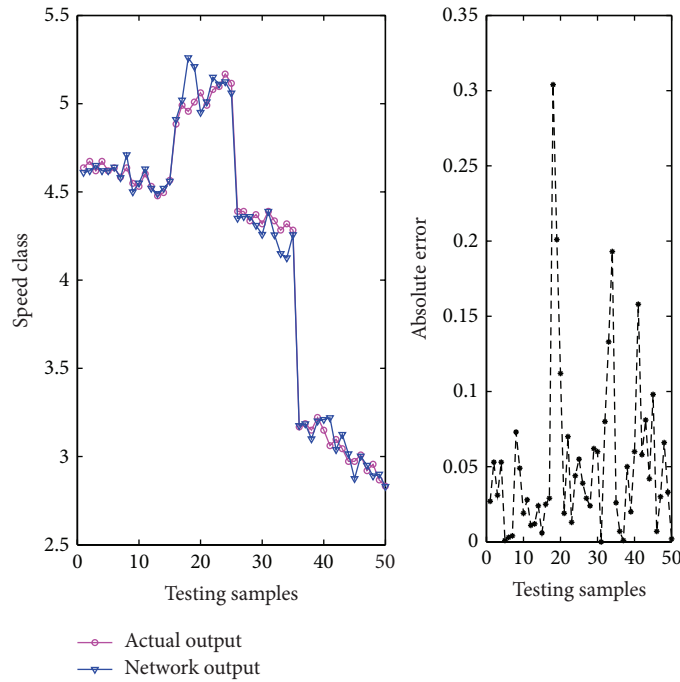


FIGURE 8: Comparison of network output and actual output.

with lower MSE, MAE, MRE, and MaxRE. Furthermore, the compared results of coupling IPSO algorithm verify the outperforming others of proposed method.

4.5. Further Discussion. In order to further compare and analyze the overall performance of T-S CIN based on IPSO, CPSO, and PSO optimization with the optimal solution (the actual value), the same 400 samples are experimented. In this example, a certain number of samples, denoted by

training-size (T_{size}), are randomly selected from the data as the training samples and 50 samples are randomly selected from the remaining $400 - T_{size}$ samples as the testing samples. Each neural network is then trained and tested 50 times and the average result is recorded as the final result. In this study, the training-size of the example varies over $T_{size} = 50, 80, 110, \dots, 350$. That is to say, we run several trials over the networks with training-size ranging from 50 to 350. According to [36], the relative error $|y - Y|/Y$ (where y is

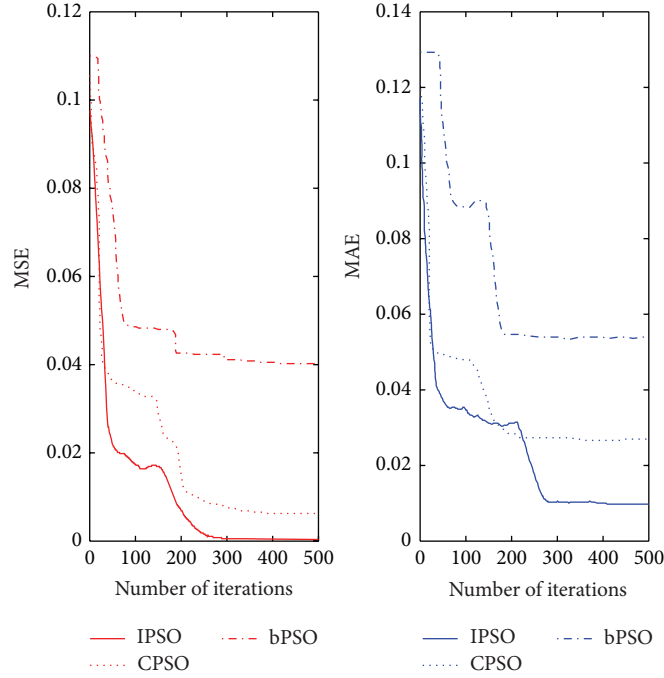


FIGURE 9: The compared learning curves with MSE and MAE of T-S CIN based on bPSO, CPSO, and IPSO.

