



Endpoint prediction model for basic oxygen furnace steel-making based on membrane algorithm evolving extreme learning machine

Min Han*, Chuang Liu

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning 116024, China



ARTICLE INFO

Article history:

Received 31 August 2012

Received in revised form

18 September 2013

Accepted 27 September 2013

Available online 8 October 2013

Keywords:

Prediction model

Extreme learning machine

Evolutionary membrane algorithm

Soft measurement

Basic oxygen furnace

Endpoint carbon content

Endpoint temperature

ABSTRACT

The endpoint parameters of molten steel, such as the steel temperature and the carbon content, directly affect the quality of the production steel. Moreover, these endpoint results cannot be the online continuous measurement in time. To solve the above-mentioned problems, an anti-jamming endpoint prediction model is proposed to predict the endpoint parameters of molten steel. More specifically, the model is constructed on the parameters of extreme learning machine (ELM) adaptively adjusted by the evolutionary membrane algorithm with the global optimization ability. In other words, the evolutionary membrane algorithm may find the suitable parameters of an ELM model which reduces the incidence of the over-fitting of ELM affected by the noise in the actual data. Finally, the proposed model is applied to predict the endpoint parameters of molten steel in steel-making. In the simulation experiments, two test problems, including 'SinC' function with the Gaussian noise and the actual production data of basic oxygen furnace (BOF) steel-making, are employed to evaluate the performance of the proposed model. The results indicate that the proposed model has good prediction accuracy and robustness in the data with noise. Therefore, the proposed model has good application prospects in the industrial field.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Basic oxygen furnace (BOF) for steel-making is not only an important smelting technology but also the most effective way [1–3]. It is used in about 65% steel factories around the world due to its high productivity and low cost [4,2]. In general, if the endpoint parameters of molten steel can be predicted accurately, the operators can adjust the added amount of the auxiliary raw materials, blowing oxygen and coolant timely [5–9]. And then, the quality of smelting steel can be improved while the production cost can be reduced [10,11]. Therefore, the establishment of a reasonable endpoint prediction model has practical significance to accelerate the development process of the steel industry [12–16].

Nowadays, various models have been proposed to predict the endpoint parameters of molten steel so as to improve the quality of steel-making and reduce the production costs. These models can be roughly divided into three categories, such as mechanism models, online measurement models and intelligent models (soft measurement models). First, based on the material balance and the heat balance, most mechanism models are established for the endpoint parameters of molten steel [17]. The validity of the models usually depends on the stationary operation procedure and on the

stability of ingredients for the steel-making raw material. However, the components of the raw materials are highly volatile in most of steel plants while the operation process usually relies on more human experience, which cause great difficulties to the use of the mechanism models. Second, with the development of the online measurement techniques, a lot of new and advanced sensors and equipments are applied to BOF steel-making in order to improve the control effect [18–21]. But the high cost of the test and maintenance equipment increases directly the cost of the steel. Third, in recent years, some scholars have established some endpoint prediction models of BOF steel-making based on statistics and smart black box models (such as neural network), and have achieved certain research results [22–30].

Extreme learning machine (ELM) is a single hidden layer feed-forward neural network learning algorithm [31]. Unlike the other learning algorithms (such as gradient descent algorithm) for neural networks, ELM only needs to set the number of the hidden nodes, and it does not need setting the parameters of input weights and thresholds, because these parameters are randomly generated in each run. Therefore, ELM has the advantages of fast learning speed, and it is suitable for modeling with real-time requirements [32].

This paper establishes a BOF steel-making endpoint prediction model which employs an evolutionary ELM. Because the endpoint results are affected by many factors in BOF steel-making, most of the common models may not be competent for this prediction. Evolutionary membrane algorithm (EMA) is an optimization

* Corresponding author. Tel.: +86 41184707847; fax: +86 41184707417.
E-mail address: minhan@dlut.edu.cn (M. Han).

technology based on P systems to solve the numerical optimization problems [33]. EMA exhibits a good ability for global search and convergence speed. Because the standard ELM is vulnerable to the impact of outlier, EMA are employed to adaptively adjust the input weights and thresholds of ELM. In other words, the prediction model is established, in which evolutionary membrane algorithm identifies the parameters of ELM. In the simulation experiments, the proposed model is evaluated on the actual production data in BOF steel-making. The output of the model is the endpoint parameters such as the carbon content and the steel temperature. The experimental results show that the prediction model can predict accurately the endpoint parameters of molten steel.

The remainder of this paper is organized as follows. Section 2 presents a brief description of EMA. Section 3 explains the steps of the evolutionary ELM. In Section 4, the details of the proposed prediction model are elaborated. Comprehensive study and experimental results are discussed in Section 5, and finally, Section 6 provides the concluding remarks of the study.

2. Evolutionary membrane algorithm

Taking inspiration by the structure and the function of the biological cells, Păun has proposed a novel distributed and parallel computing model, named as a membrane computing [34,35]. Based on the membrane computing [36], EMA is proposed as a novel global numeric optimization algorithm for solving the optimization problems [33]. A membrane computing model consists of membrane structure, multiset, symbol-objects, and reaction rules. In EMA, the membrane structure, which consists of a skin membrane containing several elementary membranes, represents an execution logic of the whole algorithm. A symbol-object represents a candidate optimal solution of a global optimization problem. Multiset denotes the set of the candidate solutions because it consists of several symbol-objects. Reaction rules, inspired by the irregular Brownian motion, may update the positions of the symbol-objects so that it can help EMA to find the approximate solutions of the global numeric optimization problem. The process of EMA for solving the numeric optimization problems is shown in Fig. 1. The detailed description of EMA is discussed as follows.

