Robotica (2023), **41**, pp. 1087–1097
doi:10.1017/S026357472200145X

RESEARCH ARTICLE

An improved Kalman particle swarm optimization for modeling and optimizing of boiler combustion characteristics

Jing Liang[1], Hao Guo[1], Ke Chen[1,*], Kunjie Yu[1], Caitong Yue[1] and Xia Li[2]

[1]School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China and [2]School of Mathematics and Information Technology, Hebei Normal University of Science and Technology, Qinhuangdao 066004, China
*Corresponding author. E-mail: chenkezixf@zzu.edu.cn

Received: 23 June 2022; **Revised:** 9 September 2022; **Accepted:** 13 September 2022; **First published online:** 24 October 2022

Keywords: circulating fluidized bed boiler, particle swarm optimization, extreme learning machine, Kalman filtering

Abstract
With the rapid development of the national economy, the demand for electricity is also growing. Thermal power generation accounts for the highest proportion of power generation, and coal is the most commonly used combustion material. The massive combustion of coal has led to serious environmental pollution. It is significant to improve energy conversion efficiency and reduce pollutant emissions effectively. In this paper, an extreme learning machine model based on improved Kalman particle swarm optimization (ELM-IKPSO) is proposed to establish the boiler combustion model. The proposed modeling method is applied to the combustion modeling process of a 300 MWe pulverized coal boiler. The simulation results show that compared with the same type of modeling method, ELM-IKPSO can better predict the boiler thermal efficiency and NOx emission concentration and also show better generalization performance. Finally, multi-objective optimization is carried out on the established model, and a set of mutually non-dominated boiler combustion solutions is obtained.

1. Introduction
With the advancement of modernization, people's consumption of electricity is gradually increasing. Thermal power generation uses lots of coal for power generation, leading to severe atmospheric pollution problems. The problem of how to increase boiler thermal efficiency and reduce pollutants has become an urgent problem to be solved. Establishing an accurate combustion model is the premise of optimization. However, the combustion of circulating fluidized bed boilers (CFBBs) has the characteristics of nonlinearity, strong coupling, and pure lag [1]. Jagtap *et al*. [2] introduced the application of the Markov probability method in boiler power generation reliability and used particle swarm optimization (PSO) to optimize the combustion process. Santra *et al*. [3] researched the synthesis of dissipative fault-tolerant cascade control for a class of singular networked cascade control systems with differentiable and non-differentiable time-varying delays, and they ultimately confirmed the viability of the suggested approach using a boiler-turbine unit from a power plant. Adams *et al*. [4] developed a deep neural network based on least-squares support vector machine to predict pollutants such as SOx and NOx produced in thermal power combustion. Even though these methods are effective, the calculations are complex, making them challenging to apply in practical engineering.

Artificial neural network (ANN) technology has many advantages, such as strong nonlinear fitting ability, good algorithm stability, strong self-learning ability, and good repeatability [5]. Therefore, ANN technology has been successfully applied in many fields, such as sample classification [6], speech recognition [7], intelligent control [8], and regression approximation [9]. However, ANN has problems such as slow learning speed, low generalization ability, and easy to fall into local optimal, which limit its use in some applications. Extreme learning machine (ELM) was proposed [10] in 2004. The ELM is a single hidden layer feedforward neural network [11], its input weights and hidden layer thresholds are randomly set and remain unchanged during the processing of data samples, and the output weights are obtained through the least-squares method [12]. Compared with the traditional ANN, ELM has a faster operation speed and overcomes the problems of iterative calculation.

However, ELM also has some shortcomings, mainly including two aspects: the setting of the number of hidden layer nodes and the randomly set input weights and hidden layer thresholds, which may lead to the problems of low model accuracy and poor generalization ability. In order to solve these problems, many scholars proposed some improved ELM models. For example, Zhu *et al*. [13] used a differential evolution algorithm to select input weights and hidden layer thresholds. This method not only improved the generalization ability of ELM but also made the model structure of ELM more compact. Cao *et al*. [14] proposed an adaptive evolutionary extreme learning machine (SaELM), used an adaptive evolutionary algorithm to optimize the hidden layer node parameters of ELM, and achieved good generalization ability. Matias *et al*. [15] proposed an optimized ELM, which used an optimization algorithm to optimize the network structure and parameters of the ELM. In ref. [16], an improved PSO was proposed. It was used to optimize the input weights and hidden layer thresholds of ELM, overcome the ill-conditioned problem of ELM, and improve the compactness of the ELM.