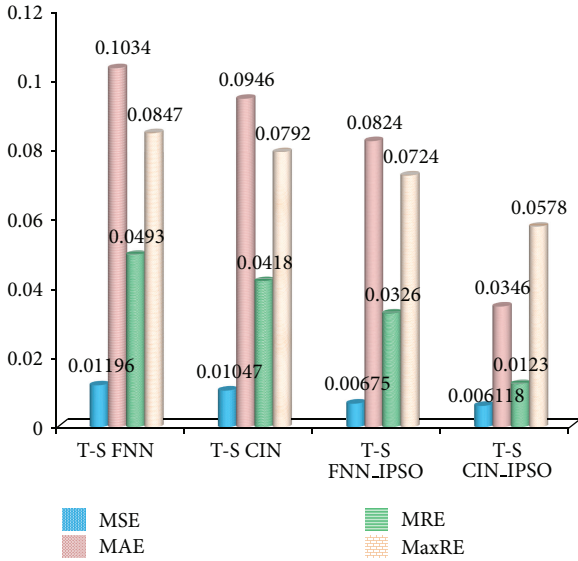


FIGURE 10: Comparison of the testing results based on four algorithms.

the network output and Y is the expected output) is chosen as the metric to express the result as a proportion of the optimal solution (the actual value).

Figure 11 plots the means of this metric (MRE) for each trial as a function of problem size T_{size} . It can be seen that for all trials the MRE decreases nonlinearly with T_{size} and the T-S CIN based on IPSO optimization outperforms T-S CIN based on CPSO optimization, which in turn outperforms T-S CIN based on bPSO optimization for all T_{size} .

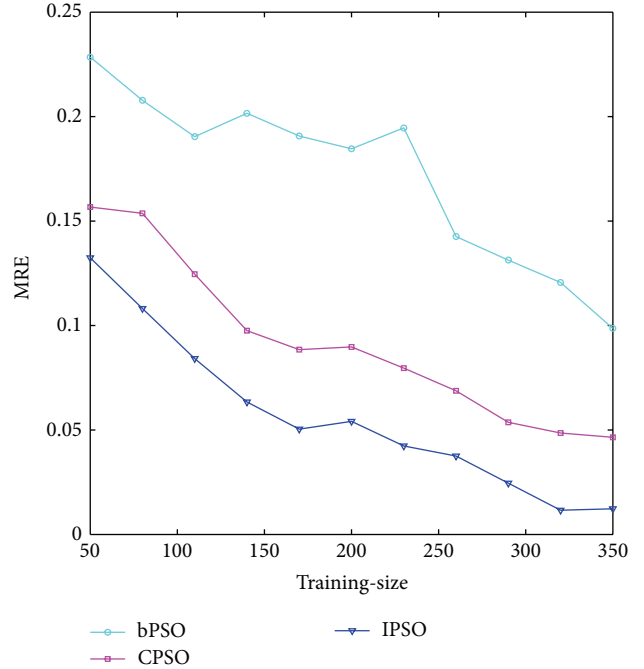


FIGURE 11: The changes of MRE with different training-sizes.

From Figure 11, it is obvious that the deviation of T-S CIN based on IPSO optimization is the smallest across different training-sizes, which means that the T-S CIN based on IPSO optimization is more stable and robust, and owns stronger generalization ability than T-S CIN based on CPSO and PSO

