Knowledge-Based Systems xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency

Guoqiang Li [a,*], Peifeng Niu [a,b], Yunpeng Ma [a], Weiping Zhang [a]

[a] Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao CO 066004, China
[b] National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Qinhuangdao, China

ARTICLE INFO

Article history:
Received 16 January 2014
Received in revised form 26 March 2014
Accepted 25 April 2014
Available online xxxx

Keywords:
Artificial bee colony
Extreme learning machine
Greedy selection mechanism
Opposition-based learning
Coal-fired boilers

ABSTRACT

In this paper, a novel optimization technique based on artificial bee colony algorithm (ABC), which is called as PS-ABCII, is presented. In PS-ABCII, there are three major differences from other ABC-based techniques: (1) the opposition-based learning is applied to the population initialization; (2) the greedy selection mechanism is not adopted; (3) the mode that employed bees become scouts is modified. In order to illustrate the superiority of the proposed modified technique over other ABC-based techniques, ten classical benchmark functions are employed to test. In addition, a hybrid model called PS-ABCII-ELM is also proposed in this paper, which is combined of the PS-ABCII and Extreme Learning Machine (ELM). In PS-ABCII-ELM, the PS-ABCII is applied to tune input weights and biases of ELM in order to improve the generalization performance of ELM. And then it is applied to model and optimize the thermal efficiency of a 300 MW coal-fired boiler. The experimental results show that the proposed model is very convenient, direct and accurate, and it can give a general and suitable way to predict and improve the boiler efficiency of a coal-fired boiler under various operating conditions.

1. Introduction

The artificial bee colony algorithm (ABC) [1] proposed by Karaboga has been become available and promising techniques for real-world optimization problems. Due to the simple concept, easy implementation and quick convergence, ABC has attracted much attention and wide applications in various fields [2–11]. In ABC, the exploration process refers to the ability of seeking for the global optimum in the solution space of various unknown optimization problems, while the exploitation process refers to the ability of applying the knowledge of previous solutions to look for better solutions. However, there are still some insufficiencies, namely, ABC is good at exploration but poor at exploitation and its convergence speed is also an issue in some cases, and sometimes traps into local optimum solutions. For these insufficiencies, a few modified or improved algorithms based on ABC are presented in recent years, such as best-so-far ABC [12], GABC [13], BSO [14], IABC and PS-ABC [15]. These improved ABCs have better performances than the original ABC, especially that the PS-ABC could achieve very good search results in need of less iterations, however, it needs more time to calculate and compare the fitness of candidate solutions.

Recently, Extreme Learning Machine (ELM) proposed in the literature [16] is a kind of single hidden layer feedforward neural network (SLFN). Compared with BP, RBF and SVM, the ELM has a much faster learning algorithm and better reliability and generalization capability. In addition, the ELM overcomes many difficulties encountered by gradient-based learning methods, such as stopping criteria, learning rate, number of epochs and local minima. Due to these advantages, ELM has been successfully applied to various domains, such as classification [17], online learning [18], function approximation [19,20], Sales forecasting [21], nontechnical loss analysis [22], terrain reconstruction [23] and protein structure prediction [24]. An ELM is usually formed by an input layer, a hidden layer and an output neuron. Each neuron in the hidden and output layers is defined as a traditional neuron node, with a given activation function defined by the user. In ELM, the input weights and bias of hidden neurons are randomly generated, and the output weights are analytically calculated by means of the Moore–Penrose generalized inverse. The minimum norm least squares solution of a general linear system plays an important role in fastening the SLFN learning process. Although ELM is fast and presents good generalization performance, there may exist some non-optimal or unnecessary input weights and biases. Also, ELM may require more

* Corresponding author. Tel.: +86 1393 3628751; fax: +86 0335 8072979.
E-mail addresses: zhihuiyuang@163.com (G. Li), niupeifeng2011@163.com (P. Niu), 1037026022@qq.com (Y. Ma), zhihuiyguang@sina.com (W. Zhang).

http://dx.doi.org/10.1016/j.knosys.2014.04.042
0950-7051/

hidden neurons than conventional tuning-based learning algorithms in some applications, which may make ELM respond slowly to unknown testing data. For these problems, many modified or improved ELMs have been proposed in literatures. In order to further enhance the performance of ELM, there are four types improved ELM models: ensemble type, optimization type, incremental type, replacement type. In the ensemble type, different ELMs are trained by disjoint subsets of data, but share the same hidden layer neurons [25–28]. In the optimization type, various optimization methods are employed to adjust input weights and hidden layer bias of ELM and optimize the network structure [29,30]. In the incremental type, the improved ELMs create new hidden layer neurons one by one according to certain criteria.[31,32]. In the replacement type, modified ELMs substitute the activation functions of ELM (sigmoid and RBF) for sine and cosine functions (or other functions), which is very helpful to improve the accuracy and the convergence rate for the problem of function approximation [33].

In this paper, for this shortage of PS-ABC, an improved PS-ABC algorithm (PS-ABCII) is proposed to achieve faster convergence speed and better solution accuracy with a minimum incremental computational burden which is more proper for real-world optimization problems. In PS-ABCII, there are three differences from the PS-ABC: (1) the greedy selection mechanism is adopted in PS-ABC but not in PS-ABCII. Namely, a new storage unit and a sort algorithm are introduced into PS-ABCII to replace the greedy selection mechanism to search candidate solutions in order to fasten the convergence speed and the running time of the PS-ABCII; (2) the initial population is generated based on the opposition-based learning method to further enhance the global convergence; (3) the mode that employed bees become scouts is modified in order to further keep the diversity of the population. Compared with ABC and PS-ABC, PS-ABCII needs much less convergence time and running time, and its search performance is more outstanding than the other two algorithms.

In addition, a novel PS-ABCII-ELM model, which adopts the PS-ABCII algorithm to tune input weights and bias of ELM, is proposed. In order to illuminate the feasibility of the proposed model, it is applied to identify the thermal efficiency of a 300 WM coal-fired boiler. Experimental results show that the PS-ABCII-ELM model can achieve good prediction effects under various operating conditions, and then it could be adopted to predict the boiler efficiency of a coal-fired boiler under various operating conditions.

The rest of this paper is arranged as follows. In Section 2, some basic concepts and related works are reviewed. The PS-ABCII is proposed in Section 3. Experimental study shows the PS-ABCII validity in Section 4. Section 5 presents PS-ABCII-ELM model and applies it to model and predict the boiler efficiency. Finally, Section 6 concludes this paper.

2. Basic concepts and related works

2.1. Extreme learning machine

Extreme Learning Machine (ELM) proposed by Huang et al. [16]is a novel single hidden layer feedforward neural network (SFLN) learning algorithm, whose structure is given in Fig. 1. In ELM, one key principle is that the input weight values and hidden layer biases are randomly assigned. After that, the SLFN would become a linear system and its output weights could be analytically calculated by the least square method. Here, we briefly describe the learning algorithm of ELM.

Given a training data with N training samples $(\mathbf{x}_i, \mathbf{t}_i)$ where $\mathbf{x}_i = [x_1^i, x_2^i, \ldots, x_n^i]$ is an n-dimensional feature vector of the ith sample and $\mathbf{t}_i = [t_1^i, t_2^i, \ldots, t_L^i]$ is the target vector. Here, the matrix

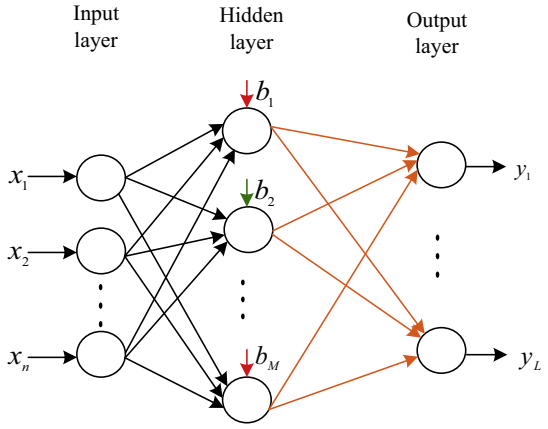

Fig. 1. Structure of extreme learning machine.

$\mathbf{W}$ and $\mathbf{B}$ are randomly assigned. Then the output ($\mathbf{T}$) of the ELM with M hidden neurons could be calculated by the following form:

$$\mathbf{t}_k^i = \sum_{j=1}^{M} \beta_{kj} g_j(\mathbf{W}, \mathbf{B}, \mathbf{X}), \quad k = 1, 2, \ldots, L \tag{1}$$

where $g_j(\cdot)$ is the activation function, $\mathbf{W}$ is $M \times n$ input weight matrix, $\mathbf{B}$ is $M \times 1$ hidden layer bias matrix and $\boldsymbol{\beta}$ is $L \times M$ output weight matrix.

The Eq. (1) could be rewritten in matrix form as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2}$$

where $\mathbf{H}$ is the hidden layer output matrix and defined as:

$$\mathbf{H}(\mathbf{W}, \mathbf{B}, \mathbf{X}) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_M \cdot \mathbf{x}_1 + b_M) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_M \cdot \mathbf{x}_N + b_M) \end{bmatrix}_{N \times M} \tag{3}$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1 \quad \cdots \quad \boldsymbol{\beta}_M]_{L \times M}^T \tag{4}$$

and

$$\mathbf{T} = [\mathbf{t}_1 \quad \cdots \quad \mathbf{t}_N]_{L \times N}^T. \tag{5}$$

The output weight matrix $\boldsymbol{\beta}$ could be estimated analytically by the minimum norm least-square solution:

$$\tilde{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| = \mathbf{H}^{+}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{+}$ is the Moore–Penrose generalized inverse of $\mathbf{H}$. If the $\mathbf{H}$ is nonsingular, the Eq. (6) could be rewritten as

$$\tilde{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{T} \tag{7}$$

The learning algorithm of ELM can be summarized as the following three steps:

1) Randomly assign the input weight values $\mathbf{W}$ and hidden layer biases $\mathbf{B}$.
2) Calculate the output matrix $\mathbf{H}$ of the hidden layer.
3) Determine the output weight matrix $\boldsymbol{\beta}$.