All the above-improved ELM methods use an evolutionary algorithm to find the input weights and hidden layer thresholds. Although the accuracy and generalization ability of the model were improved, the evolutionary algorithm used in this high-dimensional optimization problem generally has the drawbacks of fast convergence and is easy to converge into local optimization. The Kalman particle swarm optimization (KPSO) [17] combines the Kalman filter principle into the PSO [18], which reduces the number of iterations for the algorithm to find the global optimum in solving high-dimensional optimization problems. After combining the principle of the Kalman filter, the optimization ability of PSO has been improved, and it has been further improved and applied to practical engineering [19]. This paper proposes an improved Kalman particle swarm optimization (IKPSO) for modeling the combustion characteristics of CFBB. In IKPSO, the population is adaptively divided into a convergent state and a divergent state in each search process, and then the individuals in the convergent state are corrected by the Kalman filter principle. If the population has not updated the global best (*gbest*) for several generations, the entire population will be revised. By comparing with the other evolutionary algorithms, the proposed algorithm obtains a more accurate model. Finally, multi-objective optimization is carried out on the obtained model to increase the thermal efficiency of the boiler and reduce the concentration of NOx.

The remaining sections of this paper are as follows: Section 2 introduces the fundamentals of ELM and KPSO. Section 3 presents the proposed IKPSO algorithm. Section 4 lists the experiments and analysis of the results. Section 5 provides a summary and outlook.

2. Related work

2.1. Extreme learning machine

As a variant of the ANN, the ELM does not require gradient-based backpropagation to adjust the weights but sets the weights through the Moore–Penrose generalized inverse [20]. The standard ELM neural network structure is shown in Fig. 1. For any N samples (x_i, t_i), $x_i = [x_1^i, x_2^i, \cdots, x_n^i]^{\mathrm{T}} \in R^n$ is the input vector of the i sample, n is the number of input layer nodes, $t_i = [t_1^i, t_2^i, \cdots, t_l^i]^{\mathrm{T}} \in R^l$, l is the number of output layer nodes. The number of hidden layer nodes is m, and the hidden layer activation function is $g(x)$. Input weights ω, hidden layer thresholds b, and output weights β are $m \times n$, $m \times 1$, and $l \times m$ matrix, respectively. Then the mathematical model is as follows:

$$t_k = \sum_{i=1}^{m} \beta_i g_i(\omega, b, x_k) = \sum_{i=1}^{m} \beta_i g_i(\omega_i \cdot x_k + b_i), (k = 1, 2, \cdots, N) \tag{1}$$

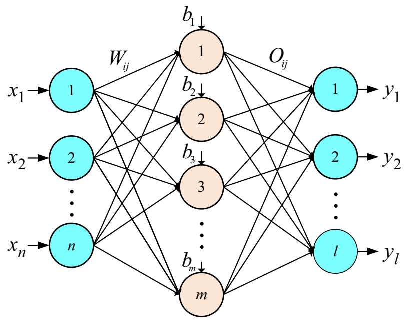

Figure 1. *ELM network structure diagram.*

which ω_i is $1 \times n$, the weights of the input layer nodes and the i hidden layer node. x_k is $n \times 1$, the i th input sample. β_i is $l \times 1$, the weights of the i hidden layer node and the output layer nodes. t_k is $l \times 1$, output of the i sample. The above N equations can be abbreviated as:

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

$$H = \begin{bmatrix} g(\omega_1 x_1 + b_1) & \cdots & g(\omega_m x_1 + b_m) \\ \vdots & \ddots & \vdots \\ g(\omega_1 x_N + b_1) & \cdots & g(\omega_m x_N + b_m) \end{bmatrix}_{N\times m}; \beta = [\beta_1, \beta_2, \cdots \beta_l]^{\mathrm{T}}_{l\times m}; T = [t_1, t_2, \cdots t_N]^{\mathrm{T}}_{N\times l} \tag{3}$$

H is called the hidden layer output matrix, β is the output layer weight matrix, and T is the expected output. According to the least square norm solution of the above equation [21], it can be obtained that $\hat{\beta} = H^{\dagger}T$, $H^{\dagger}$ is the generalized inverse matrix of H.