Step 1. The parameters of EMA need to be initialized first, such as the maximum iteration number, the number of symbol-objects, the number of elementary membranes.

Step 2. After initializing the parameters of EMA, the skin membrane is created. And then, each symbol-object $S = [S_1, S_2, \dots, S_n]$ is initialized in the feasible region of the optimization problem. n denotes the scale of the symbol-objects. The i -th symbol-object $S_i = [s_{i,1}, s_{i,2}, \dots, s_{i,D}]$ is a candidate solution for an optimization problem. D denotes the dimension of a decision variable in the optimization problem.

Step 3. The fitness of each symbol-object needs to be evaluated according to the objective function of the optimization problem. And then, these symbol-objects are sorted according to their fitness. The sorted symbol-objects are divided equally into some multisets so that each elementary membrane has its own multiset. Finally, the multiset will be sent to different elementary membranes from the skin membrane.

Step 4. The elementary membrane is a basic evolutionary unit. In the region of the elementary membranes, the cellular particles (symbol-objects) are simulated to do irregular Brownian motion. Because the simulated process is very complex and the cellular automata model can simulate from the simple state to the complex phenomenon, EMA introduces the cellular automata model to simulate the complex process. The detail forms are described as follows.

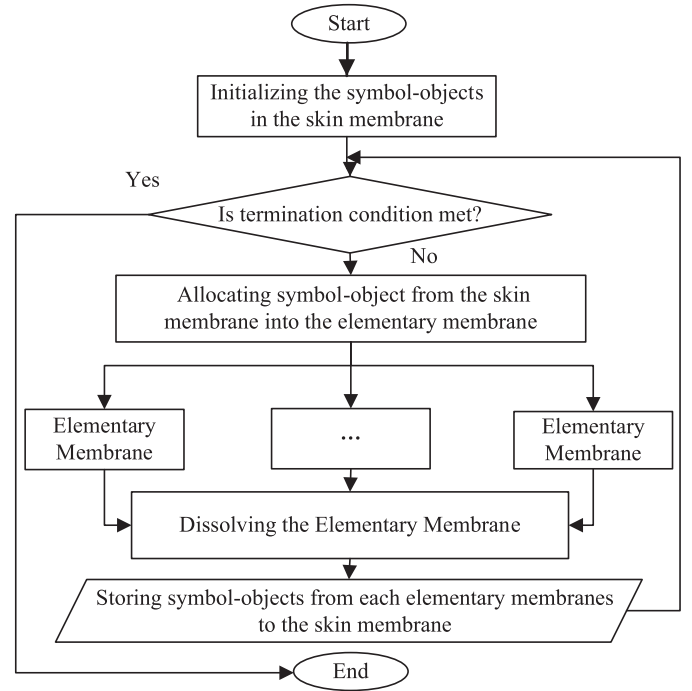


Fig. 1. The flowchart of evolutionary membrane algorithms.

Firstly, the symbol-objects of the multiset (W) in the elementary membrane are mapped into a two-dimensional grid. The row of the grid equals $\lceil \sqrt{\text{sizeof}(W)} \rceil$, where $\text{sizeof}(W)$ represents the number of the symbol-objects in the multiset; at the same time, a state grid is generated with corresponding to the above-mentioned two-dimensional grid. The state value is randomly set as either 0 or 1 in the state grid. Note that the state value 0 represents an inactive cellular particle (symbol-object); otherwise the state value 1 indicates an active cellular particle (symbol-object).

Secondly, a symbol-object is randomly selected from the two-dimensional grid. And then, according to the position of the selected symbol-object, its neighbors are found from the two-dimensional grid. And, its state and its neighbor states are also found from the state grid.

If the state value of the selected symbol-object equals 1 and the sum of its neighbor state value equals 2, then the following rule is executed.

$$\begin{aligned} S_w^1 &= \alpha * S_w^1 + (1 - \alpha) * S_w^2, \\ S_w^2 &= \alpha * S_w^2 + (1 - \alpha) * S_w^1 \end{aligned} \quad (1)$$

where S_w^1 and S_w^2 are two symbol-objects with their state 1, which are two neighbors of the current selected symbol-objects. α is a random number in the interval (0,1).

If the current state value of a symbol-object equals 1 and the sum of its neighbor state value equals 3, then the following rule is executed.

$$\begin{aligned} S_w^1 &= \frac{S_w^1 + S_w^2 + S_w^3}{3} + \alpha * (S_w^1 - S_w^2 - S_w^3), \\ S_w^2 &= \frac{S_w^1 + S_w^2 + S_w^3}{3} + \beta * (S_w^2 - S_w^1 - S_w^3), \\ S_w^3 &= \frac{S_w^1 + S_w^2 + S_w^3}{3} + \gamma * (S_w^3 - S_w^1 - S_w^2) \end{aligned} \quad (2)$$

where S_w^1 , S_w^2 , and S_w^3 are three symbol-objects with their state 1, which are three neighbors of the current selected symbol-objects. α , β , γ are random numbers in the interval (0,1), respectively.

Thirdly, the changing state rule is invoked to update the state of the current symbol-object. The specific form is detailed as the following equation.

$$S_{state} = \begin{cases} 0, & \sum_{i=0}^8 = 2 \text{ or } 3 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

where S_{state} denotes the state of the selected symbol-object. If the sum of its neighbor state is 2 or 3, the state of the current symbol-object is set to 0, otherwise its state is set to 1.

