Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Memetic algorithms-based neural network learning for basic oxygen furnace endpoint prediction*

Peng CHEN[1], Yong-zai LU‡[2]

([1]*Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China*)

([2]*Department of Automation, Zhejiang University, Hangzhou 310027, China*)

E-mail: pengchen@sjtu.edu.cn; y.lu@ieee.org

Received Oct. 31, 2009; Revision accepted July 27, 2010; Crosschecked Oct. 12, 2010

Abstract: Based on the critical position of the endpoint quality prediction for basic oxygen furnaces (BOFs) in steelmaking, and the latest results in computational intelligence (CI), this paper deals with the development of a novel memetic algorithm (MA) for neural network (NN) learning. Included in this is the integration of extremal optimization (EO) and Levenberg-Marquardt (LM) gradient search, and its application in BOF endpoint quality prediction. The fundamental analysis reveals that the proposed EO-LM algorithm may provide superior performance in generalization, computation efficiency, and avoid local minima, compared to traditional NN learning methods. Experimental results with production-scale BOF data show that the proposed method can effectively improve the NN model for BOF endpoint quality prediction.

Key words: Memetic algorithm (MA), Neural network (NN) learning, Back propagation (BP), Extremal optimization (EO), Levenberg-Marquardt (LM) gradient search, Basic oxygen furnace (BOF)

doi:10.1631/jzus.A0900664 **Document code:** A **CLC number:** TP183

1 Introduction

The basic oxygen furnace (BOF) is one of the key processes in iron and steel production. It is a typical complex batch chemical reactor with sophisticated thermal and chemical reaction processes, which convert liquid pig iron into steel with desired grade (temperature and chemical compositions, such as carbon, etc.) under oxidation conditions. The BOF steelmaking endpoint quality control is always managed with the aids of either operator experience or a first principle charge model; therefore, often resulting in a relatively poor hit ratio of the bath endpoint. The accurate prediction of the endpoint quality is vital not only for BOF operation, but also for "steelmaking-caster-hot rolling" production balance. However, due to the imprecision of the BOF charge model, lack of real-time measurements and process uncertainties, it is difficult to predict the BOF endpoint product precisely (Cox *et al*., 2002; Wang *et al*., 2006). In addition to the first principle models, the supervised learning with a neural network (NN) has played an important role in BOF endpoint quality prediction because of its capability in nonlinear mapping and simplicity in implementation (Cox *et al*., 2002; Tao *et al*., 2002; Fileti *et al*., 2006; Wang *et al*., 2006; Feng *et al*., 2008). Due to the inherent defects, however, the popular back propagation (BP) NN training algorithms based on gradient search often suffer from local minima, being sensitive to initial weights and poor generalization (Haykin, 1994; Yao and Islam, 2008).

Recent research results in the bio-inspired computational intelligence (CI) methods have motivated researchers to use them in NN learning because of their superior capabilities in solving high-dimensional

‡ Corresponding author

* Project (No. 60721062) supported by the National Creative Research Groups Science Foundation of China

nonlinear optimization problems with global search ability for complex systems (Yao and Islam, 2008; Dengiz *et al*., 2009; Pendharkar, 2009; Sedki *et al*., 2009). The applications of genetic algorithms in NN learning for metallurgical process have been reported, but only limited studies have been published for the BOF process (Feng *et al*., 2008). It should be noted that the CI methods are rather inefficient in fine-tuned local search; although they are good at global searches. In recent years, a particular class of global-local search hybrids, with both efficiency and robustness, named "memetic algorithms (MAs)" have been proposed. Motivated by Dawkins (1976)'s concept of a meme representing a unit of cultural evolution that can exhibit local refinement, MA combines the global search nature of CI with local search methods to improve solutions (Cotta and Moscato, 2007) in which the rules governing global/local search are co-evolved and self-adapted alongside the problem representation within a coupled gene-meme evolutionary system (Smith, 2007). The no-free-lunch theorem (Wolpert and Macready, 1997) clearly underpins the exploitation of problem knowledge intrinsic to MAs by stating that "a search algorithm strictly performs in accordance with the amount and quality of the problem knowledge they incorporate". The MAs have been successfully applied to hundreds of real-world problems such as nonlinear programming (NLP) optimization (Digalakis and Margaritis, 2004), vehicle routing problems (Tavakkoli-Moghaddam *et al*., 2006), nurse rostering problems (Ozcan, 2007), bioinformatics (Zhu *et al*., 2009), etc. Consequently, the MA-based hybrid optimization solutions are also applicable for the improvement of NN learning processes.