2.2. *Kalman particle swarm optimization*

The principle of Kalman filtering makes an optimal estimate of the final state by referring to the predicted and observed values [22]. Specifically, given an observation Z_{t+1}, KPSO is used to generate a Gaussian distribution about the true state. The parameters m_{t+1} and V_{t+1} of this multivariate distribution are determined by the following equations:

$$m_{t+1} = Fm_t + K_{t+1}(Z_{t+1} - GFm_t) \tag{4}$$

$$V_{t+1} = (I - K_{t+1})(FV_tF^{\mathrm{T}} + V_X) \tag{5}$$

$$K_{t+1} = (FV_tF^{\mathrm{T}} + V_X)G^{\mathrm{T}}(G(FV_tF^{\mathrm{T}} + V_X)G^{\mathrm{T}} + V_Z)^{-1} \tag{6}$$

where F and V_X describe the system transition model and process noise, while G and V_Z describe the system sensor model and observation noise [23]. The best estimate of the true state results from a Gaussian distribution:

$$x_t \sim Normal(m_t, V_t) \tag{7}$$

KPSO prescribes particle changes entirely based on Kalman's prediction principles. Every particle has its m_t, V_t, and K_t. The particles then generate observations with the following equation:

$$z_v = \phi(g - x); \quad z_p = x + z_v; \quad Z = (z_p^T, z_v^T) \tag{8}$$

where ϕ is uniformly extracted in the interval [0.2, 0.5]. Then m_t and V_t are generated from the observations of Eqs. (4), (5), and (6). After the observations are determined, the optimal estimated state of the particle is obtained by Eq. (7). The new state of the particle is obtained by sampling this distribution, and the position information of particles is extracted from x_{t+1}. The system model parameters are set as follows, where d is the individual dimension, and o is the individual vector [17].

$$F = \begin{pmatrix} I_d & I_d \\ 0 & I_d \end{pmatrix}; V_Z = diag(o); V_X = diag(o); G = I_{2d} \tag{9}$$

3. Proposed method

3.1. KPSO using Cauchy distribution

Traditional KPSO uses complex computations when updating particle predictions and covariance matrices, which can be time-consuming in practical engineering applications. In this paper, KPSO is improved according to the calculation method of Kalman gain in ref. [19], and the Cauchy distribution is used to generate the optimal estimate. The Cauchy distribution has the characteristics of no expectation and no variance, and its probability density function is

$$f(x) = \frac{\beta}{\pi\left[\beta^2 + (x-\alpha)^2\right]}, \beta > 0 \tag{10}$$

where α is the position parameter that defines the peak position of the distribution, and β is the scale parameter that defines the half-width at half the maximum value. There are two reasons for using the Cauchy distribution function: (1) The existing KPSOs use a Gaussian distribution for optimal estimation, and the velocity information of particles is used as the covariance matrix of the distribution. However, the velocity information is directional, which does not meet the requirements of a positive definite covariance matrix, and the calculation is also complex; (2) When using the Cauchy distribution function to generate offspring, it has a certain probability of generating a mutation value. Due to the non-expectation and non-variance nature of the Cauchy distribution function, mutation value is allowed, which increases the diversity of the population in the search process and is more accommodating to search. In summary, when using Cauchy distribution to generate offspring, the diversity of the population is enhanced and the computational efficiency is improved.

3.2. Hierarchical division in KPSO search process

PSO was proposed by Eberhart and Kennedy in 1995 [24]. PSO is a swarm intelligence algorithm designed by simulating the predation behavior of birds. There are food sources of different sizes in the search space, and birds are tasked with finding the largest food source (*gbest*). Birds in the search process, through mutual transmission of their information, cooperate to find the optimal solution. Essentially, PSO finds *gbest* through an iterative process. In each iteration, each particle produces a new particle. The newly generated particles are called the offspring, and the original particles are called the parents.

In the early stages of PSO search, it is intended that the particles in the population will diverge as much as possible and enhance the global exploration capability. In the late search, particles need to convergence as much as possible to enhance their local exploitation ability. In the proposed algorithm, the distance between the parents and the offspring in each iteration is calculated, and the first $\lfloor W * N \rfloor$ individuals are selected to be defined as the convergent state and the rest as the divergent state, where W is the inertia weight, and N is the population size. The individuals in the convergence state are modified to ensure the exploration and exploitation ability of the algorithm.