In addition, the operator of the chaotic search is applied to enhance the diversity of the symbol-objects to avoid falling into the local minima. In EMA, Tent Map is employed to implement the chaotic characteristics [37]. Its specific description is shown as follows:

First, a symbol-object is randomly selected from the multiset. And it is mapped from the decision space to the chaotic space by using Formula (4).

$$C^0 = \frac{S - S_l}{S_u - S_l} \quad (4)$$

where S denotes the selected symbol-object. C^0 denotes a chaotic symbol-object which is converted by S . S_l is the lower boundary of a symbol-object. S_u represents the upper boundary of a symbol-object.

Second, the next generation of the chaotic variables C^{k+1} is calculated by C^k according to the Tent Map chaotic systems. The implementation of the Tent Map chaotic systems is discussed as follows.

$$C^{k+1} = \begin{cases} \frac{C^k}{0.7}, & C^k < 0.7 \\ \frac{1}{0.3} * C^k * (1 - C^k), & \text{otherwise} \end{cases} \quad (5)$$

Third, until k in Formula (5) is iterated from 1 to 20, the chaotic variable C^{k+1} is converted back to the normal symbol-object S . Its form is given as follows:

$$S' = S_l + C^{k+1} * (S_u - S_l) \quad (6)$$

Finally, an excellent symbol-object is determined by comparing between S and S' . If the fitness of S' is better than S , the symbol-object S is replaced by S' . Otherwise, the symbol-object S is retained.

Step 5. After executing Step 4, all symbol-objects in the elementary membrane are sent back to the skin membrane. In the skin membrane, the new generation multisets are reconstructed by regrouping the symbol-objects from different elementary membranes. The reorganization of the symbol-objects enables their information from different elementary membranes to be effectively shared. The operator is conducive to find the global optimal solution of the optimization problems and accelerate the convergence speed of EMA.

Step 6. When the end condition is reached, the algorithm execution is terminated. Otherwise, the algorithm will go back to Step3. Finally, in the skin membrane, a symbol-object with the best fitness is chosen as a global optimal approximate solution.

3. Evolutionary ELM

3.1. ELM

ELM is a fast learning algorithm for the single hidden layer feed-forward neural networks (SLFNs). A typical structure of the ELM network is shown in Fig. 2. Unlike the other neural networks, input weights and thresholds of ELM is randomly generated, and then its output weights is determined by using pseudo-inverse [31]. ELM

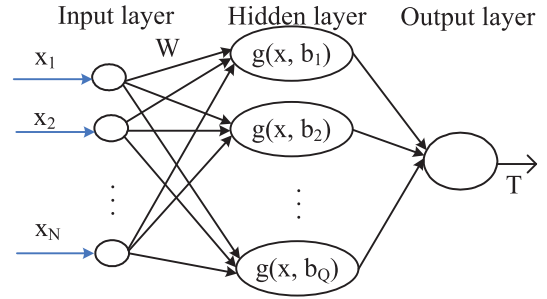


Fig. 2. A structure of ELM.

is trying to provide the best generalization performance with the extreme learning speed. Compared with the traditional algorithms for SLFNs, ELM has better generalization performance. Particularly in learning speed, it is a thousand times faster. Therefore, the study of ELM has received more and more attention from scholars all over the world [38–40]. The description of ELM is given as follows.

For N different random samples (x_i, t_i) , $i = 1, 2, \dots, N$, the mathematical model of SLFNs is given with \bar{N} hidden nodes and activation function $g(x)$ in Formula (7), where $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]^T \in R^n$ and $t_i = [t_{i,1}, t_{i,2}, \dots, t_{i,m}]^T \in R^m$.

$$\sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) = o_j, j = 1, \dots, N \quad (7)$$

where $w_i = [w_{i,1}, w_{i,2}, \dots, w_{i,n}]^T$ is the i -th weight vector for the hidden neurons. $\beta_i = [\beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,m}]^T$ is the i -th weight vector for output neuron weight. b_i is the i -th threshold of the hidden neurons. $w_i \cdot x_j$ represents the internal output of w_i and x_j .

The standard SLFN with \bar{N} hidden nodes can make the predicted value to approximate to the actual value under activation function $g(x)$. That standard SLFNs can approximate these N samples with zero error means that $\sum_{j=1}^N \|o_j - t_j\| = 0$, these exist β_i , w_i and b_i in Formula (8).

$$\sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) = t_j, j = 1, \dots, N \quad (8)$$

The above N equations can be written in matrix as

$$H\beta = T \quad (9)$$

where

$$H(w_1, \dots, w_{\bar{N}}, b_1, \dots, b_{\bar{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) \cdots g(w_{\bar{N}} \cdot x_1 + b_{\bar{N}}) \\ \vdots \\ g(w_1 \cdot x_N + b_1) \cdots g(w_{\bar{N}} \cdot x_N + b_{\bar{N}}) \end{bmatrix}_{N \times \bar{N}} \quad (10)$$

and

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\bar{N}}^T \end{bmatrix}_{\bar{N} \times m}, T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (11)$$

H is called the output matrix of the hidden layer of neural network; The i -th column of H is the output weights of the i -th hidden neuron with input x_1, \dots, x_N .

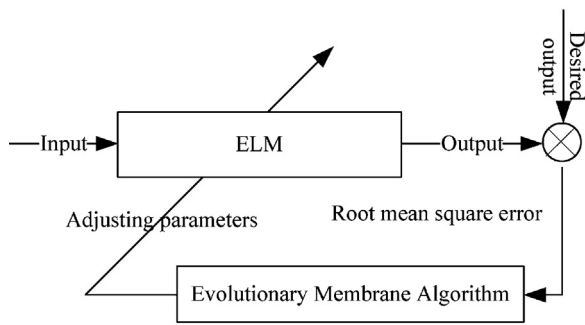


Fig. 3. Identified the parameters of ELM by evolutionary membrane algorithms.