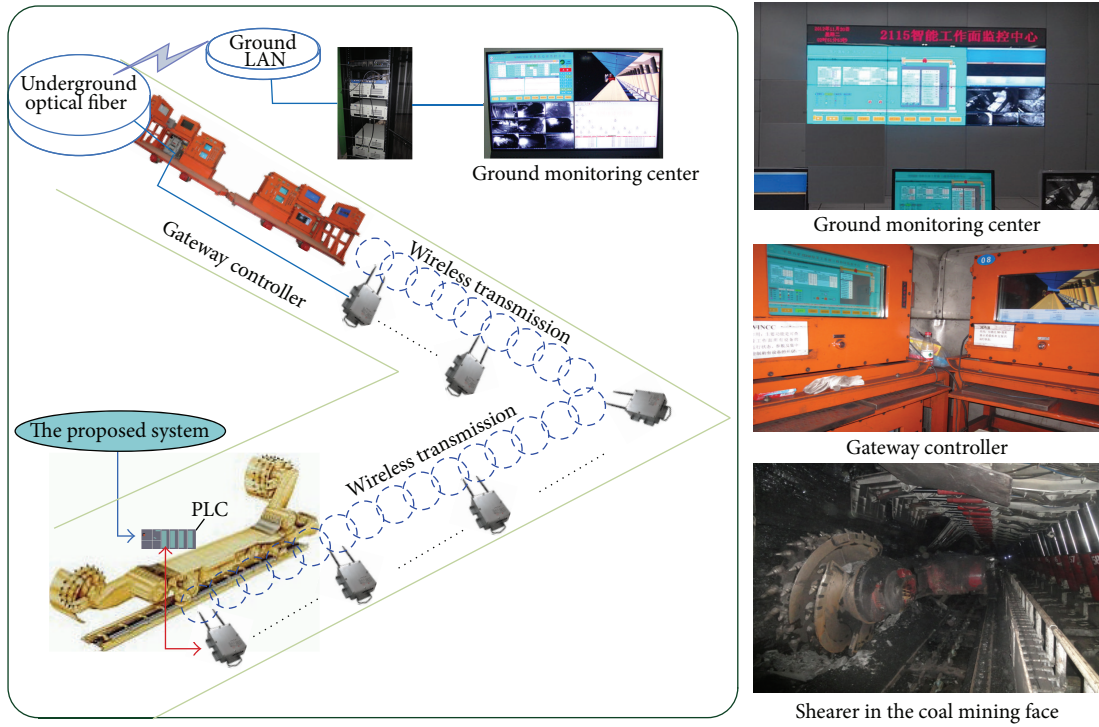


FIGURE 12: Industrial application example of proposed method.

optimization regardless of the training-size. Therefore, the T-S CIN based on IPSO optimization can obtain a relative high accuracy to provide an effective support tool for fuzzy and uncertain adjustment for shearer traction speed.

5. Industrial Application

In this section, a system based on proposed approach has been developed and applied in the field of coal mining face as shown in Figure 12.

As Figure 12 has shown, the “Gateway controller” and “Ground monitoring center” are used to control and monitor the shearer working parameters, which are located underground and on the ground, respectively. The proposed system is uploaded into the PLC (programmable logic controller) installed on the shearer and the speed level can be obtained. The traction speed of shearer can be adjusted through the speed level with Figure 5. The parameters of shearer are transferred into the “Gateway controller” through the wireless network. The “Ground monitoring center” receives these data through the communication of the underground optical fiber and the ground LAN.

For the shearer, the aim of adjusting traction speed is to ensure shearer mine coal smoothly and efficiently when shearer cuts the coal with gangue. In order to illustrate the application effect of proposed system, the shearer operator records the location of cutting the coal or the coal with gangue. This effect can be perfectly reflected through the changes of cutting motor current. In this experiment, the cutting motor current is collected every 1 Hz and the collected

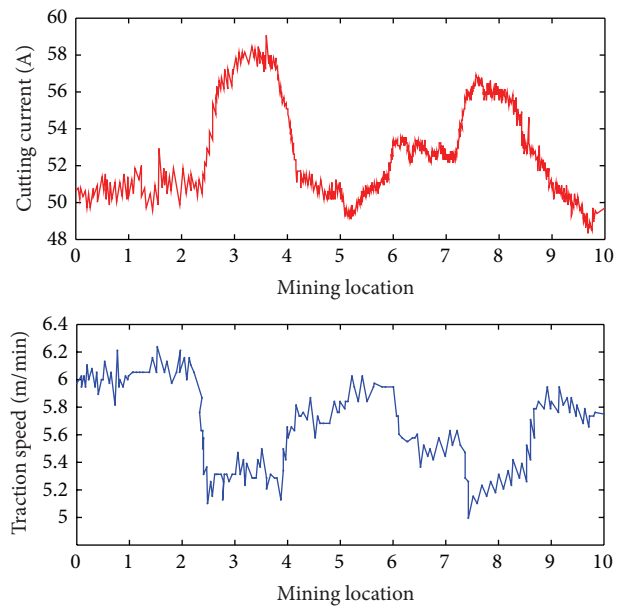


FIGURE 13: Application effect of proposed system.

data are transmitted to the “Gateway controller” and “Ground monitoring center.” The change curve of cutting motor current is plotted to illustrate the application effect of proposed system, as shown Figure 13.