Based on the complexity of nonlinear optimization involved in NN learning, this study deals with the development of a novel MA-based hybrid method called "EO-LM" learning algorithm, which combines the extremal optimization (EO) proposed by Boettcher and Percus (1999) with the popular Levenberg-Marquardt (LM) gradient search algorithm (Hagan and Menhaj, 1994). The proposed EO-LM learning is further applied to the endpoint quality prediction of the BOF process. Industrial test results show that the performance of the BOF endpoint quality prediction model can be improved by this algorithm.

2 Problem formulation of neural network supervised learning with computational intelligence

For a feed-forward multilayer perceptron (MLP) network with a single hidden layer shown in Fig. 1, the log-sigmoid function and linear activation function are selected as the transfer functions for hidden and output layers, respectively. The mapping from the jth input x_j (j=1, 2, …, m) to the ith output of the NN $\hat{y}_i$ (i=1, 2, …, n) can be written as

$$\begin{aligned}\hat{y}_i &= f(\boldsymbol{X},\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{r})\\ &= \sum_{k=1}^{p}(v_{ki}z_k + r_i)\\ &= \sum_{k=1}^{p}\left(v_{ki}\log\mathrm{sig}\left(\sum_{j=1}^{m}\omega_{jk}x_j+\theta_k\right)+r_i\right),\\ &\qquad i=1,2,...,n,\end{aligned} \tag{1}$$

where z_k is the kth hidden layer variable (k=1, 2, …, p); ω_{jk} is the weight linking the jth input variable with the kth hidden layer variable; v_{ki} is the weight linking the kth hidden layer variable with the ith output variable; θ_k is the bias of the kth hidden layer variable; r_i is the bias of the ith output variable; log sig is transfer function log sig (a)=1/[1+exp(−a)].

In NN training, the learning samples are often divided into a training dataset and a validation dataset. The aim of this study is to develop a novel MA-based hybrid EO-LM approach to optimize the synaptic connections and biases of an MLP network, which may provide good generalization performance with minimum output errors for both the training and the validation samples:

$$\begin{aligned}&\min_{s.t.\ \boldsymbol{w}\in\mathbb{R}^{m\times p},\boldsymbol{v}\in\mathbb{R}^{p\times n},\boldsymbol{\theta}\in\mathbb{R}^{p},\boldsymbol{r}\in\mathbb{R}^{n}} E(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{r})\\ &=\min\left[\sum_{i=1}^{n}\sum_{l=1}^{n_{\_\mathrm{Train}}}(y_i^l-\hat{y}_i^l)^2+\sum_{i=1}^{n}\sum_{l=1}^{n_{\_\mathrm{Valid}}}(y_i^l-\hat{y}_i^l)^2\right],\end{aligned} \tag{2}$$

where $n_{\_\mathrm{Train}}$ and $n_{\_\mathrm{Valid}}$ represent the numbers of the training data and the validation data, respectively. Here, $\boldsymbol{w}$, $\boldsymbol{v}$, $\boldsymbol{\theta}$, and $\boldsymbol{r}$ are bounded by searching space of the optimization algorithm, and y_i represents the ith desired output.