2.2. Opposition-based learning

Opposition-based learning (OBL), which is a new concept in computational intelligence, was first proposed by Tizhoosh in 2005. OBL has been proved to be an effective method to enhance various optimization techniques [34,35].

When searching a solution x to a given problem, we usually make an estimate x', which is not an exact solution but based on experience or on a totally random guess. The random guess, if near the optimal solution, may result in fast convergence. If the random guess is far from the existing solution, it is in the opposite location, so approximation, search or optimization requires considerably more time, or may not be attained. The lack of a priori knowledge may counteract the best initial guess. Logically, we should search in all directions simultaneously, or, more concretely, in the opposite direction. Searching in the opposite direction could provide more chance for finding a candidate solution which is closer to the global optimum. If we are searching for a solution x and we agree that searching in the opposite direction could be advantageous. The first step is becoming calculating opposite number x'.

Definition 1. [34]: Let $x \in R$ be a real number defined on a certain interval: $x \in [a,b]$, then the opposite number x' is defined as follows:

$$x' = a + b - x \tag{8}$$

The above definition can be extended to higher dimensions as follows:

Definition 2. [36]: Let $P(x_1,x_2,\ldots,x_n)$ be a point in n-dimensional coordinates with $x_1,x_2,\ldots,x_n \in R$ and $x_i \in [a_i,b_i] \forall i \in \{1,2,\ldots,n\}$. Opposite point P' is completely defined by its coordinates $x'_1,x'_2,\ldots,x'_n$ where $x'_i = a_i + b_i - x_i$.

By applying Definition 2, the opposition-based optimization can be defined as follows:

Opposition-based optimization [34]: Let $X = (x_1,x_2,\ldots,x_D)$ be a point in a D-dimensional space (i.e. a candidate solution). Assume $f(X)$ is a fitness function which is used to evaluate the candidate's fitness. According to the definition of the opposite point, $X' = (x'_1,x'_2,\ldots,x'_D)$ is the opposite of $X = (x_1,x_2,\ldots,x_D)$. If $f(X')$ is better than $f(X)$, then update X with X'; otherwise keep the current point X. Hence, the current point and its opposite point are evaluated simultaneously in order to continue with the fitter one.

2.3. PS-ABC algorithm

The PS-ABC algorithm is a modified high-efficiency hybrid ABC algorithm. The PS-ABC algorithm has already proved to be a much more effective technique for solving global optimum solutions than other modified optimization methods. The PS-ABC not only has extremely fast convergence iteration, but also owns better search ability. The PS-ABC is briefly described as follows:

In PS-ABC, all bees are divided into three groups: employed bees, onlookers and scouts. Half of the colony consists of the employed bees, and the other half consists of the onlookers. The number of employed bees equals to the number of food sources. When a food source has been abandoned by bees, the abandoned employed bee would become a scout. The position of a food source corresponds to a possible solution to the optimization problem, and the nectar amount of each food source represents their quality (fitness) of the associated solution. Firstly, the PS-ABC would randomly generate an initial population $P(C=0)$ of SN food source positions, where SN is the size of food sources. Every solution $x_i (i = 1,2,\ldots,SN)$ is a D-dimensional vector, where D denotes the dimension of optimization parameters. After the initialization, the population of solutions is subject to repeated cycles $C = 1,2,\ldots,MCN$ of the search courses of employed bees, onlookers and scouts. In PS-ABC, each employed bee firstly works out three new solutions by three different solution search equations, and then chooses and determines the best one as the candidate solution. The operation process can be modified as the following form:

$$v^1_{ij} = x_{ij} w_{ij} + 2\left(\phi^1_{ij} - 0.5\right)(x_{ij} - x_{kj})\Phi_1 + \varphi^1_{ij}(y_j - x_{ij})\Phi_2 \tag{9}$$

$$v^2_{ij} = x_{ij} + 2\left(\phi^2_{ij} - 0.5\right)(x_{ij} - x_{kj}) + \varphi^2_{ij}(y_j - x_{kj}) \tag{10}$$

$$v^3_{ij} = x_{ij} + \phi^3_{ij}(x_{ij} - x_{kj}) \tag{11}$$

$$v_{ij} = \begin{cases} v^1_{ij} & Fitness(v^1_i) \geqslant \max\{Fitness(v^2_i), Fitness(v^3_i)\} \\ v^2_{ij} & else\ if\ Fitness(v^2_i) \geqslant Fitness(v^3_i) \\ v^3_{ij} & others \end{cases} \tag{12}$$

where $i = 1,2,\ldots,SN$; $j = 1,2,\ldots,D$, x_{ij} (or v_{ij}) denotes the jth element of the ith food source x_i (or candidate food source v_i). w_{ij} is the inertia weight which controls impacts of the previous solution x_i. y_j is the jth parameter of the best-so-far solution; ϕ^1_{ij}, ϕ^2_{ij}, ϕ^3_{ij}, φ^1_{ij} and φ^2_{ij} are random numbers between $[0,1]$, Φ_1 and Φ_2 is a positive parameter that could control the maximum step size. The inertia weight and acceleration coefficients are defined as functions of the fitness in the search process of PS-ABC.

$$w_{ij} = \Phi_1 = 1/(1 + \exp(-Fitness(i)/ap)^{iter}) \tag{13}$$

$$\Phi_2 = \begin{cases} 1 & if\ a\ bee\ is\ employed\ one \\ 1/(1 + \exp(-Fitness(i)/ap)^{iter}) & if\ a\ bee\ is\ onlooked\ one \end{cases} \tag{14}$$

where ap is the $Fitness(1)$ in the first iteration and $iter$ is the current iteration number.

Due to obtaining the candidate solution before the employed bee decide where they should go to explore, the process of calculating new food positions is called 'predict'. After bees 'predict' new candidate solutions by three different solution search equations (Eqs. (9)–(11)), they select the best one from the three solutions as the candidate solution by Eq. (12).

If the fitness value of the candidate solution is better than the best fitness value achieved so far, then the employed bee moves to this new food source and synchronously abandons the old one, otherwise it remains the previous food source in its mind. When all employed bees have finished this process, they share the fitness information with the onlookers, each of which selects a food source according to the probability given in Eq. (15).

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \tag{15}$$

As in the case of the employed bee, each onlooker 'predicts' three modifications on the position in her memory, and then selects the best one as the candidate source and checks the fitness value of the candidate source. Providing that the fitness value of the candidate source is better than that of the previous one, the bee would memorize the new position and forget the old one.

Suppose, a source X_i could not be further improved by the predetermined number 'limit', the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout. The scout produces a food source by the following form:

$$x_{ij} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{16}$$

After each source position updated, it would be evaluated, and its fitness is compared with that of its previous one. If the fitness of the new food source equals or is better than that of the previous one, it would be replaced with the previous one in its memory. Otherwise, the previous one would be retained in its memory.

The main difference between the PS-ABC and other modified ABCs is how to determine the candidate solution process. Namely, the PS-ABC has the abilities of prediction and selection but other

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

modified ABCs do not. In PS-ABC algorithm, every employed bee or onlooked bee could predict their next solutions of three different solution search equations and select the best one as its candidate solution. Then the bee could decide where they should go to explore or to exploit.

3. The proposed PS-ABCII

Although the PS-ABC not only has extremely fast convergence iteration, but also owns better search ability, the PS-ABC need more time to calculate and compare the fitness of candidate solutions. So this paper proposes another improved ABC algorithm based on PS-ABC, which is called PS-ABCII. In PS-ABCII, there are three major differences from PS-ABC: (1) population initialization; (2) the technique that chooses candidate solutions; (3) the mode that employed bees become scouts. The descriptions of PS-ABCII is following in detail.

3.1. Population initialization

Population initialization is a crucial task in ABC, because it could affect the convergence speed and the quality of the final food sources. In lack of any priori information about the optimum food sources, the random initialization is the most commonly adopted method to generate candidate food sources. Considering the search direction and its opposite at the same time will bear more likelihood to reach the best population in a shorter time. The main idea behind opposition-based learning is the simultaneous consideration of an estimate and its corresponding opposite estimate for a food source in order to achieve a better approximation for the current candidate food source. A mathematical proof [36] was proposed to show that, in general, an opposite number is more likely to be closer to the optimal solution than a purely random one. So replacing the random initialization with the opposition-based population initialization can get better initial food sources and then accelerate the convergence speed of the algorithm. Based on opposition-based learning, we propose the following algorithm to generate an initial population. Following steps show that procedure: (*SN* is the size of food sources, namely the number of employed bees; *D* denotes the dimension of a food source).

(1) Initialize the population of solutions $X = \{x_{ij}\}$, $i = 1, 2, \ldots, SN, j = 1, 2, \ldots, D$.
(2) Calculate opposite population $X' = \left\{x'_{ij}\right\}$ by

$$x'_{ij} = x_{\min,j} + x_{\max,j} - x_{ij} \quad i = 1, 2, \ldots, SN;\ j = 1, 2 \ldots, D \qquad (17)$$

where x_{ij} and x'_{ij} denote jth variable of the ith vector of the population and the opposite population, respectively.
(3) Select the *SN* fittest individuals from $\{X \cup X'\}$ as initial population.