3.2. Improved ELM

Input weights and thresholds of the standard ELM are both randomly generated. However, their random initialization may lead to decrease in the generalization performance of ELM and consume more time to train the model due to adjusting these parameters repeatedly. Scholars all over the world have proposed many improvements to increase the generalization performance of ELM [40–45].

In this paper, the problem that the parameters of ELM are determined adaptively is converted to a global numerical optimization problem. EMA is introduced to solve the numerical optimization problem since EMA is a global optimization algorithm. In other words, EMA is employed to identify the parameters of ELM, called EMA-ELM. Fig. 3 shows the process which EMA identifies the parameters of the ELM model. Fig. 4 depicts the pseudo-codes of the main steps of the optimal model.

The implementation process, which EMA identifies the parameters of ELM, is further described as follows. First, the sample data is chosen as the input node of the ELM model. Second, Root mean square error (RMSE) is used to evaluate the precision of the model which can be calculated in Formula (12). Third, according to the RMSE value, EMA updates the information of the symbol-objects so that the parameters of ELM can be adaptively adjusted. Finally, the learning process will be performed continuously until the termination condition is reached.

$$rmse = \sqrt{\frac{\sum_{i=1}^n (p_i - r_i)^2}{n}} \quad (12)$$

Input: the sample data
Output: the optimal structure of ELM

Step1. The parameters are configured, such as evolutionary membrane algorithm and the number of hidden node of ELM.

Step 2. Symbol-objects will be initialized under meeting the constraints of input weights and the threshold.

Step 3. The symbol-object is assigned to initialize the ELM model, and the root mean square error is calculated as the fitness of the symbol-object.

Step 4. According to their fitness, the symbol-objects are sorted. These sorted symbol-objects are divided equally to several multisets. The multiset is sent to the elementary membrane.

Step 5. In the elementary membrane, the symbol-object will be evolved by invoking the operator of cellular automata and the operator of chaotic search.

Step 6. The evolved symbol-objects in each elementary membranes are sent back to the skin membrane.

Step 7. The terminated condition dose not be reached, Step3 will be performed continuously.

Step 8. The optimal symbol-object in the skin membrane is final the structure of ELM network.

Fig. 4. The pseudo-codes of the proposed model.

where p_i is the prediction value. r_i is the practical value. n is the number of the predicted samples.

4. Endpoint prediction model for BOF based on evolutionary ELM

In the steel industry, whether the quality of the steel is good or not directly relates to the endpoint parameters of molten steel, such as the carbon content and the steel temperature. The accurate endpoint prediction can not only improve the quality of the steel and increase the types of the steel, but also shorten the smelting time, save energy and reduce production costs. Therefore, the effectiveness of the predicting endpoint model is imperative to improve the level of the steel production.

So far, the common approaches are to determine the endpoint parameters of molten steel by the operators' experience. The approaches make the operating workers to work hard and the steel with poor quality. To solve the above-mentioned problems, an endpoint prediction model is established in BOF steel-making, in which EMA identifies the parameters of ELM. The model, with some advantages such as low-cost, fast-response, and high-precision, is very suitable for the application of the endpoint prediction in BOF steel-making.

4.1. Endpoint prediction model for carbon content and temperature

This paper presents a prediction model, and applies it to predict the endpoint parameters of molten steel. Compared with the original ELM, evolutionary ELM improves ELM with a better generalization performance about the endpoint prediction in BOF steel-making. Fig. 5 shows the structure of the prediction model.

4.1.1. Initialization

First of all, the parameters of EMA and ELM are initialized respectively, such as the number of symbol-objects, the number of elementary membranes, the end conditions, and the constraint boundary including input weights, thresholds and hidden nodes. When EMA optimizes the parameters of ELM, the parameters of ELM is encoded as a symbol-object of EMA. In other words, a symbol-object represents a whole set of network parameters of ELM, which consists of w_1, \dots, w_N and $\theta_1, \dots, \theta_N$. w_1, \dots, w_N denote the input weights of ELM, which are initialized on the interval $[-1, 1]$. $\theta_1, \dots, \theta_N$ are the thresholds of ELM, which are initialized on the interval $[-1, 1]$.

4.1.2. The establishment of endpoint prediction model

As depicted in Fig. 5, the endpoint prediction model is created by ELM. From the figure, the input data of ELM consists of weight of hot metal, weight of scrap, weight of lime, weight of mixing material, weight of iron sheet, weight of ore, weight of dolomite, volume of blowing oxygen, process carbon measured by sub-lance, and process temperature measured by sub-lance. The hidden layer of ELM contains 20 hidden nodes. When the endpoint steel temperature is predicted, the output of ELM is the predicting value of the endpoint steel temperature; otherwise, the output of ELM is the predicting value of the endpoint carbon content.