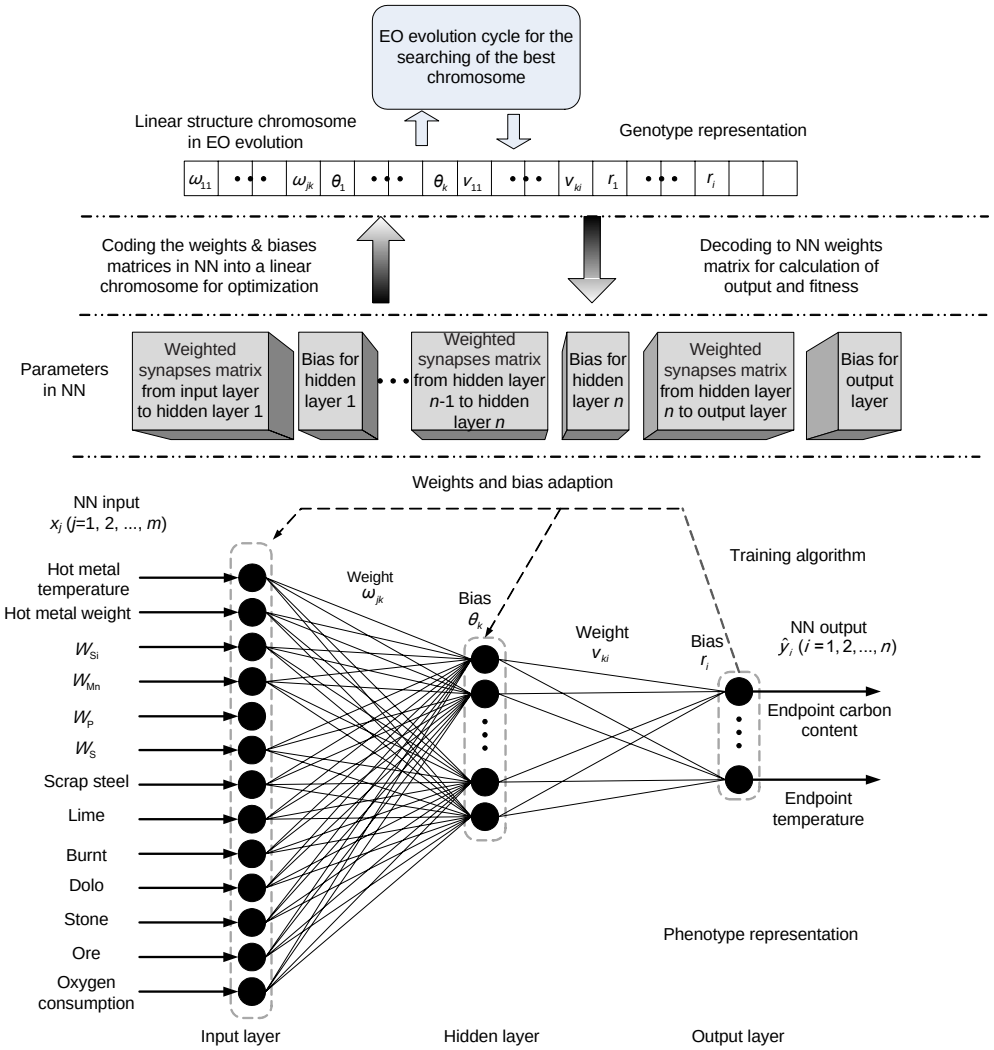

Fig. 1 Hybrid EO-LM learning algorithm structure diagram of neural network model for BOF endpoint prediction
W_{Si}, W_{Mn}, W_P, and W_S are the weight ratios of Si, Mn, P, and S in hot metal, respectively

3 EO-LM algorithm for neural network supervised learning

3.1 Extremal optimization

Based on the Bak-Sneppen (BS) model which simulates far from equilibrium dynamics in statistical physics, the EO proposed by Boettcher and Percus (1999) is derived from the fundamentals of statistical physics and self-organized criticality (SOC) (Bak *et al*., 1987). SOC states that large interactive systems evolve to a state where a change in one single of their elements may lead to avalanches or domino effects that can reach any other element in the system. For an optimization problem with n decision variables, EO proceeds as follows (Boettcher and Percus, 2000):

Step 1. Initialize a configuration S at will, set $S_{best}=S$.

Step 2. For the "current" solution S.

(a) Evaluate the fitness for each decision variable (component) x_i.

(b) Rank all the components by their fitness and find the component with the "worst fitness".

(c) Choose one solution S' in the neighborhood of S, i.e., such that the worst component x_j must change its state.

(d) Accept $S=S'$ unconditionally.

(e) If $F(S)<F(S_{best})$, set $S_{best}=S$.

Step 3. Repeat Step 2 as long as desired.

Step 4. Return S_{best} and $F(S_{best})$.

Generally speaking, EO is particularly applicable

to large complex problems with rough landscape, phase transitions or multiple local optima. It is less likely to be trapped in local minima than the traditional gradient-based search. The research results by previous studies have shown that EO and its derivatives can be effectively applied in solving combinatory and multi-objective hard benchmarks with real-world optimization problems (Chen M.R. *et al*., 2007; Chen Y.W. *et al*., 2007; Lu *et al*., 2007; Chen and Lu, 2008).

3.2 Levenberg-Marquardt algorithm

The LM gradient search algorithm has been introduced to MLP network training to provide better performance (Hagan and Menhaj, 1994). Basically, the LM gradient search is a Hessian-based algorithm for nonlinear least squares optimization without having to compute the Hessian matrix. Under the assumption that the error function is some kind of squared sum, then the Hessian matrix can be approximated as

$$\boldsymbol{H} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\,, \tag{3}$$

and the gradient information can be computed as follows:

$$\boldsymbol{g} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{e}, \tag{4}$$

where $\boldsymbol{J}$ is the Jacobian matrix that contains first derivatives of the network errors with respect to weights and biases, and $\boldsymbol{e}$ is an error vector. The Jacobian matrix can be computed through a standard BP technique that is much less complex than computing the Hessian matrix (Hagan and Menhaj, 1994).