3.2. The optimization process

In the PS-ABCII algorithm, the colony of artificial bees is still divided into three groups: employed bees, onlookers and scouts like ABC and PS-ABC. The first half of the colony consists of the employed bees, and the other half is the onlookers. In ABC and PS-ABC, when a food source has been abandoned by bees, the abandoned employed bee would become a scout. However, in PS-ABCII, a fixed proportion of employed bees would become scouts in order to keep the diversity of the population. In PS-ABCII, the position of a food source corresponds to a possible solution to the optimization problem, and the nectar amount of each food source represents their quality (fitness) of the associated solution. The number of the employed bees equals to the number of food sources.

In the first place, the PS-ABCII would initialize the population according to Section 3.1 and then the population of solutions is subject to repeated cycles $C = 1, 2, \ldots, MCN$ of the search processes of employed bees, onlookers and scouts. The PS-ABCII still keeps the original abilities of the exploration and the exploration because PS-ABC shows better search performance in the processes of exploration and the exploitation. Namely, the search forms (Eqs. (9)–(11)) of the PS-ABC are still as the modified forms of PS-ABCII.

In employed bee phase, an employed bee x_i generates three modifications (v_i^1, v_i^2 and v_i^3) in the neighborhood of its present position by using Eqs. (9)–(11). The major difference between the PS-ABC and PS-ABCII is that the greedy selection mechanism is adopted in PS-ABC but not in PS-ABCII. In PS-ABCII, the obtained three modifications for each employed bee would not be evaluated at once and not be compared with previous food sources but be kept in an archive. So, a new variable (Ψ) is introduced to memory previous food source and all modifications of all employed bees in a cycle. Namely, $\Psi = \{v^1, v^2, v^3, x\}$, where v^i and x respectively denote the set about all modifications by the ith modification form and the set about all previous food sources for employed bees. When all employed bees complete the process of exploration, the set $\Psi = \{v^1, v^2, v^3, x\}$ would be evaluated only once in some cycle for employed bees, and then the set $\Psi = \{v^1, v^2, v^3, x\}$ would be resorted in descending order of fitness, namely, $\beta = sort(\Psi)$, where β is the sorted set. The first *SN* food sources of the set β are chosen as the candidate food source for employed bees, namely $x = \beta(1 : SN)$.

In onlooker bee phase, all employed bees have completed the search process and would share the information about nectar amounts and positions of food sources with onlookers. An onlooker evaluates the nectar information which is owned by all employed bees, and then chooses a food source with a probability (Eq. (15)). As in the case of the employed bee, the onlooker can produce three modifications on the position in its memory, In PS-ABCII, when these modifications are obtained, the nectar amount of the new sources would not be checked at once but be kept in the set Ψ. That is to say, the greedy selection mechanism is not adopted by onlooker bees between previous and three new food sources. When all onlooker bees completed the exploitation process, the set $\Psi = \{v^1, v^2, v^3, x\}$ would be evaluated only once in some cycle for onlooker bees, and then the set $\Psi = \{v^1, v^2, v^3, x\}$ would be resorted in descending order of fitness. The first *SN* food sources of the set β are chosen as the candidate food source for onlooker bees, namely $x = \beta(1 : SN)$. Then all onlooker bees have completed the search process. Here, the first member of the candidate food source set is the best solution for optimization problems in the current cycle. If the found global best solution is not better than the first member, the global solution would be replaced by the first member of the candidate food source, or the global solution is still kept.

In scout bee phase, the difference between PS-ABC and PS-ABCII is that the mode that employed bees become scouts. In PS-ABC, a food source x_i could not be further improved by the predetermined number 'limit', the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout. However, in PS-ABCII, a fixed proportion P_t of employed bees would become scouts. Namely, the $P_t \times SN$ employed bees, which correspond to the $P_t \times SN$ worst food sources according to their fitness values, would become scouts in order to keep the population diversity. These scouts would find new food sources to replace with the previous ones by the Eq. (16).

Based on the above explanation of the initializing population and the optimization process, the flowchart of the PS-ABCII algorithm is shown in Fig. 2.

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

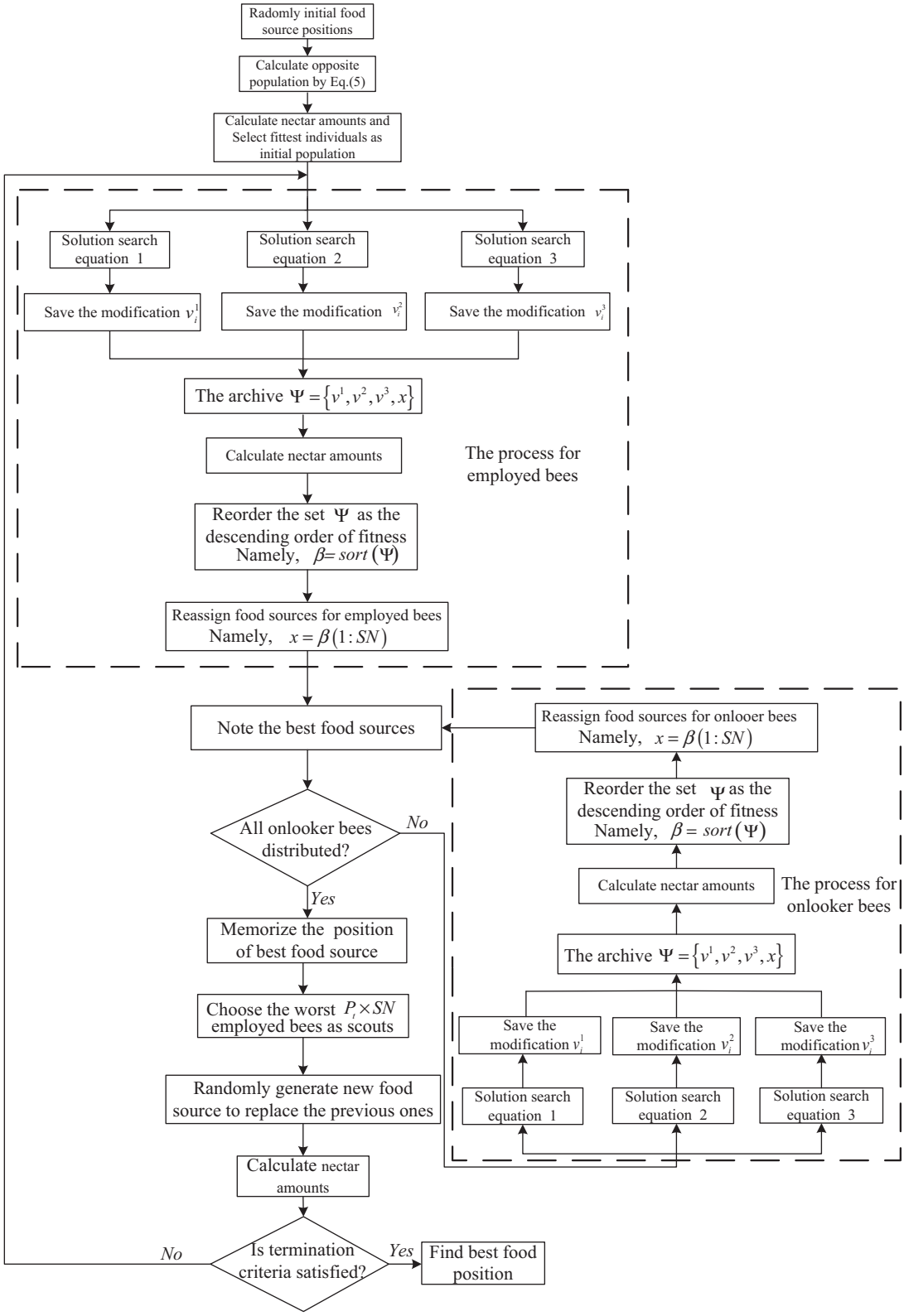

Fig. 2. Flowchart of the PS-ABCII algorithm.

4. Experimental study and discussion

The performance of the improved PS-ABCII algorithm was evaluated in 10 standard test function problems presented in Table 1. In Table 1, Functions $f_1(\vec{x})$ to $f_6(\vec{x})$ are unimodal high-dimensional functions. Functions $f_7(\vec{x})$ to $f_{10}(\vec{x})$ are multimodal high-dimensional ones. n is the dimension of classical functions, S denotes a subset of R^n. The global minimum values of 10 classical benchmark functions are zeros, except for the function $f_7(\vec{x})$ which has a minimum value of $-418.9829 \times n$, The optimum location for the functions presented in Table 1 are in $[0]^n$, except for $f_7(\vec{x})$ in $[420.96]^n$.

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

Table 1
Classical benchmark functions.

Test function	S
$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$
$f_2(\vec{x}) = \sum_{i=1}^{n} \mid x_i \mid + \prod_{i=1}^{n} \mid x_i \mid$	$[-10, 10]^n$
$f_3(\vec{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]^n$
$f_4(\vec{x}) = \max_i\{\mid x_i \mid, 1 \leqslant i \leqslant n\}$	$[-100, 100]^n$
$f_5(\vec{x}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	$[-30, 30]^n$
$f_6(\vec{x}) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	$[-1.28, 1.28]^n$
$f_7(\vec{x}) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\mid x_i \mid}\right)$	$[-500, 500]^n$
$f_8(\vec{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^n$
$f_9(\vec{x}) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$
$f_{10}(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$

In order to investigate the feasibility and effectiveness of the novel PS-ABCII in this work, a series of experiments are conducted, and further comparative experimental results with the standard ABC and PS-ABC are also given. In ABC and PS-ABC, the number of colony size is taken as 40, the maximum number of cycles is set as 1000 and 'limit' equals to 200. In PS-ABCII, the size of the colony is 40, the maximum number of cycles is 1000 and the proportion P_t that employed bees would become scouts is set as 40%. In addition, for every benchmark function, respectively assume the dimension as 30, 50 and 100.