Seen from Figure 13, the cutting currents at the location of 2.5 m to 4.0 m and 7.3 m to 8.2 m are a little higher than other locations because shearer cut the coal with gangue,

and the corresponding traction speeds are adjusted timely to lower levels through the proposed system. The application effect indicates that the system based on proposed method can provide a feasible strategy for safe and efficient coal mining.

6. Conclusions

In this paper, a novel adjustment method for shearer traction speed is proposed, which is based on T-S CIN with integrating IPSO algorithm. IPSO enables T-S CIN to dynamically evolve its parameters by using a specific individual representation and evolutionary scheme. To improve efficiency of PSO in global search and fine-tuning of the solutions, parameter automation adjustment strategy and velocity resetting are used in IPSO algorithm. To demonstrate the performance of proposed method, some simulation examples are provided and some comparisons with other methods are carried out. The results verify that the IPSO-based T-S CIN is an effective support tool for fuzzy and uncertain traction speed adjusting of shearer.

Conflict of Interests

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

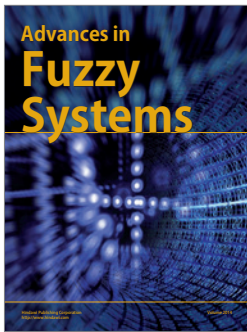
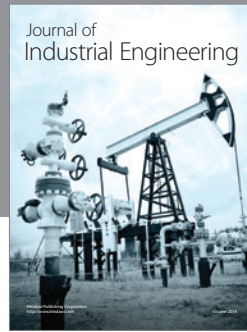
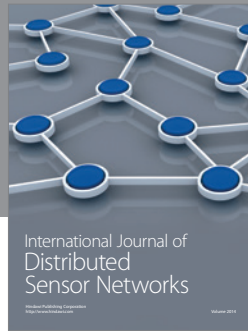
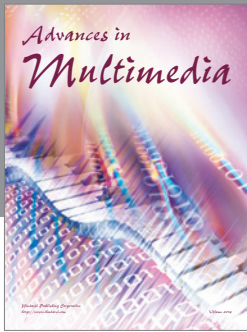
Acknowledgments

The supports of National High Technology Research and Development Program of China (no. 2013AA06A411), National Key Basic Research Program of China: Key Fundamental Research on the Unmanned Mining Equipment in Deep Dangerous Coal Bed (no. 2014CB046300), and the Priority Academic Program Development of Jiangsu Higher Education Institutions in carrying out this research are gratefully acknowledged.

References

- [1] L. Si, Z. B. Wang, X. H. Liu, and C. Tan, "Cutting path planning of shearer based on prediction of coal seam distribution," *Journal of China University of Mining & Technology*, vol. 43, no. 3, pp. 475–482, 2014.
- [2] Z. Wang, L. Si, C. Tan, and X. Liu, "A novel approach for shearer cutting load identification through integration of improved particle swarm optimization and wavelet neural network," *Advances in Mechanical Engineering*, vol. 2014, Article ID 521629, 13 pages, 2014.
- [3] M. Otadi, "System of fully fuzzy nonlinear equations with fuzzy neural network," *Neural Computing and Applications*, vol. 21, no. 1, pp. 369–376, 2012.
- [4] M. R. Mosavi, "Comparing DGPS corrections prediction using neural network, fuzzy neural network, and Kalman filter," *GPS Solutions*, vol. 10, no. 2, pp. 97–107, 2006.
- [5] Z. X. Wang, Y. G. Sun, Q. Zhang, J. Qin, X. W. Sun, and H. Y. Shen, "Research on fuzzy neural network algorithms for nonlinear network traffic predicting," *Optoelectronics Letters*, vol. 2, no. 5, pp. 373–375, 2006.
- [6] Y. Chen, B. Yang, A. Abraham, and L. Peng, "Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 3, pp. 385–397, 2007.
- [7] Y. Yoon, T. Guimaraes, and G. Swales, "Integrating artificial neural networks with rule-based expert systems," *Decision Support Systems*, vol. 11, no. 5, pp. 497–507, 1994.
- [8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [9] F.-J. Lin, C.-H. Lin, and P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 751–759, 2001.
- [10] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [11] D. Y. Li, C. Y. Liu, and L. Y. Liu, "Study on the universality of the normal cloud mode," *Engineering Sciences*, vol. 6, no. 8, pp. 28–34, 2004.
- [12] S. Q. Gan, H. P. Liu, and Z. J. Shen, "An improved T-S fuzzy neural network," *Control Engineering of China*, vol. 12, no. 5, pp. 442–445, 2005.
- [13] W. Pedrycz and M. Reformat, "Evolutionary fuzzy modeling," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 5, pp. 652–665, 2003.
- [14] X. Zhang, W. Zhao, S. Zhang, and T. Xu, "Method of flatness pattern recognition based on improved T-S cloud inference network," *Journal of Central South University (Science and Technology)*, vol. 44, no. 2, pp. 580–586, 2013.
- [15] S.-K. Oh, W. Pedrycz, and H.-S. Park, "Hybrid identification in fuzzy-neural networks," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 399–426, 2003.
- [16] L. Wang and J. Yen, "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter," *Fuzzy Sets and Systems*, vol. 101, no. 3, pp. 353–362, 1999.
- [17] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 149–186, 2005.
- [18] A. M. Tang, C. Quek, and G. S. Ng, "GA-TSKfnn: parameters tuning of fuzzy neural network using genetic algorithms," *Expert Systems with Applications*, vol. 29, no. 4, pp. 769–781, 2005.
- [19] C.-J. Lin and Y.-J. Xu, "A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications," *Fuzzy Sets and Systems*, vol. 157, no. 8, pp. 1036–1056, 2006.
- [20] A. Homaifa and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 9, pp. 129–139, 1995.
- [21] C.-F. Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 155–170, 2002.
- [22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, December 1995.
- [23] P. Damodaran, D. A. Diyadawagamage, O. Ghrayeb, and M. C. Vélez-Gallego, "A particle swarm optimization algorithm for