The LM algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + \mu I]^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{e}, \tag{5}$$

where μ is a scalar controlling the behavior of the algorithm. The convergence behavior of the LM is similar to the Gauss-Newton method. Near a solution with small residuals, it performs well and gives a very fast convergence rate; while for the large-residual cases, the performances of the Gauss-Newton and LM algorithms are usually poor (Nocedal and Stephen, 2006).

3.3 Hybrid EO-LM solution for BOF endpoint quality prediction model

As mentioned above, BOF is a complex multi-phase batch chemical reactor with more than a dozen of input variables to produce the desired steel grade. In this study, an EO-LM algorithm is developed and applied in MLP network training for BOF endpoint quality prediction with the abilities of avoiding local minimum and performing detailed local search. Based on the mechanism of the BOF process, a feed-forward MLP network {13, 10, 2} is adopted for real applications, which consists of one hidden layer with ten hidden nodes. As shown in Fig. 1, the inputs are 13 variables including uncontrollable hot metal information (hot metal temperature, weight and chemical compositions, etc.) and the operational receipt variables (oxygen, scrap and a variety of recipes, etc.). Two outputs are defined as the endpoint temperature and the carbon content, respectively. The EO-LM learning is executed in two parallel phases: the genotype for EO-LM and the phenotype for MLP network. The synaptic weights and biases are encoded as a real-valued chromosome, to be evolved during EO-LM iterations.

3.4 Fitness function

The fitness function measures how fit an individual (solution) is, and the "fittest" one has a greater chance to be inherited into the next generation. A "global fitness" must be defined to evaluate a solution. In this study, the global fitness is defined as the sum of the root mean square error (RMSE) on a training set (LRMSE) and a validation set (GRMSE) to prevent "over-fitting",

$$\begin{aligned}&\mathrm{Fitness}_{\mathrm{global}}(S)\\&=\mathrm{LRMSE}_{S(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{r})}+\mathrm{GRMSE}_{S(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{r})}\\&=\sqrt{\frac{\sum_{i=1}^{n}\sum_{l=1}^{n_{\_\mathrm{Train}}}(y_i^l-\hat{y}_i^l)^2}{n\times n_{\_\mathrm{Train}}}}+\sqrt{\frac{\sum_{i=1}^{n}\sum_{l=1}^{n_{\_\mathrm{Valid}}}(y_i^l-\hat{y}_i^l)^2}{n\times n_{\_\mathrm{Valid}}}}.\end{aligned} \tag{6}$$

Unlike general algorithm, which works with a population based solutions, EO only stands on a single individual (i.e., chromosome) based evolution. This requires a suitable representation which permits each component to be assigned with a quality measure (i.e., fitness) called "local fitness". In this study, the

local fitness λ_k is defined as an improvement in LRMSE made by the mutation imposed on the kth component of the best-so-far chromosome S:

$$\begin{aligned}\lambda_k &= \text{Fitness}_{\text{local}}(k) = \Delta\text{LRMSE}(k)\\ &= \text{LRMSE}_{S(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{r})} - \text{LRMSE}_{S'_k(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{r})}.\end{aligned} \tag{7}$$

As long as the NN is not over-fitting, the improvement on the local fitness λ_k will also improve the global fitness $\text{Fitness}_{\text{global}}$ described in Eq. (6).

3.5 Workflow and algorithm

This section will illustrate the workflow of the proposed EO-LM algorithm and introduce three mutation (global/local search) operators adopted in this study: standard EO mutation, LM mutation, and multi-start Gaussian mutation. All of these operators are available in the evolution process and, at each decision point, make a choice of which to apply based on a probability parameter p_m. There are two evolutionary levels during the optimization. On one hand, evolution takes place at the "chromosome level" as in any other evolutionary algorithm; chromosomes (genes) represent solutions and features of the problem that one is trying to solve. On the other hand, evolution also happens at the "meme level", that is, the behaviors that individuals will use to alter the survival value of their chromosomes (Krasnogor and Gustafson, 2004). Accordingly, the solutions are evaluated by fitness functions of two different levels: the fitness of the respective gene itself (global fitness in Eq. (6)), and the interaction fitness between the respective gene and the respective meme (local fitness in Eq. (7)). Thus, both genetic and meme materials are co-evolved, the evolutionary changes at the gene level are expected to influence the evolution at the meme level, and vice versa. The proposed EO-LM is able to self-assemble different mutation operators and co-evolve the behaviors to successfully solve the NN supervised learning problem. The flowchart of the EO-LM algorithm is shown in Fig. 2.