Every experiment is repeated 30 times each starting from a random population with different random seeds. The mean of runtime and the convergence time, the mean and the standard deviation of the found best solution for every function are reported in Table 2–4. The notations dim, C.I., Mean and S.D. denote the dimension, the convergence iteration which is the iteration number firstly achieving the best solutions for the mean of best-so-far function values over 30 runs, the mean and the standard deviation of the found best function values, respectively. In addition, the notation $t_{C.I.}$ denotes the mean of convergence time which an algorithm spends when it achieves the convergence iteration. t_{run} is the runtime for an algorithm. All the experiments are carried out under windows 2000 and Matlab 7.1 with AMD 9650, 2.31 GHz CPU and 2G RAM.

As seen from Tables 2 and Figs. 3–8, it can be seen that the PS-ABCII could find the theoretical global optima on 5 functions ($f_1, f_2, f_4, f_8, f_{10}$) and extremely close to the theoretical optima on 4 functions (f_6, f_7, f_9). Compared to ABC, PS-ABCII could achieve much better solutions than ABC on 8 functions ($f_1, f_2, f_4, f_6, f_7, f_8, f_9, f_{10}$), meanwhile the convergence iteration is much smaller than ABC. While ABC could own better search performance than PS-ABCII on only 2 functions (f_3, f_5), except for f_5 under dim = 100. As seen from Table 3, for the 10 optimization problems, both the running time and the achieving convergence time are much shorter than ABC.

Compared with PS-ABC, PS-ABCII could find similar or better optimal solutions than the PS-ABC on 7 functions ($f_1, f_2, f_4, f_6, f_8, f_9, f_{10}$) especially for f_6, f_4 under dim = 50 and dim = 100, f_9 under dim = 100 and f_{10} under dim = 100. For all optimization problems presented in Table 1, PS-ABCII is much superior to PS-ABC in the convergence iteration, the running time and the achieving convergence time. Although PS-ABC could find better solutions than the PS-ABCII on only 3 functions (f_3, f_5, f_7) except for f_5 under dim = 100, the running time is much longer than PS-ABCII.

Table 2
Q7 The mean, the standard deviations and the convergence iteration of functions of Table 1.

F	Dim	ABC			PS-ABC			PS-ABCII		
		C.I.	Mean	SD	C.I.	Mean	SD	C.I.	Mean	SD
f_1	30	1000	3.3955×10^{-9}	4.5376×10^{-9}	316	**0**	0	30	**0**	0
	50	1000	1.1483×10^{-5}	1.6272×10^{-5}	774	**0**	0	56	**0**	0
	100	1000	4.9461×10^{-3}	1.1389×10^{-2}	979	8.3417×10^{-47}	4.5689×10^{-46}	114	**0**	0
f_2	30	1000	5.1029×10^{-6}	1.8417×10^{-6}	20	**0**	0	17	**0**	0
	50	1000	2.8511×10^{-3}	1.3944×10^{-3}	54	**0**	0	34	**0**	0
	100	1000	2.7814×10^{-1}	4.0035×10^{-1}	338	**0**	0	64	**0**	0
f_3	30	1000	1.2598×10^{4}	2.9192×10^{3}	1000	$\mathbf{7.2696 \times 10^{3}}$	1.4359×10^{3}	980	4.0756×10^{4}	8.1760×10^{3}
	50	1000	4.6422×10^{4}	6.9821×10^{3}	1000	$\mathbf{3.0638 \times 10^{3}}$	3.4739×10^{3}	1000	1.2539×10^{5}	2.1047×10^{4}
	100	1000	1.8854×10^{5}	2.1886×10^{4}	1000	$\mathbf{1.3544 \times 10^{5}}$	1.2851×10^{4}	1000	5.4823×10^{5}	1.0256×10^{5}
f_4	30	1000	2.4044×10^{1}	3.3935	890	**0**	0	54	**0**	0
	50	1000	5.6020×10^{1}	5.1905	1000	1.8782×10^{1}	5.7908	94	**0**	0
	100	1000	8.2376×10^{1}	3.0440	1000	7.2160×10^{1}	4.0371	382	**0**	0
f_5	30	1000	3.2873	3.4035	1000	**1.4048**	2.7168	1000	2.8408×10^{1}	0.1154
	50	1000	3.7224×10^{1}	3.6453×10^{1}	1000	$\mathbf{3.1451 \times 10^{1}}$	2.9224×10^{1}	1000	4.8504×10^{1}	1.3535×10^{-1}
	100	1000	3.3118×10^{2}	3.8309×10^{2}	1000	2.0376×10^{2}	6.7028×10^{1}	1000	$\mathbf{9.8590 \times 10^{1}}$	1.5702×10^{-1}
f_6	30	995	1.5788×10^{-1}	3.6701×10^{-1}	985	1.8545×10^{-2}	5.3198×10^{-3}	999	$\mathbf{5.5447 \times 10^{-4}}$	1.2352×10^{-3}
	50	1000	4.2726×10^{-1}	8.2393×10^{-2}	988	5.7802×10^{-2}	1.6469×10^{-2}	975	$\mathbf{5.4388 \times 10^{-4}}$	6.4470×10^{-4}
	100	1000	1.5950	3.2657×10^{-1}	1000	2.2021×10^{-1}	4.1119×10^{-2}	990	$\mathbf{1.6151 \times 10^{-3}}$	3.2646×10^{-3}
f_7	30	1000	−12185.9	1.4299×10^{2}	995	**−12549.7**	4.4891×10^{1}	997	−12088.9	1.8715×10^{2}
	50	1000	−19359.1	3.1097×10^{2}	1000	**−20893.4**	7.9224×10^{1}	1000	−19414.1	3.3738×10^{2}
	100	1000	−34413.8	5.0878×10^{2}	1000	**−39976.6**	3.3634×10^{2}	1000	−37405.7	5.5665×10^{2}
f_8	30	1000	4.0160×10^{-1}	6.2228×10^{-1}	98	**0**	0	26	**0**	0
	50	1000	8.1857	2.4195	178	**0**	0	40	**0**	0
	100	1000	8.5540×10^{1}	1.1018×10^{1}	468	**0**	0	80	**0**	0
f_9	30	1000	2.4076×10^{-5}	1.2439×10^{-5}	170	$\mathbf{8.8817 \times 10^{-16}}$	0	31	$\mathbf{8.8817 \times 10^{-16}}$	0
	50	1000	4.0637×10^{-2}	3.2467×10^{-2}	304	$\mathbf{8.8817 \times 10^{-16}}$	0	54	$\mathbf{8.8817 \times 10^{-16}}$	0
	100	1000	3.8186	3.6198×10^{-1}	980	2.3270×10^{-14}	1.2259×10^{-13}	84	$\mathbf{8.8817 \times 10^{-16}}$	0
f_{10}	30	1000	1.4335×10^{-3}	4.0152×10^{-3}	620	**0**	0	58	**0**	0
	50	999	9.9977×10^{-3}	1.1718×10^{-2}	862	**0**	0	66	**0**	0
	100	1000	1.4344×10^{-1}	1.3282×10^{-1}	942	1.6904×10^{-3}	6.4786×10^{-3}	206	**0**	0

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

Table 3
Related time of functions of Table 1.

F	Dim	ABC		PS-ABC		PS-ABCII	
		$t_{C.I.}$	t_{run}	$t_{C.I.}$	t_{run}	$t_{C.I.}$	t_{run}
f_1	30	1.9396	1.9396	1.4713	4.6422	0.0287	0.7885
	50	2.0239	2.0239	3.7969	4.9026	0.0541	0.8813
	100	2.0068	2.0068	4.9446	5.0547	0.1287	1.1177
f_2	30	2.1396	2.1396	0.1068	5.3182	0.0138	0.7510
	50	2.4912	2.4912	0.3005	5.3911	0.0416	1.0750
	100	2.3037	2.3037	1.9218	5.6572	0.0880	1.3651
f_3	30	7.3963	7.3963	20.9364	20.9364	1.5662	1.5995
	50	11.3235	11.3235	31.3536	31.3536	2.5338	2.5338
	100	20.2052	20.2052	58.7151	58.7151	4.5557	4.5557
f_4	30	1.8208	1.8208	4.0963	4.6089	0.0417	0.7052
	50	1.8443	1.8443	4.5718	4.5718	0.0964	0.9968
	100	1.9255	1.9255	4.8062	4.8062	0.5162	1.3573
f_5	30	2.2917	2.2917	5.3437	5.3437	0.9109	0.9109
	50	2.2781	2.2781	5.7327	5.7327	1.0484	1.0484
	100	2.3602	2.3602	6.0634	6.0634	1.4233	1.4233
f_6	30	3.0883	3.1065	7.6681	7.7947	1.1582	1.1592
	50	3.3044	3.3044	7.7724	7.8749	1.3801	1.4171
	100	3.3242	3.3242	8.4218	8.4218	1.7963	1.8171
f_7	30	2.1796	2.1796	5.1944	5.2257	1.2260	1.2328
	50	2.2806	2.2806	5.5852	5.5852	1.5587	1.5587
	100	2.5062	2.5062	6.4252	6.4252	2.3811	2.3811
f_8	30	2.0504	2.0504	0.4927	4.9602	0.0287	0.9849
	50	2.2432	2.2432	0.9445	5.2306	0.0568	1.2552
	100	2.4740	2.4740	2.6687	5.6452	0.1473	1.8114
f_9	30	2.4140	2.4140	0.9959	5.8254	0.0410	1.1437
	50	2.4302	2.4302	1.8452	6.0264	0.0785	1.4088
	100	2.6484	2.6484	6.2592	6.3915	0.1714	1.9401
f_{10}	30	3.3499	3.3499	5.2275	8.4337	0.0958	1.5161
	50	3.4802	3.4869	6.0724	8.7962	0.1245	1.8302
	100	3.7468	3.7468	8.8332	9.3827	0.5519	2.6818

