Applied Soft Computing 19 (2014) 430–437

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

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 overfitting 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.

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 feedforward 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).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.09.012

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, Pău n 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 $S=[S_1, S_2, \ldots, S_n]$ is initialized in the feasible region of the optimization problem. n denotes the scale of the symbol-objects. The i-th symbol-object $S_i=[s_{i,1}, s_{i,2}, \ldots, s_{i,D}]$ 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.

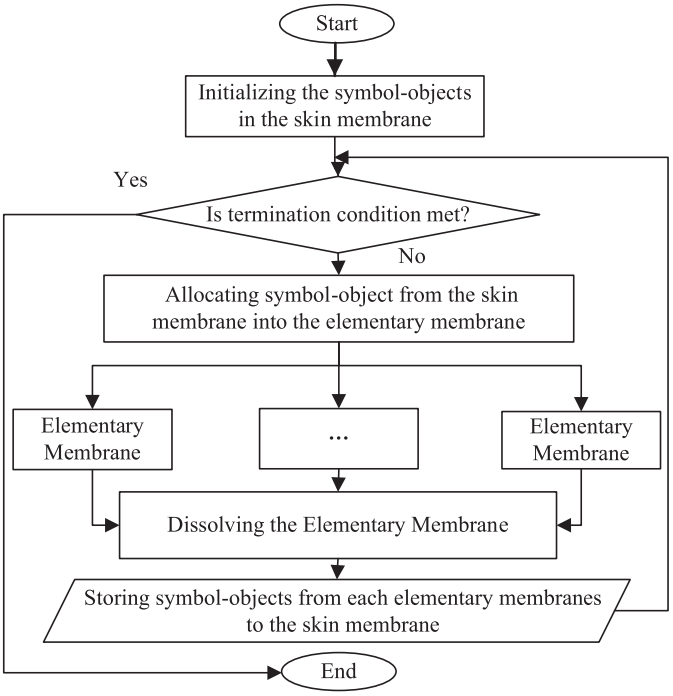

Fig. 1. The flowchart of evolutionary membrane algorithms.

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 $\lceil\sqrt{sizeof(W)}\rceil$, where $sizeof(W)$ 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.

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

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.

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

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.