The work steps of the proposed algorithm can be described as below:

Step 1. Define the number of neurons for the hidden layer, input layer, and output layer and set the control parameters used in the EO-LM algorithm.

Step 2. Initialize the NN weights and biases randomly.

Step 3. Map the weights/biases of the NN from the problem oriented phenotype space to a chromosome, as shown in Fig. 1.

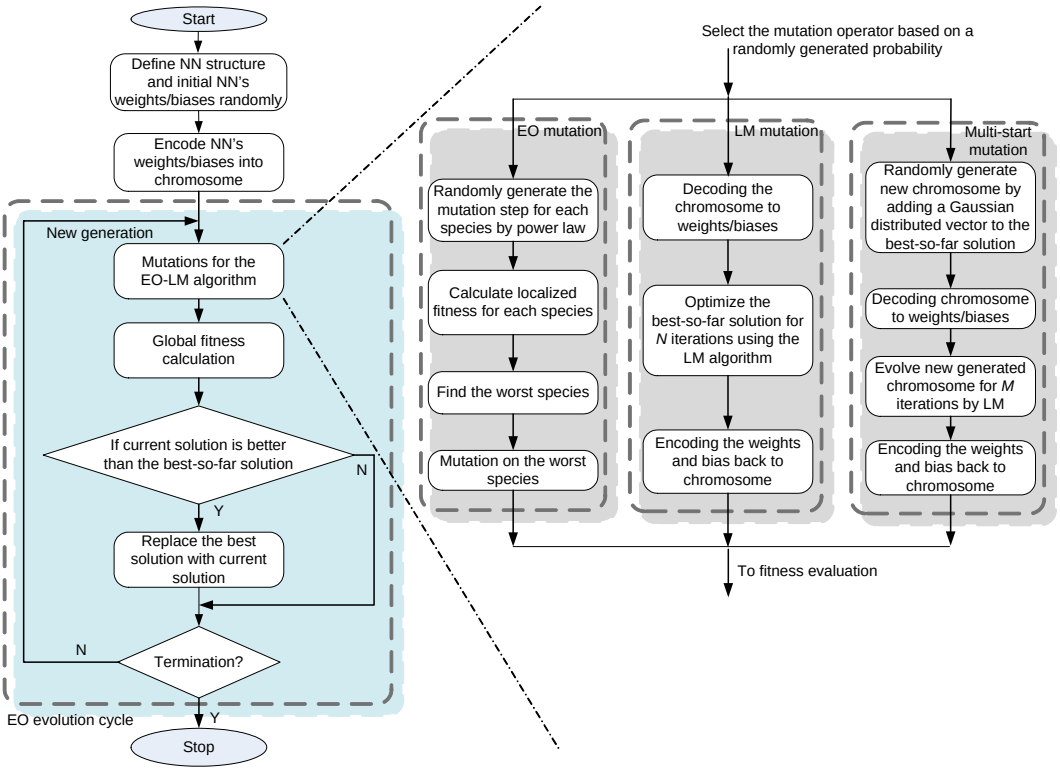

Fig. 2 Flowchart of the EO-LM based NN learning

Step 4. For the first iteration of EO, decode the initial chromosome S to weights/biases and calculate the object fitness function, set $S_{\text{best}}=S$.

Step 5. Decide what kind of mutation operator should be imposed to the current chromosome S based on randomly generated probability parameters P_m. If $P_m \le P_{m\_\text{basic}}$, go to (a); if $P_{m\_\text{basic}} < P_m \le P_{m\_\text{LM}}$, go to (b); else if $P_m > P_{m\_\text{LM}}$, go to (c).

(a) Perform the standard EO mutation on the best-so-far solution S in the following orders.

(i) Change the value of each component of S and get a set of new solutions S'_k, $k \in [1, 2, \ldots, n]$.

(ii) Sequentially evaluate the localized fitness λ_k specified in Eq. (7) for each S'_k, and rank them according to their fitness values.

(iii) Choose the best solution S' from the new solutions set [S'].

(b) Perform the LM mutation on the best-so-far chromosome S as below.

(i) Decode the chromosome S to weights/biases matrices in MLP networks.

(ii) The weights vector is updated for N iterations by

$$S' = S - [\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + \mu I]^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{e}, \tag{8}$$

where $\boldsymbol{e}$ is a vector of network errors defined in Eq. (9),

$$\boldsymbol{e}(x) = \sum_{i=1}^{n}\sum_{l=1}^{n_{\_\text{Train}}}(y_i^l - \hat{y}_i^l)^2. \tag{9}$$

(iii) Encode the updated weights/biases matrices to the chromosome S'.