As seen from Tables 2 and 3, although the PS-ABCII could find the theoretical global optima or be very close with the theoretical global optima on 6 functions $(f_1,f_2,f_4,f_8,f_9,f_{10})$ under the given three dimensions, it needs more and more convergence iterations, the runtime and the mean of convergence time with the growth of the dimension. In addition, on the other 4 functions (f_3,f_5,f_6,f_7), the found optimum solutions are more and more far from the theoretical global optima with the growth of the dimension, and simultaneously other performances (C.I., $t_{C.I.}$, t_{run}) become worse and worse.

For different P_t, we rerun PS-ABCII with the same population, the maximum iteration number and the dim = 30. The results are given in Table 4. With the growth of the parameter P_t, the proportion that employed bees become scouts is bigger and bigger, then it needs more time to exploit new food sources. As seen from Table 4, the runtime becomes obviously big with the growth of the parameter P_t. However, the convergence iteration is smaller and smaller for most optimization problems. Under the given maximum iteration, PS-ABCII could find or be very close with the theoretical global optima on most questions $(f_1,f_2,f_4,f_6,f_8,f_9,f_{10})$. For the questions (f_5,f_7), the found optimum solutions are more and more close with the theoretical global optima with the growth of the parameter P_t. For the question (f_3), its optimum solutions are far away from theoretical global optima when the parameter P_t is growing. So an appropriate parameter P_t could make PS-ABCII find better solutions with less runtime, convergence iteration and the mean of convergence time.

On the whole, the PS-ABCII not only owns a faster convergence iteration, much shorter the achieving convergence time and the running time than ABC and PS-ABC, but also finds better solutions. So it is concluded that PS-ABCII is much more effective and propitious to be applied to real-world optimization problems for the very short running time.

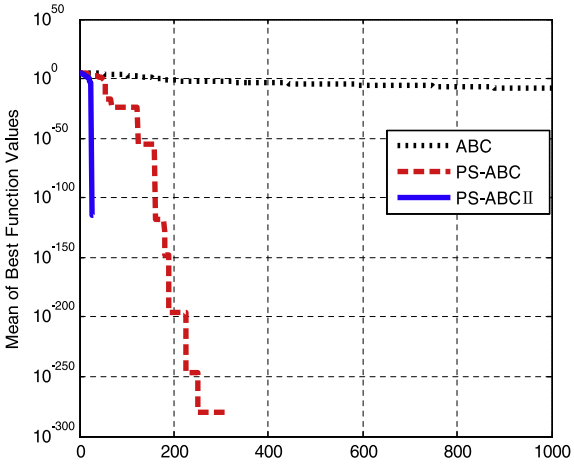

Fig. 3. Comparison of performance of 3 algorithms for minimization of f_1 with dim = 30.

5. Real-world design problem

5.1. Tuning ELM using PS-ABCII

ELM is a simple yet accurate and fast learning algorithm for training single-hidden layer feed-forward artificial neural networks (SLFNs) with random hidden neurons. However, there are

Table 4
Optimization results of PS-ABCII with different fixed proportions P_t and dim = 30.

P_t		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
10%	C.I.	54	23	1000	58	997	959	1000	30	37	35
	Mean	0	0	3.0710×10^4	0	2.8686×10^1	1.5419×10^{-4}	−11515.3	0	8.8817×10^{-16}	0
	S.D.	0	0	1.2307×10^3	0	0.0998	1.4786×10^{-4}	2.7041×10^2	0	0	0
	$t_{C.I.}$	0.0352	0.0241	1.5401	0.0464	0.7500	1.0860	1.0406	0.0242	0.0354	0.0526
	t_{run}	0.6599	0.6802	1.5401	0.6927	0.7659	1.1297	1.0406	0.7970	1.0172	1.2916
40%	C.I.	30	17	980	54	1000	999	997	26	31	58
	Mean	0	0	4.0756×10^4	0	2.8408×10^1	5.5447×10^{-4}	−12088.9	0	8.8817×10^{-16}	0
	S.D.	0	0	8.1760×10^3	0	0.1154	1.2352×10^{-3}	1.8715×10^2	0	0	0
	$t_{C.I.}$	0.0287	0.0138	1.5662	0.0417	0.9109	1.1582	1.2260	0.0287	0.0410	0.0958
	t_{run}	0.7885	0.7510	1.5995	0.7052	0.9109	1.1592	1.2328	0.9849	1.1437	1.5161
60%	C.I.	27	22	971	46	1000	989	999	24	26	24
	Mean	0	0	4.7387×10^4	0	2.8357×10^1	5.1020×10^{-4}	−12101.6	0	8.8817×10^{-16}	0
	S.D.	0	0	7.4218×10^3	0	0.1449	5.0582×10^{-4}	2.2142×10^2	0	0	0
	$t_{C.I.}$	0.0390	0.0219	1.5968	0.0444	0.9295	1.1720	1.5471	0.0326	0.0375	0.7201
	t_{run}	0.8359	0.8123	1.6458	0.8005	0.9295	1.1880	1.5630	1.0505	1.2970	1.6560

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

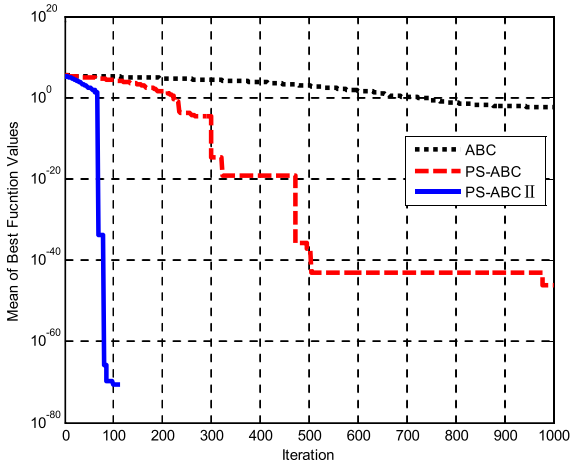

Fig. 4. Comparison of performance of 3 algorithms for minimization of f_1 with dim = 100.

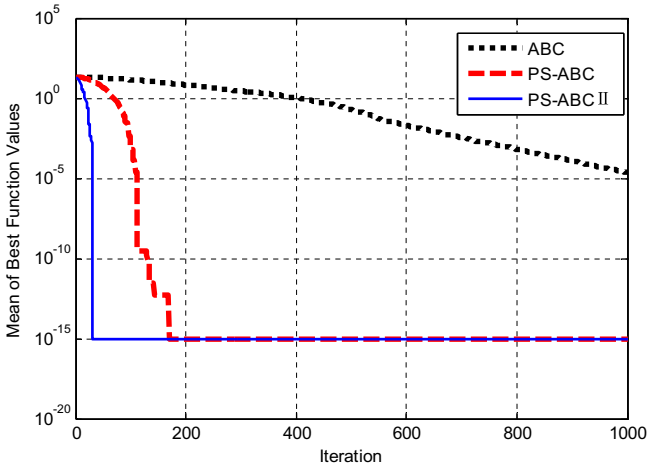

Fig. 7. Comparison of performance of 3 algorithms for minimization of f_9 with dim = 30.

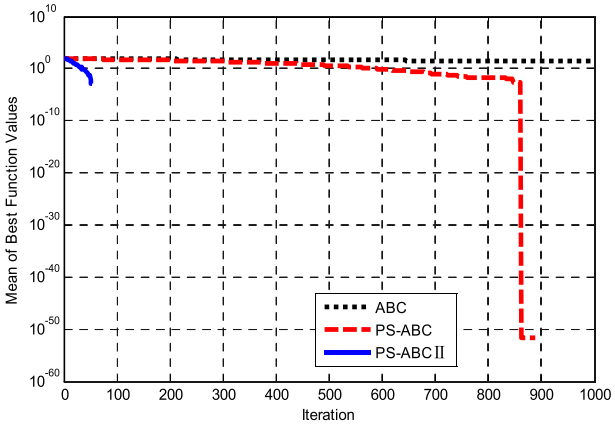

Fig. 5. Comparison of performance of 3 algorithms for minimization of f_4 with dim = 30.