- minimizing makespan of nonidentical parallel batch processing machines,” *International Journal of Advanced Manufacturing Technology*, vol. 58, no. 9–12, pp. 1131–1140, 2012.
- [24] A. D. Deng, L. Zhao, and W. Wei, “The application of wavelet neural network optimized by particle swarm in localization of acoustic emission source,” in *Neural Information Processing*, vol. 5864 of *Lecture Notes in Computer Science*, pp. 738–745, Springer, 2009.
- [25] A. Khosla, S. Kumar, and K. K. Aggarwal, “A framework for identification of fuzzy models through particle swarm optimization,” in *Proceedings of the IEEE Indicon Conference*, pp. 388–391, Chennai, India, 2005.
- [26] A. Khosla, S. Kumar, and K. R. Ghosh, “A comparison of computational efforts between particle swarm optimization and genetic algorithm for identification of fuzzy models,” in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '07)*, pp. 245–250, June 2007.
- [27] G. Du, Z. Jiang, X. Diao, Y. Ye, and Y. Yao, “Variances handling method of clinical pathways based on t-s fuzzy neural networks with novel hybrid learning algorithm,” *Journal of Medical Systems*, vol. 36, no. 3, pp. 1283–1300, 2012.
- [28] G. Du, Z. Jiang, X. Diao, and Y. Yao, “Knowledge extraction algorithm for variances handling of CP using integrated hybrid genetic double multi-group cooperative PSO and DPSO,” *Journal of Medical Systems*, vol. 36, no. 2, pp. 979–994, 2012.
- [29] Y. Hou and H. Zhao, “Traffic flow prediction based on T-S fuzzy neural network optimized improved particle swarm optimization,” *Computer Engineering and Applications*, vol. 50, no. 4, pp. 236–239, 2014.
- [30] C. J. Lin, “An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design,” *Fuzzy Sets and Systems*, vol. 159, no. 21, pp. 2890–2909, 2008.
- [31] F. van den Bergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [32] D. Chen, C. Zhao, and H. Zhang, “An improved cooperative particle swarm optimization and its application,” *Neural Computing and Applications*, vol. 20, no. 2, pp. 171–182, 2011.
- [33] L. Wang, “An improved cooperative particle swarm optimizer,” *Telecommunication Systems*, vol. 53, no. 1, pp. 147–154, 2013.
- [34] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, vol. 3, pp. 101–106, 1999.
- [35] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [36] M. Coffin and M. J. Saltzman, “Statistical analysis of computational tests of algorithms and heuristics,” *INFORMS Journal on Computing*, vol. 12, no. 1, pp. 24–44, 2000.



Hindawi

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