(c) Perform the multi-start Gaussian mutation on the best-so-far chromosome S. Gaussian mutation has its main objective to increase diversity, whereby a larger part of the search space is explored. This strategy is often adopted to explore the neighborhood of the current solution in the following steps.

(i) Generate a new chromosome S'_0 by adding a Gaussian distribution random vector with n dimension to the best-so-far chromosome S.

$$S'_0 = S + \text{Scale} \times N(0,1), \tag{10}$$

where n is the length of chromosome, Scale is the mutation step size.

(ii) Decode the chromosome S'_0 to weights/biases matrices in MLP networks.

(iii) Train the MLP network based on LM algorithm in Eq. (8) for M iterations.

(iv) Encode the updated weights/biases matrices to the chromosome S'.

Step 6. Decode the chromosome S' to weights/biases matrices and calculate the global object fitness function. If $F(S') < F(S_{\text{best}})$, set $S_{\text{best}}=S'$.

Step 7. If the termination criteria are not satisfied, go to step 5, else go next.

Step 8. Return S_{best}.

4 Application to endpoint temperature prediction model of a production scale basic oxygen furnace

In this section, the proposed hybrid EO-LM algorithm is applied to a practical engineering problem for a production scale BOF in steelmaking. To evaluate the effectiveness of the proposed hybrid EO-LM for the predictions of endpoint temperature and carbon content, the simulation experiment is performed using real industry data. Over 1600 pairs of data are gathered from the steel plant database, among them, 800 pairs are selected randomly as training data, 480 pairs as validation data and the rest 320 pairs as test data. To evaluate the performance of the proposed EO-LM based NN model, the conventional LM algorithm is applied with the same test dataset as used in the EO-LM model.

The simulation performance of the EO-LM model is evaluated in terms of RMSE, mean error (ME) and correlation coefficient. Table 1 gives the RMSE and ME values for the two different models of the testing phase. Compared with the conventional LM algorithm, the proposed algorithm reduces the prediction RMSE by 8.65% and 14% for endpoint temperature and carbon content, respectively.

The scatter diagram in Fig. 3 shows the extent of the match between the measured and predicted values by EO-LM and LM learning algorithms. It can be seen that the EO-LM model shows a better agreement with the target than those by the conventional LM learning algorithm.

The comparison of prediction error distributions for endpoint temperature and carbon content between the hybrid EO-LM algorithm and the conventional

Table 1 Comparison between the EO-LM and conventional LM on BOF endpoint prediction model

Performance	Endpoint temperature (°C)		Endpoint carbon content (%)	
	RMSE	ME	RMSE	ME
EO-LM	17.5232	−1.1372	0.0129	−7.7257E−004
Conventional LM	19.1821	−1.7186	0.0150	−0.0030
Improvement	8.65%	33.83%	14%	74%

LM algorithm are shown in Fig. 4. It can also be seen that the range of prediction residuals are reduced by the hybrid EO-LM algorithm compared to the conventional LM algorithm.

The experimental results indicate that the proposed EO-LM can easily avoid the local minima, over-fitting and under-fitting problems suffered by traditional gradient search based training algorithms, and provide better prediction results.

5 Conclusions

The BOF steelmaking is a highly complex process and difficult to model and control. In this study, a novel hybrid NN training method with the integration of EO and LM is presented for BOF

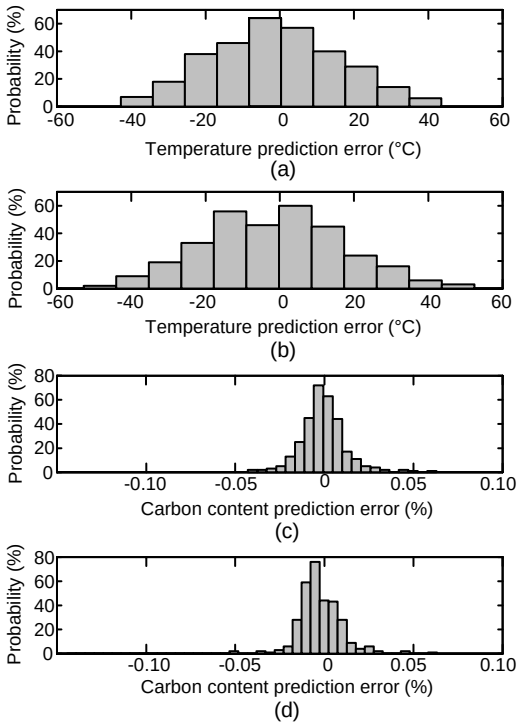

Fig. 4 Prediction error distribution comparison between the EO-LM and conventional LM.
Temperature prediction error by the (a) EO-LM and (b) conventional LM; carbon content prediction error by the (c) EO-LM and (d) conventional LM