The sample data of ELM is determined through the statistical analysis of the actual production data in BOF. Before training samples and testing samples are utilized, they are firstly normalized in order to avoid the annihilation caused by the different scales of the sample data. In order to get the right prediction results, the output of the predicted model need to be anti-normalized. According to the

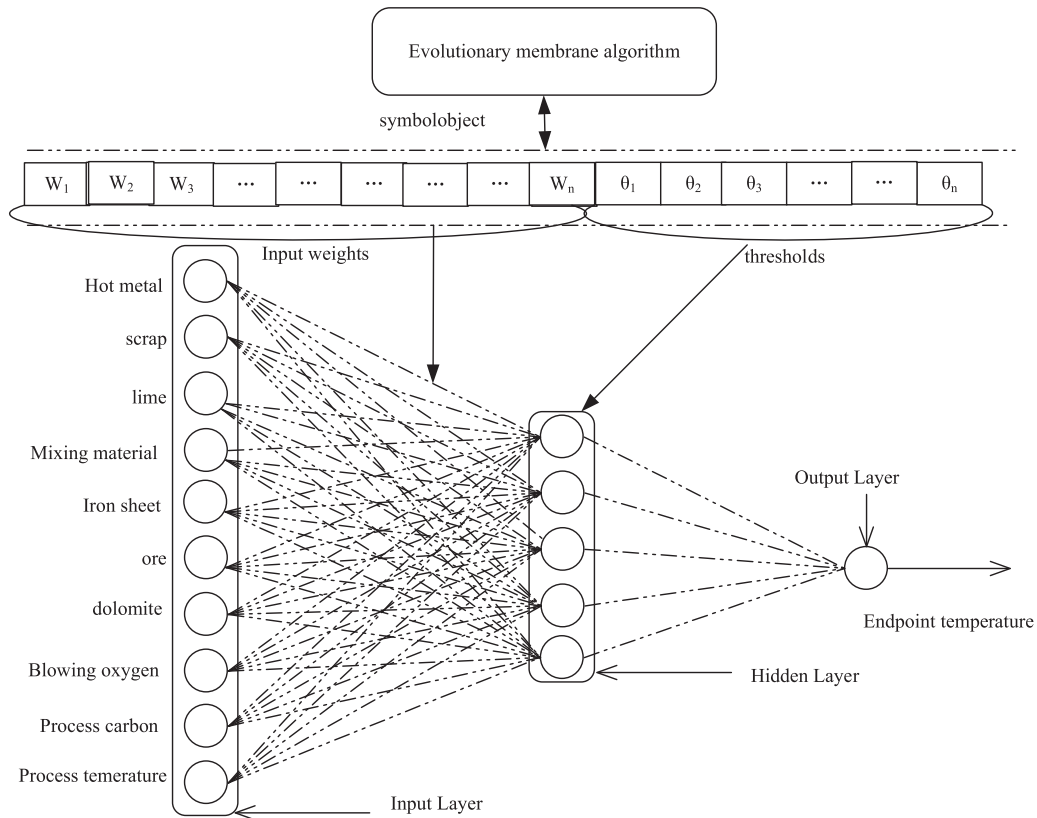


Fig. 5. The prediction model of ELM based on membrane algorithms.

characteristics of the actual data in steel-making, the normalized input equation is designed as shown in Formula (13).

$$x'_i = \frac{x_i - x^l}{x^u - x^l} \quad (13)$$

where x'_i is the normalized input data. x_i is the actual input data. x^l is a lower boundary of x_i . x^u is the upper boundary of x_i .

According to the normalized Formula (13), the anti-normalization equation is designed as shown in Formula (14).

$$x'_o = x^l + x_o * (x^u - x^l) \quad (14)$$

where x'_o is an anti-normalized output. x_o is an output of the prediction model. x^l is the lower boundary of x_o . x^u is the upper boundary of x_o .

4.1.3. The adjustment of input weights and threshold

The production process in BOF steel-making is very complex physical and chemical change with more uncertainty. In Section 4.1.2, the endpoint prediction model is established. Then, EMA is employed to find the input weights and thresholds of ELM. Finally, the global optimal symbol-object will be found, which represents an ELM model with a good generalization performance.

5. Experimental study and discussions

To test the validity and applicability of EMA optimizing the parameters of ELM, two groups of data are used in the simulation experiments. One group is a standard 'SinC' function with the noise, which detects the regression ability of the proposed model; another group is some actual production data collected from BOF in a steel plant. The parameters of EMA are shown in Table 1.

Table 1
The parameters setting of EMA.

Parameters	Iteration	Symbol-objects	Elementary membrane
Set value	20	100	5

5.1. The experiments of 'SinC'

'SinC' is a sin function $\sin(x)$ multiplied by a monotonically decreasing function $1/x$, as shown in Formula (15). The regression performance of all predicted models may be tested by the function. At first, 200 data points with the noise of $\mu(0, 0.3)$ which is a Gaussian distribution function are generated by the 'SinC' function as the training samples. And 200 data points are generated in the same way as the testing samples. Then, the training samples are employed to test ELM [31], DE-ELM (a hybrid learning algorithm which uses the differential evolutionary algorithm (DE) to select the input weights and Moore Penrose (MP) generalized inverse to analytically determine the output weights.) [46], and the proposed model EMA-ELM. The number of hidden nodes of all predicted models are set to 10. Finally, test samples are utilized to test the trained models. The test results are shown in Fig. 6.

$$\text{SinC}(x) = \frac{\sin(x)}{x}, -10 < x < 10 \quad (15)$$

As seen in Fig. 6, the proposed model EMA-ELM has better approximation to 'SinC' function than ELM, and DE-ELM. The results indicate that the proposed model has a good regression ability, and it is proved that EMA is effective for improving the generalization performance of ELM. In order to further illustrate the performance of three models, these models are run 50 times, and Tables 2 and 3 show the statistical predicted results from different models. As shown in Tables 2 and 3, the results achieved by the proposed

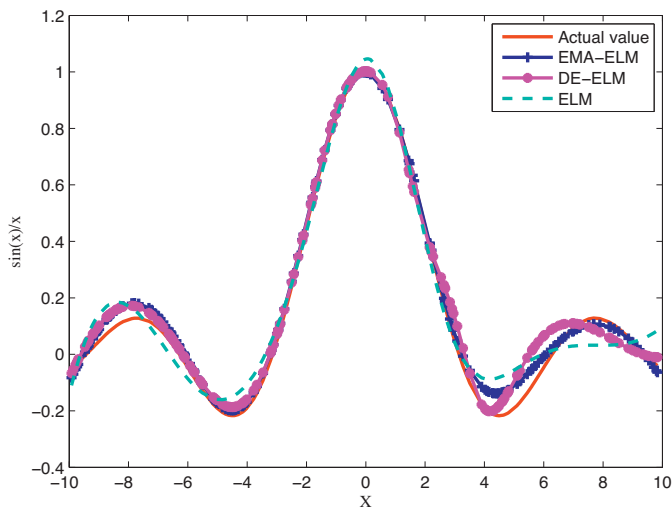


Fig. 6. The results of 'SinC' functions.