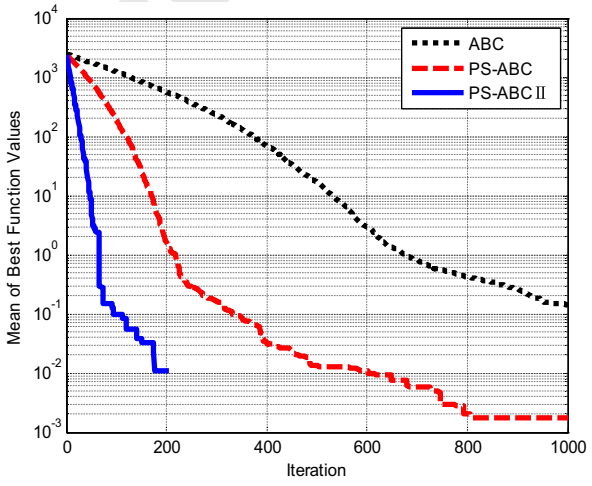

Fig. 8. Comparison of performance of 3 algorithms for minimization of f_{10} with dim = 100.

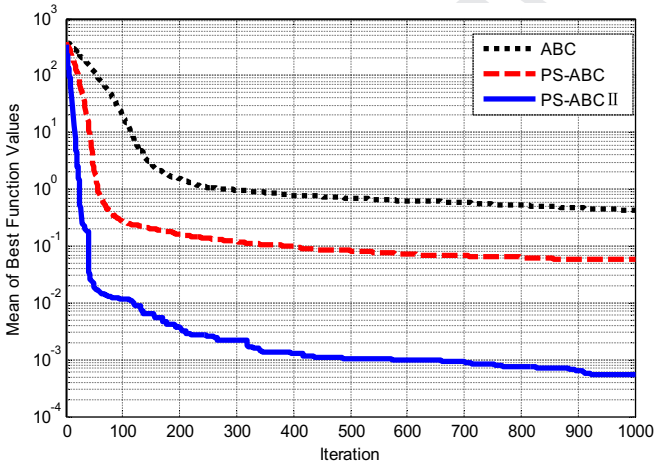

Fig. 6. Comparison of performance of 3 algorithms for minimization of f_6 with dim = 50.

still some shortcomings that restrict the further development of ELM. Randomly choosing of input weights and biases easily causes the hidden layer output matrix not full column rank. This sometimes makes the linear system that is used to train output weights unsolvable, which means the system may have large condition number [37]. It also lowers the predicting accuracy. Therefore, it is necessary to develop a more effective learning method that can overcome this shortcoming.

In this section, this paper presents a new evolutionary ELM model called as PS-ABCII-ELM which adopts the PS-ABCII algorithm to tune the input weights and biases of ELM. For a given N observation samples, we firstly split observation samples into training samples N_{train} and testing samples N_{test}. And then we randomly initialize the population. Each individual in the population is composed of a set of input weights and hidden biases, namely the dimension of each individual θ is $(n+1)\times M$.

$$\theta = [w_{11}, w_{12}, \ldots, w_{1M}, w_{21}, w_{22}, \ldots, w_{2M}, \ldots, w_{n1}, w_{n2}, \ldots, w_{nM}, b_1, b_2, \ldots, b_M] \quad (18)$$

where w_{ij} are randomly initialized within the range of $[-1, 1]$, and b_i are $[0, 1]$. n and M is the numbers of input neurons and hidden neurons.

Thirdly, for each individual, the corresponding output weights are analytically computed by Eq. (6).

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

Table 5
Performance comparison for 4 methods.

Criteria	ELM		ABC-ELM		PS-ABC-ELM		PS-ABCII-ELM	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MAPE (%)	0.36304	0.36023	0.29636	0.29128	0.26178	0.28597	0.26251	0.28975
RMSE	5.7732×10^{-3}	4.0952×10^{-3}	3.3768×10^{-3}	3.5953×10^{-3}	3.1843×10^{-3}	3.3677×10^{-3}	3.1696×10^{-3}	3.3332×10^{-3}
R	0.98882	0.98992	0.99247	0.99226	0.99365	0.99337	0.99371	0.99328
R^2	0.97775	0.97993	0.98500	0.98458	0.98734	0.98679	0.98747	0.98660
T (s)	0.03125		1849.86		3412.72		919.45	

The objective function of each individual is defined as the root mean squared error (RMSE) of training samples. The objective function is the following form:

$$f(\theta) = \sqrt{\frac{\sum_{j=1}^{N_{training}} \left\| \sum_{i=1}^{M} \boldsymbol{\beta}_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - t_j \right\|_2^2}{M \times N_{training}}} \tag{19}$$

The next step, the fitness should be evaluated. According to fitness, the PS-ABCII could search the best input weights and hidden biases for ELM in the limited maximum iteration.

5.2. Modeling the boiler thermal efficiency

In ELM, the input weights are randomly chosen and the output weights are analytically calculated. The generalization performance of the ELM algorithm depends critically on the input weights and the bias values. Then these parameters need to be optimally chosen in order to achieve better generalization performance. Here we adopt PS-ABCII to optimize the input weights and the hidden layer biases for a given number of hidden neurons H, we call the optimized model as PS-ABCII-ELM.

If the fuel and environment conditions are specified, the combustion efficiency of a given boiler mainly depends on various operating parameters such as the air to fuel ratio, and on the distribution of air and fuel to burners at different locations if more burners are used. For different coal quality and different furnace configurations, the best air to fuel ratio and the best air and fuel distributions are surely different. Due to the complexity, uncertainty, non-stability, and nonlinearity of combustion process, it's difficult to set up an accurate mathematical model of a boiler's combustion process by the theory of thermodynamics [38–41]. So this paper adopts the hybrid PS-ABCII-ELM to model the boiler efficiency in order to obtain good approximations and generalization performances, and simultaneously provide a good basis for tuning boiler operating parameters to improve combustion efficiency. In addition, in order to validate the proposed hybrid PS-ABCII-ELM model, 600 samples are gained from a 300 MW coal-fired boiler. These samples are divided into 2 parts: 500 samples as training set and 100 samples as testing set. The proposed PS-ABCII-ELM model in Section 5.A is applied to establish the mathematical model of the boiler efficiency, which illuminates the mapping relation between the boiler efficiency and 16 operational conditions of the boiler. Here, the boiler efficiency is taken as the output of the hybrid model. The following 16 variables are chosen as inputs of the hybrid model.

The 16 characterizing operational conditions of the boiler, including: (1) coal feeder feeding rate (t/h), five variables; (2) the air flow rate through the mill (t/h), five variables; (4) total air rate of the secondary air (t/h), one variable; (5) oxygen concentration in the flue gas (%), two variables; (6) load (MW) one variable; (7) the carbon content of fly ash, two variables.

In this paper, the maximum iteration is set as 1000, the colony size is taken as 40 and the proportion P_t that employed bees would become scouts is set as 40%. The dimension is set the number which equals to $N_{in} \times H + H$, where N_{in} is 16 that are the inputs to ELM, $H = 20$ is the number of hidden neurons. The parameters of ABC and PS-ABC are the same as PS-ABCII, in addition, the parameter 'limit' is set as 200 for ABC and PS-ABC.

In order to state the PS-ABCII-ELM model validity, it is compared with other three methods (ELM, ABC-ELM, PS-ABC-ELM), where the ABC-ELM and PS-ABC-ELM are respectively adopted the ABC and PS-ABC to optimize the input weights and hidden layer biases according to the objective function Eq. (19). In addition, in order to the proposed model superiority, five performance criteria: the mean absolute percentage error (MAPE), root mean square error (RMSE), the correlation coefficient (R), the coefficient of determination (R^2) and the runtime (T) of obtaining the ELM parameters are given as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{20}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{21}$$

$$R = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{23}$$

The two performance criteria: MAPE and RMSE are employed to estimate the prediction performance. The smaller the values of MAPE and RMSE are, the better the prediction performance of a model. The criteria R and R^2 denotes the degree of matching between the predicted and original values. The closer R^2 is to 1,

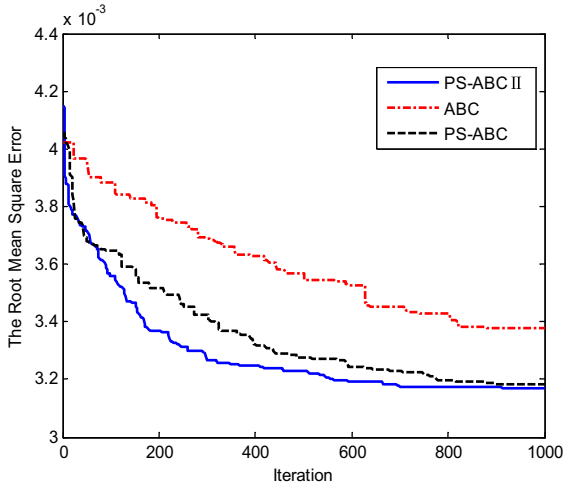

Fig. 9. The search process of input weights and the bias values of ELM.

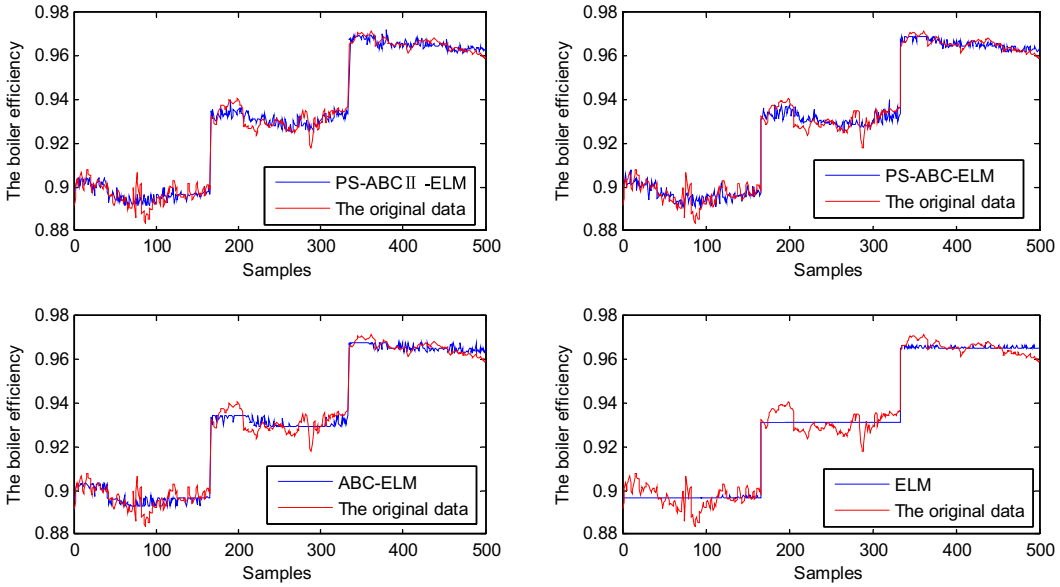

Fig. 10. Compare outputs of the new model with the original system for training set.