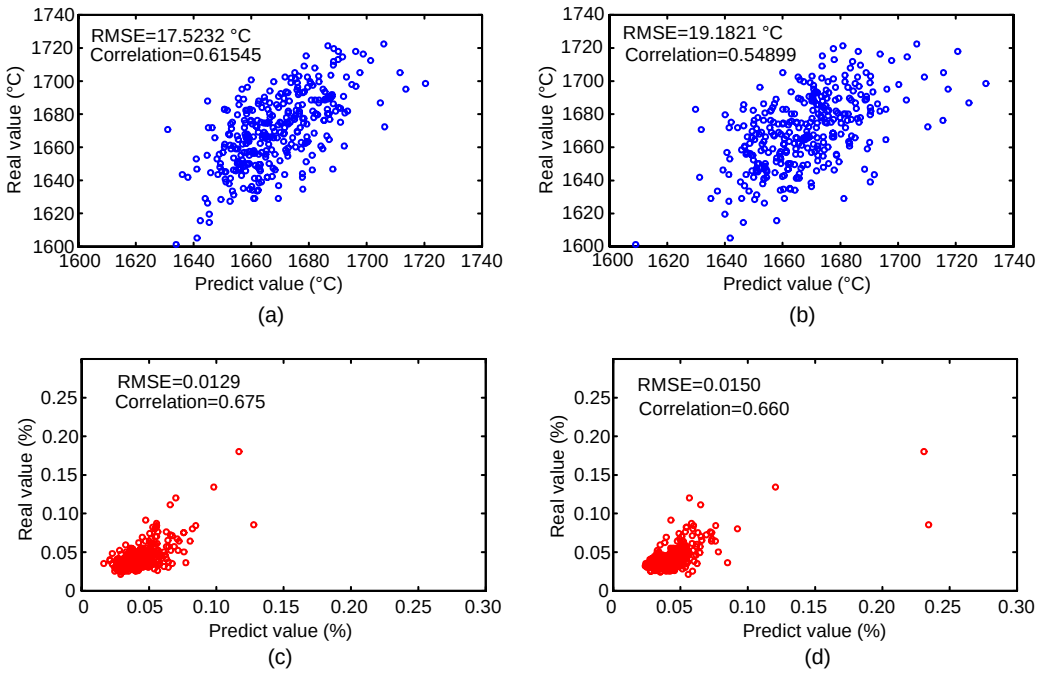

Fig. 3 Prediction accuracy comparison between the EO-LM and conventional LM
Endpoint temperature prediction by the (a) EO-LM and (b) conventional LM; endpoint carbon content prediction by the (c) EO-LM and (d) conventional LM

endpoint quality prediction. The main advantage of the proposed EO-LM algorithm is to use the superior features of EO and LM in global and local search, respectively. As a result, the application of the proposed EO-LM algorithm in NN learning may create a BOF endpoint prediction model with better performance, such as avoiding local minima and having good generalization capability.

Future studies will involve applications of EO-LM in optimizing the architecture of NN and applications to additional real-world problems.

References

Bak, P., Tang, C., Wiesenfeld, K., 1987. Self-organized criticality: An explanation of the 1/*f* noise. *Physical Review Letters*, **59**(4):381-384. [doi:10.1103/PhysRevLett.59.381]

Boettcher, S., Percus, A.G., 1999. Extremal Optimization: Methods Derived from Co-evolution. Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco, p.825-832.

Boettcher, S., Percus, A.G., 2000. Nature's way of optimizing. *Artificial Intelligence*, **119**(1-2):275-286. [doi:10.1016/S0004-3702(00)00007-2]

Chen, M.R., Lu, Y.Z., 2008. A novel elitist multiobjective optimization algorithm: Multiobjective extremal optimization. *European Journal of Operational Research*, **188**(3):637-651. [doi:10.1016/j.ejor.2007.05.008]

Chen, M.R., Lu, Y.Z., Yang, G.K., 2007. Multiobjective extremal optimization with applications to engineering design. *Journal of Zhejiang University-SCIENCE A*, **8**(12):1905-1911. [doi:10.1631/jzus.2007.A1905]

Chen, Y.W., Lu, Y.Z., Chen, P., 2007. Optimization with extremal dynamics for the traveling salesman problem. *Physica A: Statistical Mechanics and Its Applications*, **385**(1):115-123. [doi:10.1016/j.physa.2007.06.014]

Cotta, C., Moscato, P., 2007. Memetic Algorithms. Handbook of Approximation Algorithms and Metaheuristics, CRC Press, New York.