Table 2

The statistical training error of prediction models.

	Min	Mean	Median	Max	Std
EMA-ELM	0.2677	0.2988	0.2995	0.3284	0.0135
DE-ELM [46]	0.2672	0.3012	0.3006	0.3335	0.0148
ELM [31]	0.2665	0.2950	0.2952	0.3263	0.0139

Table 3

The statistical testing error of prediction models.

	Min	Mean	Median	Max	Std
EMA-ELM	0.0285	0.0713	0.0716	0.1344	0.0221
DE-ELM [46]	0.0285	0.0814	0.0806	0.1424	0.0258
ELM [31]	0.0393	0.0741	0.0734	0.1139	0.0177

model are superior to the other models. The results indicate that the proposed model has good stability and robustness.

5.2. The experiments based on steel-making dataset

The dataset consists of the measurement results for two-stage, the main blowing stage and auxiliary blowing stage. There are many factors affecting the endpoint results. More specifically, the endpoint results are affected by weight of molten iron, weight of scrap, weight of lime, weight of mixing material, weight of iron sheet, weight of ore, weight of dolomite, the process temperature measured by sub-lance, the process carbon content measured by sub-lance, and volume of the second-blowing oxygen.

Generally speaking, molten steel is formed by the decarburization and rising temperature of molten iron. More specifically, the content of carbon is reduced from 4% to 0.08% or less, while the temperature is increased from 1250 °C to 1650 °C. When the error of endpoint carbon content satisfies $|\Delta C| < 0.05\%$, the prediction carbon content can be accepted. When the error of endpoint temperature of the prediction model satisfies $|\Delta T| < 15^\circ\text{C}$, the prediction temperature can be accepted.

In simulation, there are 60 groups of the actual production data from the steel plant. Among them, 30 groups sample data are selected randomly as the training samples for the endpoint prediction model, while the remaining samples as the testing samples. The number of hidden nodes of all prediction models is set to 20.

The predicting results for the endpoint carbon content and the endpoint temperature are shown in Figs. 7 and 8, respectively. As observed in Figs. 7 and 8, the proposed model for predicting the

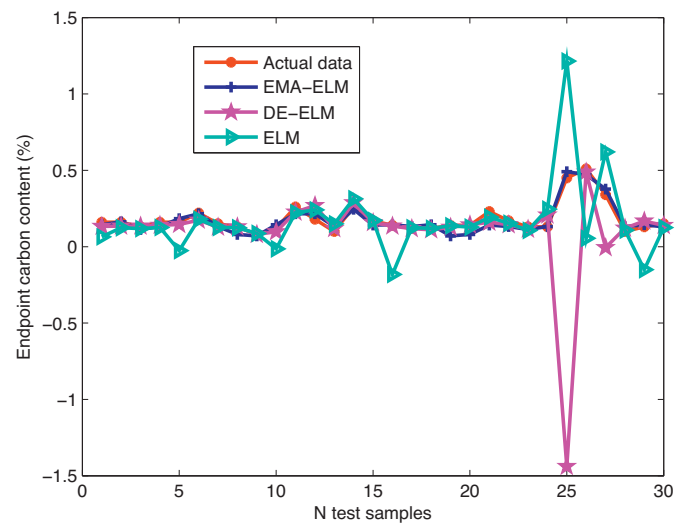


Fig. 7. Predicting the endpoint carbon content.

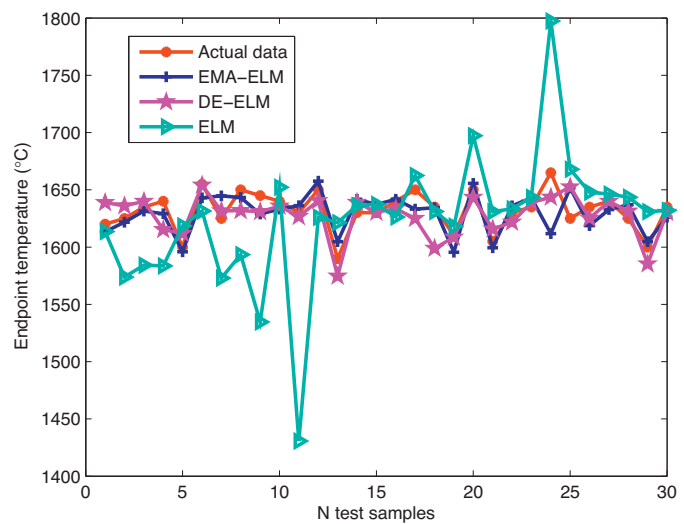


Fig. 8. Predicting the endpoint temperature.

carbon content and the temperature has better results than the other models at the blowing endpoint.