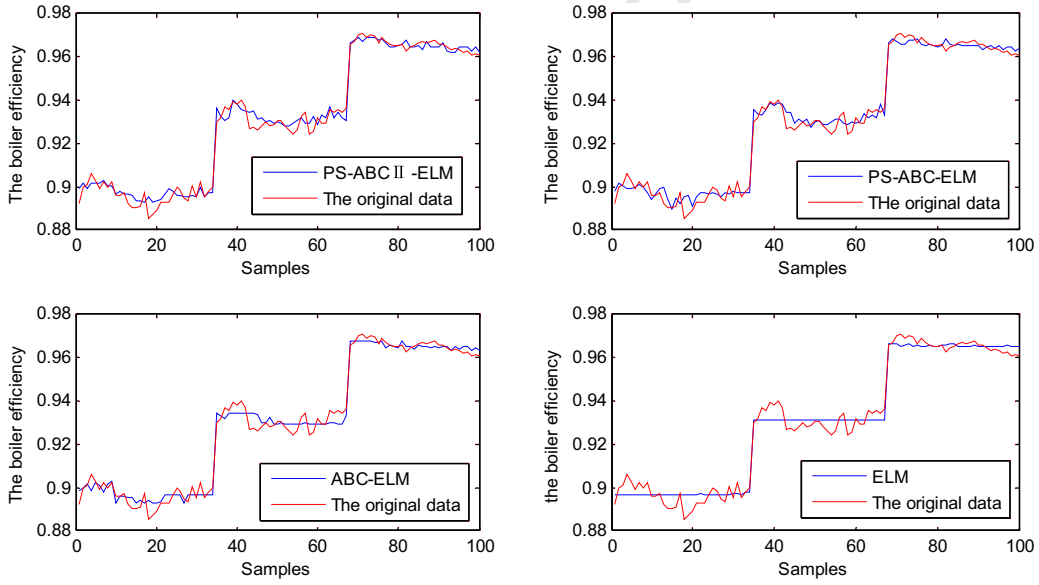

Fig. 11. Compare outputs of the new model with the original system for testing set.

the more accurate the prediction performance. The performance comparison results are given in Table 5. The optimized process of input weights and hidden layer biases of ELM is given in Fig. 9. In addition, the output of the proposed model is compared with the original boiler efficiency and the outputs of the other three models, which is shown in Figs. 10 and 11.

As seen from Fig. 9–11 and Table 5, PS-ABCII could find better solutions than either of ABC and PS-ABC, and simultaneously expends less time to obtain input weights and hidden layer biases. Although the MAPE of PS-ABCII-ELM is a little bigger than that of PS-ABC-ELM, it is much less than those of ELM and ABC-ELM. For the performance criteria: MAPE, the best model is the PS-ABC-ELM, the next is PS-ABCII-ELM, and the worst one is the ELM. However, for the criteria RMSE, The RMSE of PS-ABCII-ELM could achieve 3.1696×10^{-3} for training samples and 3.3332×10^{-3} for testing samples, these values are the lest among the four models, and for the R and R^2, their values of PS-ABCII-ELM are the biggest. In addition, the PS-ABC-IIneeds much less running time to obtain better input weights and hidden layer biases than the other two methods (ABC-ELM and PS-ABC-ELM). Although the running time of ELM is much less than the PS-ABCII-ELM, the regression precision and generalization ability of the PS-ABCII-ELM are much better than those of ELM. So the model PS-ABC-II-ELM shows better nonlinear identification and generalization ability.

In a word, The PS-ABCII-ELM model can achieve good prediction effects under various operating conditions, and then it could be adopted to model the boiler efficiency. It is verified with the experimental results, which show that the PS-ABCII-ELM model is very convenient, direct and accurate, and it can give a general and suitable way to predict the boiler efficiency of a coal-fired boiler under various operating conditions.

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

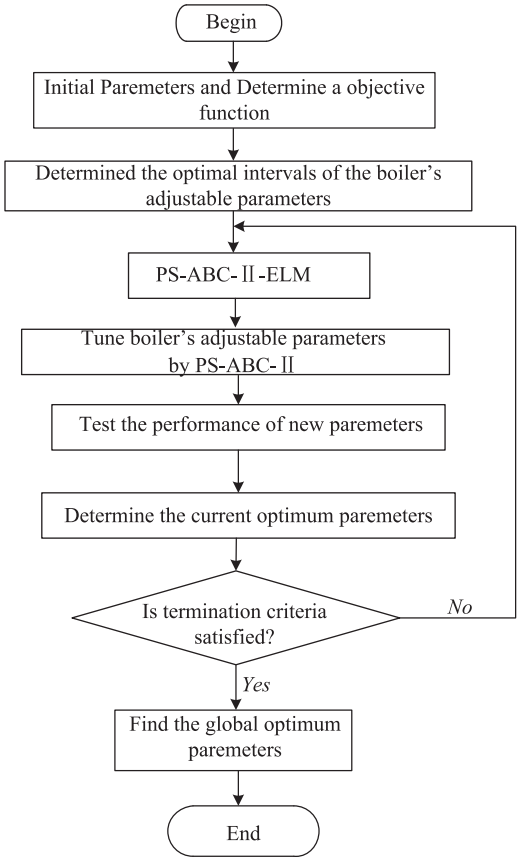

Fig. 12. Flowchart of the coal-fired boiler's combustion optimization.

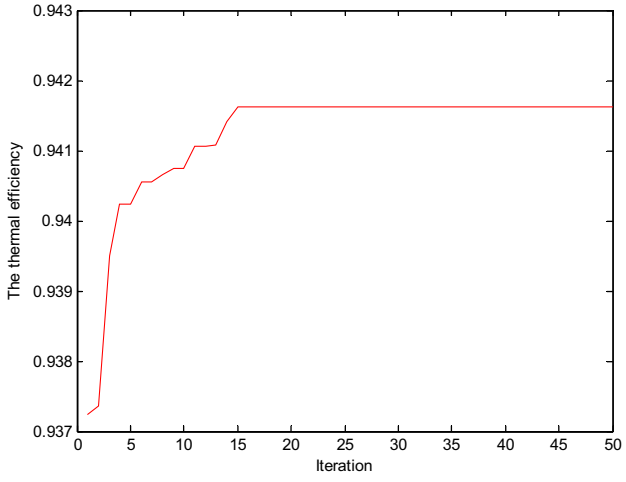

Fig. 13. The optimized process of the boiler efficiency by PS-ABCII for Case 372.

5.3. Improving the boiler thermal efficiency

The purpose of a boiler combustion optimization is to tune the boiler adjustable parameters in order to make the boiler maintain an optimum combustion state under the premise of the assurance of safety and reliability. Therefore, the PS-ABCII would be adopted to optimize adjustable parameters based on the PS-ABCII-ELM model which may be considered simply as mapping functions between operational conditions and the output (the thermal efficiency).

In PS-ABCII, the colony size is set as 40, the max iteration is 50, P_t is set as 40% and its objective function is given as the following form:

$$\max\ f(\boldsymbol{x}) = f_\eta(\boldsymbol{x}) \tag{24}$$

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}] \tag{25}$$

$$s.t.\ a_i \leqslant x_i \leqslant b_i \tag{26}$$

In which a_i, b_i is the upper and lower limits of the ith adjustable parameter x_i (Coal feeder feeding rate; The air flow rate through the mill; Total air rate of the secondary air; Oxygen concentration in the flue gas).

Here, we make the boiler achieve combustion optimization under 3 cases (Case 13, Case 372, Case 421). Namely, for the 3 cases, their non-adjustable parameters (Load, the carbon content of fly ash) remains unchanged, and the adjustable parameters (Coal feeder feeding rate; The air flow rate through the mill; Total air rate of the secondary air; Oxygen concentration in the flue gas) are tuned by PS-ABCII under their respective optimal interval. The optimization process of Case 372 is given in Fig.13, and the optimized adjustable parameters and the efficiency of 3 cases are given in Table 6 (see Fig.12).

As seen from Table 6, all the optimized efficiencies of 3 cases are improved by tuning adjustable parameters. For Case 13, Case 372 and Case 421, the net increments of optimized boiler thermal efficiencies are 0.8116%, 0.7265% and 0.5493%, respectively. In addition, the PS-ABCII could find the optimum efficiency very quickly based on the PS-ABCII-ELM model, and make the boiler achieve the objective of combustion optimization. Therefore, the PS-ABCII could be adopted to search the optimized operational parameters.

Table 6
Optimized adjustable parameters by PS-ABCII for the effiecieny of the 300 MW boiler.