Cox, I.J., Lewis, R.W., Ransing, R.S., Laszczewski, H., Berni, G., 2002. Application of neural computing in basic oxygen steelmaking. *Journal of Materials Processing Technology*, **120**(1-3):310-315. [doi:10.1016/S0924-0136(01)01136-0]

Dawkins, R., 1976. The Selfish Gene. Oxford University Press, Oxford.

Dengiz, B., Alabas-Uslu, C., Dengiz, O., 2009. A tabu search algorithm for the training of neural networks. *Journal of the Operational Research Society*, **60**(2):282-291. [doi:10.1057/palgrave.jors.2602535]

Digalakis, J., Margaritis, K., 2004. Performance comparison of memetic algorithms. *Applied Mathematics and Computation*, **158**(1):237-252. [doi:10.1016/j.amc.2003.08.115]

Feng, M.X., Li, Q., Zou, Z.S., 2008. An outlier identification and judgment method for an improved neural-network BOF forecasting model. *Steel Research International*, **79**(5):323-332. [doi:10.2374/SRI07SP051-2008-323]

Fileti, A.M.F., Pacianotto, T.A., Cunha, A.P., 2006. Neural modelling helps the BOS process to achieve aimed end-point conditions in liquid steel. *Engineering Applications of Artificial Intelligence*, **19**(1):9-17. [doi:10.1016/j.engappai.2005.06.002]

Hagan, M.T., Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, **5**(6):989-993. [doi:10.1109/72.329697]

Haykin, S., 1994. Neural Networks, a Comprehensive Foundation. Macmillan, New York.

Krasnogor, N., Gustafson, S., 2004. A study on the use of "self-generation" in memetic algorithms. *Natural Computing*, **3**(1):53-76. [doi:10.1023/B:NACO.0000023419.83147.67]

Lu, Y.Z., Chen, M.R., Chen, Y.W., 2007. Studies on Extremal Optimization and Its Applications in Solving Real World Optimization Problems. Proceedings of IEEE Symposium on Foundations of Computational Intelligence, Hawaii, USA, p.162-168. [doi:10.1109/FOCI.2007.372163]

Nocedal, J., Stephen, J.W., 2006. Numerical Optimization. Springer, New York.

Ozcan, E., 2007. Memes, self-generation and nurse rostering. *LNCS*, **3867**:85-104. [doi:10.1007/978-3-540-77345-0_6]

Pendharkar, P.C., 2009. Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services. *Expert Systems with Applications*, **36**(3):6714-6720. [doi:10.1016/j.eswa.2008.08.050]

Sedki, A., Ouazar, D., El Mazoudi, E., 2009. Evolving neural network using real coded genetic algorithm for daily rainfall-runoff forecasting. *Expert Systems with Applications*, **36**(3):4523-4527. [doi:10.1016/j.eswa.2008.05.024]

Smith, J.E., 2007. Coevolving memetic algorithms: a review and progress report. *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, **37**(1):6-17. [doi:10.1109/TSMCB.2006.883273]

Tao, J., Wang, X., Chai, T.Y., 2002. Intelligent Control Method and Application for BOF Steelmaking. Proceedings of the IFAC World Congress, Barcelona, Spain, p.1071-1076. [doi:10.3182/20020721-6-ES-1901.00726]

Tavakkoli-Moghaddam, R., Saremi, A.R., Ziaee, M.S., 2006. A memetic algorithm for a vehicle routing problem with backhauls. *Applied Mathematics and Computation*, **181**(2):1049-1060. [doi:10.1016/j.amc.2006.01.059]

Wang, X., Wang, Z.J., Tao, J., 2006. Multiple neural network modeling method for carbon and temperature estimation in basic oxygen furnace. *LNCS*, **3973**:870-875. [doi:10.1007/11760191_127]

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, **1**(1):67-82. [doi:10.1109/4235.585893]

Yao, X., Islam, M.M., 2008. Evolving artificial neural network ensembles. *IEEE Computational Intelligence Magazine*, **3**(1):31-42. [doi:10.1109/MCI.2007.913386]

Zhu, Z., Ong, Y.S., Zurada, J.M., 2009. Identification of full and partial class relevant genes. *IEEE/ACM Transaction on Computational Biology and Bioinformatics*, **7**(2):263-277. [doi:10.1109/TCBB.2008.105]