Hit is used to judge whether the prediction results satisfy the request of practical production or not. To further test the predicting capability of the experimental models, two evaluating indicators of hit rate are designed. These indicators are shown in Formula (16).

$$C_{hit} = \frac{N_c}{N_f}, \quad T_{hit} = \frac{N_t}{N_f} \quad (16)$$

where N_c represents the number of the furnace with meeting the error constraints of the carbon content. C_{hit} is the hit ratio of carbon content on the total number of furnaces. N_t represents the number of furnaces which meet the error constraints of the temperature. T_{hit} denotes the hit ratio of temperature on the total number of furnaces. N_f is the sum of all furnaces.

As observed in Tables 4 and 5, compared with the statistical results of the other two models, the proposed model achieves more better results on the endpoint hit rate for both the carbon content and the temperature. The results indicate that EMA can find the suitable parameters of ELM, thereby the proposed model not only has higher prediction accuracy but also has good robustness.

Table 4

The hit rate of the endpoint of temperature.

	Min	Mean	Media	Max	Std
EMA-ELM	60.0000	78.6000	80.3333	93.3333	7.7749
DE-ELM [46]	53.3333	75.1333	76.6667	90.0000	6.8416
ELM [31]	26.6667	53.8000	55.0000	76.6667	11.3091

Table 5

The hit rate of the endpoint of carbon content.

	Min	Mean	Media	Max	Std
EMA-ELM	59.3333	76.7333	79.6667	93.3433	9.0110
DE-ELM [46]	56.6667	76.6667	76.6667	93.3333	8.6766
ELM [31]	33.3333	61.2667	61.6667	83.3333	9.5687

6. Conclusions

This paper presents a novel endpoint prediction model in BOF steel-making, which EMA identifies the parameters of ELM. The proposed model employs EMA to adaptively learn the parameters such as input weights and thresholds of ELM, and thus reduce overfitting and poor robustness causing by those parameters generated randomly of ELM. In the standard ‘SinC’ function with some noise, the proposed model not only has good regression ability, but also has good stability and robustness. Then, the actual production data in BOF steel-making is utilized, and the endpoint temperature model and the endpoint carbon content model are established, respectively. The good hit rate on both the temperature and the carbon content of molten steel has been attained, which indicates that the proposed model can be used for the endpoint prediction in the actual BOF steel-making. The proposed model is not only applied to the endpoint prediction in BOF, but also applied to the other areas of industrial production.

Acknowledgements

This research is supported by the project (61074096) of the National Nature Science Foundation of China, the project (2007AA04Z158) of the National High Technology Research and Development Program of China (863 Program).