Parameters		Case 13		Case 372		Case 421	
		Measured	Optimized	Measured	Optimized	Measured	Optimized
Coal feeder feeding rate (t/h)	A	−0.0615	−0.0615	−0.0615	−0.0615	43.3336	43.3336
	B	44.9164	39.9482	44.9604	40.8715	48.0382	47.2027
	C	44.7845	36.6947	41.6629	38.0576	−0.0176	−0.0176
	D	−0.0175	−0.0175	36.2550	39.4206	50.720	52.0830
	E	30.9350	30.0557	38.0137	34.5403	48.8735	46.1915
The air flow rate through the mill (t/h)	A	0	0	0	2.0177	150	150
	B	75.1718	70.2538	119.3484	113.1951	97.6221	90.3850
	C	109.447	118.2517	115.6936	122.3898	0	0
	D	0	0	98.2334	89.1030	89.6061	81.5809
	E	96.9719	79.4258	108.6741	104.0554	100.3284	96.9720
Total air rate of the secondary air (t/h)		406.2938	414.0853	527.2554	537.7491	705.0360	733.0919
Oxygen concentration in the flue gas (%)		4.5986	4.2523	4.1850	3.9061	2.7587	2.6397
Efficiency (%)		90.1371	90.9487	93.4352	94.1617	97.0954	97.6447

The negative values indicate that the corresponding equipment did not work.

Q1 Please cite this article in press as: G. Li et al., Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowl. Based Syst. (2014), http://dx.doi.org/10.1016/j.knosys.2014.04.042

6. Conclusion

A sophisticated PS-ABCII algorithm for application to real-world optimization problems is proposed to fasten the convergence time and running time, and simultaneously improve the search ability. The new optimization algorithm is applied to 10 classical test function problems, and the results indicate that the convergence time and running time of PS-ABCII is much shorter than that of ABC and PS-ABC, and the search performance is more outstanding than the other two algorithms. Due to these advantages, the PS-ABCII is more suitable for application in actual engineering. In addition, a novel model PS-ABCII-ELM, which is combined PS-ABCII with ELM, is also presented in this paper. In order to validate the proposed model, it is employed to model the thermal efficiency of a 300 MW coal-fired boiler. Experimental results show that the model is very convenient, direct and accurate, and it could be a suitable and effective technique to predict and optimize the boiler efficiency of a coal-fired boiler under various operating conditions. In the future, the PS-ABCII and the new model would be used for the boiler combustion optimization in order to reduce the NO_x emissions while improving or keeping the thermal efficiency of a coal-fired boiler.

Acknowledgements

Q4 Q5 This work was supported in part by the National Natural Science Foundation of China (Grant no. 60774028) and Natural Science Foundation of Hebei Province, China (Grant no. F2010001318).

References

[1] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Erciyes University, Kayseri, Turkey, Technical Report-TR06, 2005.
[2] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, LNCS: Advances in Soft Computing-Foundations of Fuzzy Logic and Soft Computing, vol. 4529, Springer-Verlag, 2007, pp. 789–798.
[3] D. Karaboga, B. Akay, C. Ozturk, Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, LNCS: Modeling Decisions for Artificial Intelligence, vol. 4617, Springer-Verlag, 2007, pp. 318–329.
[4] D. Karaboga, C. Ozturk, Neural networks training by artificial bee colony algorithm on pattern classification, Neural Network World 19 (3) (2009) 279–292.
[5] D. Karaboga, C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, Appl. Soft Comput. 11 (1) (2011) 652–657.
[6] C. Ozturk, D. Karaboga, B. Gorkemli, Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm, Sensors 11 (6) (2011) 6056–6065.
[7] D. Karaboga, S. Okdem, C. Ozturk, Cluster based wireless sensor network routing using artificial bee colony algorithm, Wireless Netw. 18 (7) (2012) 847–860.
[8] D. Karaboga, C. Ozturk, N. Karaboga, B. Gorkemli, Artificial bee colony programming for symbolic regression, Inform. Sci. 209 (2012) 1–15.
[9] Dervis Karaboga, Bahriye Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[10] Yueh-Min Huang, Jin-Chen Lin, A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems, Expert Syst. Appl. 38 (2011) 5438–5447.
[11] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, J. Franklin Inst. 346 (2009) 328–348.
[12] Anan Banharnsakun, Tiranee Achalakul, Booncharoen Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, Appl. Soft Comput. 11 (2011) 2888–2901.
[13] Guopu Zhu, Sam Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010) 3166–3173.
[14] R. Akbari, A. Mohammadi, K. Ziarati, A novel bee swarm optimization algorithm for numerical function optimization, Commun. Nonlinear Sci. Number Simul. 15 (2010) 3142–3155.
[15] Guoqiang Li, Peifeng Niu, Xingjun Xiao, Development and investigation of efficient artificial bee colony algorithm for numerical function optimization, Appl. Soft Comput. 12 (2012) 320–332.
[16] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[17] S. Suresh, S. Saraswathi, N. Sundararajan, Performance enhancement of extreme learning machine for multi-category sparse data classification problems, Eng. Appl. Artif. Intell. 23 (2010) 1149–1157.
[18] Jianwei Zhao, Zhihui Wang, DongSun Park, Online sequential extreme learning machine with forgetting mechanism, Neurocomputing 87 (2012) 79–89.
[19] Fei Han, De-Shuang Huang, A new learning algorithm for function approximation incorporating a priori information into extreme learning machine, Lecture Notes in Computer Science, LNCS, vol. 3971, 2006, pp. 631–636.
[20] Guoqiang Li, Peifeng Niu, An enhanced extreme learning machine based on ridge regression for regression, Neural Comput. Appl. (2011), http://dx.doi.org/10.1007/s00521-011-0771-7.
[21] Z.-L. Sun, T.-M. Choi, K.-F. Au, Y. Yu, Sales forecasting using extreme learning machine with applications in fashion retailing, Decis. Support Syst. 46 (1) (2008) 411–419.
[22] A.H. Nizar, Z.Y. Dong, Y. Wang, Power utility nontechnical loss analysis with extreme learning machine method, IEEE Trans. Power Syst. 23 (3) (2008) 946–955.
[23] Chee-Wee Thomas Yeu, Meng-Hiot Lim, Guang-Bin Huang, Amit Agarwal, Yew-Soon Ong, A new machine learning paradigm for terrain reconstruction, IEEE Geosci. Remote Sens. Lett. 3 (3) (2006) 382–386.
[24] Guoren Wang, Yi Zhao, Di Wang, A protein secondary structure prediction framework based on the extreme learning machine, Neurocomputing 72 (2008) 262–268.
[25] G. Feng, G.-B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw. 20 (8) (2009) 1352–1357.
[26] Y. Lan, Y.C. Soh, G.-B. Huang, Ensemble of online sequential extreme learning machine, Neurocomputing 72 (2009) 3391–3395.
[27] Guohu Li, Min Liu, Mingyu Dong, A new online learning algorithm for structure-adjustable extreme learning machine, Comput. Math. Appl. 60 (2010) 377–389.
[28] H.-J. Rong, Y.-S. Ong, A.-H. Tan, Z.-X. Zhu, A fast pruned-extreme learning machine for classification problem, Neurocomputing 72 (2008) 359–366.
[29] Q.Y. Zhu, A.K. Qin, P.N. Suganthan, G.B. Huang, Evolutionary extreme learning machine, Pattern Recogn. 38 (10) (2005) 1759–1763.
[30] D.N.G. Silva, L.D.S. Pacifico, T.B. Ludermir, An evolutionary extreme learning machine based on group search optimization, in: 2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, pp:574-580.
[31] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (16–18) (2008) 3460–3468.
[32] G.B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16–18) (2007) 3056–3062.
[33] F. Han, D.S. Huang, Improved extreme learning machine for function approximation by encoding a priori information, Neurocomputing 69 (16–18) (2006) 2369–2373.
[34] Shahryar Rahnamayan, Hamid R. Tizhoosh, Magdy M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79.
[35] Hui Wang, Zhijian Wu, Shahryar Rahnamayan, Yong Liu, Mario Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Inform. Sci. 181 (2011) 4699–4714.
[36] Shahryar Rahnamayan, Hamid R. Tizhoosh, Magdy M.A. Salama, Opposition versus randomness in soft computing techniques, Appl. Soft Comput. 8 (2008) 906–918.
[37] Guopeng Zhao, Zhiqi Shen, Chunyan Miao, Zhihong Man, On Improving the conditioning of extreme learning machine: a linear case, in: The 7th International Conference on Information, Communications and Signal Processing (ICICS), 2009. pp. 1–5.
[38] Yi Zhang, Yanjun Ding, Zhansong Wu, Liang Kong, Tao Chou, Modeling and coordinative optimization of NO_x emission and efficiency of utility boilers with neural network, Korean J. Chem. Eng. 24 (6) (2007) 1118–1123.
[39] Hao Zhou, JiaPei Zhao, LiGang (Zheng), ChunLin Wang, KeFa Cen, Modeling NO_x emissions from coal-fired utility boilers using support vector regression with ant colony optimization, Eng. Appl. Artif. Intell. 25 (2012) 147–158.
[40] Guoqiang Li, Peifeng Niu, Chao Liu, Weiping Zhang, Enhanced combination modeling method for combustion efficiency in coal-fired boilers, Appl. Soft Comput. 12 (2012) 3132–3140.
[41] Zhe Song, Andrew Kusiak, Constraint-based control of boiler efficiency: a data-mining approach, IEEE Trans. Ind. Inform. 3 (1) (2007) 3–83.