References

- [1] R. Fruehan, Overview of steelmaking processes and their development, in: *The Making, Shaping and Treating of Steel: Steelmaking and Refining*, 1998.
- [2] M. Han, Y. Zhao, Dynamic control model of BOF steelmaking process based on ANFIS and robust relevance vector machine, *Expert Systems with Applications* 38 (12) (2011) 14786–14798.
- [3] X. Wang, M. Han, J. Wang, Applying input variables selection technique on input weighted support vector machine modeling for BOF endpoint prediction, *Engineering Applications of Artificial Intelligence* 23 (6) (2010) 1012–1018.
- [4] A. Ghosh, A. Chatterjee, *Ironmaking and Steelmaking: Theory and Practice*, Prentice-Hall of India, 2008.
- [5] F. Pettersson, H. Saxen, K. Deb, Genetic algorithm-based multicriteria optimization of ironmaking in the blast furnace, *Materials and Manufacturing Processes* 24 (3) (2009) 343–349.
- [6] H. Saxén, F. Pettersson, K. Gunturu, Evolving nonlinear time-series models of the hot metal silicon content in the blast furnace, *Materials and Manufacturing Processes* 22 (5) (2007) 577–584.
- [7] F. Pettersson, N. Chakraborti, H. Saxén, A genetic algorithms based multi-objective neural net applied to noisy blast furnace data, *Applied Soft Computing* 7 (1) (2007) 387–397.
- [8] M. Sayed-Mouchaweh, E. Lughofer, *Learning in Non-Stationary Environments: Methods and Applications*, Springer, New York, 2012.
- [9] M. Han, L. Jiang, Endpoint prediction model of basic oxygen furnace steelmaking based on PSO-ICA and RBF neural network, in: *2010 International Conference on Intelligent Control and Information Processing (ICICIP)*, IEEE, 2010, pp. 388–393.
- [10] G. Brooks, N. Dogan, M. Alam, J. Naser, M. Rhamdhani, Developments in the modelling of oxygen steelmaking, in: *Guthrie Honorary Symposium, 6th C9*, June 2011, Montreal, QC, Canada, 2011.
- [11] R. Kornblum, M. Tribus, The use of bayesian inference in the design of an end-point control system for the basic oxygen steel furnace, *IEEE Transactions on Systems Science and Cybernetics* 6 (4) (1970) 339–348.
- [12] P. Chen, Y.Z. Lu, Memetic algorithms-based neural network learning for basic oxygen furnace endpoint prediction, *Journal of Zhejiang University Science A* 11 (11) (2010) 841–848.
- [13] I. Cox, R. Lewis, R. Ransing, H. Laszczewski, G. Berni, Application of neural computing in basic oxygen steelmaking, *Journal of Materials Processing Technology* 120 (1) (2002) 310–315.
- [14] A. Cunha, T. Pacianotto, A. Frattini Fileti, Steelmaking process: neural models improve end-point predictions, *Computer Aided Chemical Engineering* 18 (2004) 631–636.
- [15] A. Fileti, T. Pacianotto, A.P. Cunha, Neural modeling helps the BOS process to achieve aimed end-point conditions in liquid steel, *Engineering Applications of Artificial Intelligence* 19 (1) (2006) 9–17.
- [16] N. Rajesh, M.R. Khare, S.K. Pabi, Feed forward neural network for prediction of end blow oxygen in LD converter steel making, *Materials Research* 13 (1) (2010) 15–19.
- [17] F. De Medeiros, S. Noblat, A. Fileti, Reviving traditional blast furnace models with new mathematical approach, *Ironmaking & Steelmaking* 34 (5) (2007) 410–414.
- [18] L.F. Xu, W. Li, M. Zhang, S.X. Xu, J. Li, A model of basic oxygen furnace (BOF) end-point prediction based on spectrum information of the furnace flame with support vector machine (SVM), *Optik-International Journal for Light and Electron Optics* 122 (7) (2011) 594–598.
- [19] L.F. Xu, W.S. Li, S.X. Xu, J. Li, Y.Q. Wang, A new spectral analysis technique used in converter steelmaking BOF endpoint control, *Advanced Materials Research* 139 (2010) 689–692.
- [20] H.Y. Wen, Q. Zhao, Y.R. Chen, M.C. Zhou, M. Zhang, L.F. Xu, Converter end-point prediction model using spectrum image analysis and improved neural network algorithm, *Optica Applicata* 38 (4) (2008) 693–704.
- [21] Y. Iida, Fully automatic blowing technique for basic oxygen steelmaking furnace, *Transactions of the Iron and Steel Institute of Japan* 24 (7) (1984) 540–546.
- [22] W. Birk, A. Johansson, A. Medvedev, R. Johansson, Model-based estimation of molten metal analysis in the LD converter: experiments at SSAB Tunplatt AB in Lulea, *IEEE Transactions on Industry Applications* 38 (2) (2002) 565–570.
- [23] B.Y. Cai, H. Zhao, Y.J. Yue, Research on the BOF steelmaking endpoint temperature prediction, in: *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, IEEE, 2011, pp. 2278–2281.
- [24] K.X. Peng, J. Zhang, J. Dong, Intelligent method in the end-point control of BOF, *Applied Mechanics and Materials* 20 (2010) 796–800.
- [25] A. Bouchachia, E. Lughofer, D. Sanchez, Editorial of the special issue: online fuzzy machine learning and data mining, *Information Sciences* 220 (2013) 1–4.
- [26] P. Angelov, D.P. Filev, N. Kasabov, *Evolving Intelligent Systems: Methodology and Applications*, vol. 12, Wiley.com, 2010.
- [27] E. Lughofer, *Evolving Fuzzy Systems—Methodologies, Advanced Concepts and Applications*, Springer, 2011.
- [28] A. Bouchachia, N. Nedjah, Adaptive incremental learning in neural networks, *Neurocomputing* 74 (11) (2011) 1783–1784.
- [29] J. Gama, P.P. Rodrigues, E.J. Spinosa, A.C.P.L.F. de Carvalho, *Knowledge Discovery from Data Streams*, Citeseer, 2010.
- [30] K. Kasabov, N. Kasabov, *Evolving Connectionist Systems: The Knowledge Engineering Approach*, Springer, 2007.
- [31] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70 (1) (2006) 489–501.
- [32] G. Li, P. Niu, C. Liu, W. Zhang, Enhanced combination modeling method for combustion efficiency in coal-fired boilers, *Applied Soft Computing* 12 (10) (2012) 3123–3140.
- [33] C. Liu, M. Han, X.Z. Wang, A novel evolutionary membrane algorithm for global numerical optimization, in: *2012 Third International Conference on Intelligent Control and Information Processing (ICICIP)*, IEEE, 2012, pp. 727–732.
- [34] G. Păun, Computing with membranes, *Journal of Computer and System Sciences* 61 (1) (2000) 108–143.
- [35] G. Păun, G. Rozenberg, A. Salomaa, *The Oxford Handbook of Membrane Computing*, Oxford University Press, Inc, 2010.
- [36] G. Ciobanu, G. Păun, M.J. Pérez-Jiménez, *Applications of Membrane Computing*, Springer, 2006.
- [37] R. Caponetto, L. Fortuna, S. Fazzino, M. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* 7 (3) (2003) 289–304, <http://dx.doi.org/10.1109/TEVC.2003.810069>.
- [38] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42 (2) (2012) 513–529.
- [39] R. Zhang, Y. Lan, G.-B. Huang, Z.-B. Xu, Universal approximation of extreme learning machine with adaptive growth of hidden nodes, *IEEE Transactions on Neural Networks and Learning Systems* 23 (2) (2012) 365–371.
- [40] S. Suresh, R. Venkatesh Babu, H. Kim, No-reference image quality assessment using modified extreme learning machine classifier, *Applied Soft Computing* 9 (2) (2009) 541–552.
- [41] S. Saraswathi, S. Sundaram, N. Sundararajan, M. Zimmermann, M. Nilsen-Hamilton, ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins

- are highly represented, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8 (2) (2011) 452–463.
- [42] S.S. Kim, K.C. Kwak, Incremental modeling with rough and fine tuning method, *Applied Soft Computing* 11 (1) (2011) 585–591.
- [43] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, *IEEE Transactions on Neural Networks* 21 (1) (2010) 158–162.
- [44] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Transactions on Neural Networks* 17 (4) (2006) 879–892.
- [45] N. Liu, H. Wang, Weighted principal component extraction with genetic algorithms, *Applied Soft Computing* 12 (2) (2012) 961–974.
- [46] Q.Y. Zhu, A.K. Qin, P.N. Suganthan, G.B. Huang, Evolutionary extreme learning machine, *Pattern Recognition* 38 (10) (2005) 1759–1763.