3.3. Overview of the proposed approach

The pseudo-code of the IKPSO is shown in Algorithm 1. Firstly, the PSO principle is used to generate the offspring *pops*, and then the Euclidean distance *dis* between *pops* and *pop* is calculated. Secondly, the population is divided into a convergence state and a divergence state according to the distance index,

Algorithm 1: IKPSO

```
Input: $N$: The population size; $maxFE$: Maximum number of fitness evaluations; $N_c$: Number
       of optimal estimates; $N_s$: Maximum number of stagnation updates; $\gamma$: Cauchy
       distribution scale parameter; $fitness$: fitness evaluation function.
Output: $gbest$ : Global optimal solution.
1  $pop \leftarrow$ initialize the population, $fit = fitness(pop), FE = FE + N$;
2  $pbest, pbestvalue, gbest, gbestvalue \leftarrow$ record population information;
3  while $FE < maxFE$ do
4      $pops \leftarrow$ generated offspring using $pop$, $fit = fitness(pops), FE = FE + N$;
5      $pbest, pbestvalue, gbest, gbestvalue \leftarrow$ update population information;
6      $dis \leftarrow \sum_{i \in N} d(pops_i, pop_i)$, $index1 \leftarrow Argsort(dis)$, $index2 \leftarrow Argsort(pbestvalue)$;
7      for $i$ = $1{:}floor(\omega *N)$ do
8          $Q \leftarrow d(pops_{index1[i]}, gbest) = \|pops_{index1[i]} - gbest\|^2$;
9          $R \leftarrow d(pops_{index1[i]}, pbest) = \|pops_{index1[i]} - pbest_{index2[i]}\|^2$;
10         $K = R(R+Q)^{-1}$, $x_1 = pops_{index1[i]} + K * (pops_{rand} - pops_{index1[i]})$;
11         for $j = 1 : N_c$ do
12             $k_j = Cauchy(x_1, \gamma)$, $kfit_j = fitness(k_j)$, $FE = FE + 1$;
13         end
14         $pop_{index1[i]} \leftarrow$ Select the best individual in the set $k$;
15     end
16     Update the population information and record the number of $gbest$ stagnation updates. If
         not updated for $N_s$ consecutive times, all individuals in the population are corrected.
17 end
18 return $gbest$.
```

and the individuals in the convergence state are corrected by Kalman filtering. Finally, the number of the stagnations of *gbest* is calculated. If the upper limit is reached, a Kalman filtering correction is performed on the entire population.

3.4. Fitness function

As mentioned above, the input weights ω and hidden layer thresholds b of ELM are randomly set, and they cannot be guaranteed that the model has good robustness. Aiming at this problem, the artificial intelligence optimization algorithm is used to optimize the input weights and thresholds of ELM to improve the robustness of the model.

In this paper, the IKPSO algorithm is used to optimize ω and b of ELM. Assuming that the number of input nodes of ELM is n and the number of hidden layer nodes is m, the individual dimension to be optimized is $(n+1)*m$. An optimization individual can be represented as $\alpha_i = [\omega_{11}, \omega_{12}, \cdots, \omega_{1n}, \cdots, \omega_{mn}, b_1, \cdots, b_m]$, α_i is the i individual to be optimized, where $\omega_{mn} \in [-1, 1]$, $b_m \in [0, 1]$. The root mean square error (RMSE) function [25] between the output value of the model and the target value is used as the fitness function, as shown in Eq. (11), which $fit(\alpha_i)$ is the fitness value of individual α_i, N_{train} is the number of samples, and Y_i and $f(\alpha_i)$ are the true value and model output value of the i sample, respectively.

$$fit(\alpha_i) = \sqrt{\frac{1}{N_{train}} \sum_{i=1}^{N_{train}} (Y_i - f(\alpha_i))^2} \tag{11}$$

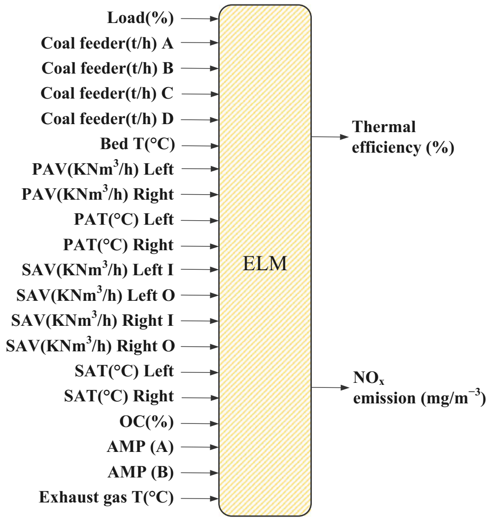

Figure 2. *Simplified CFBB model.*

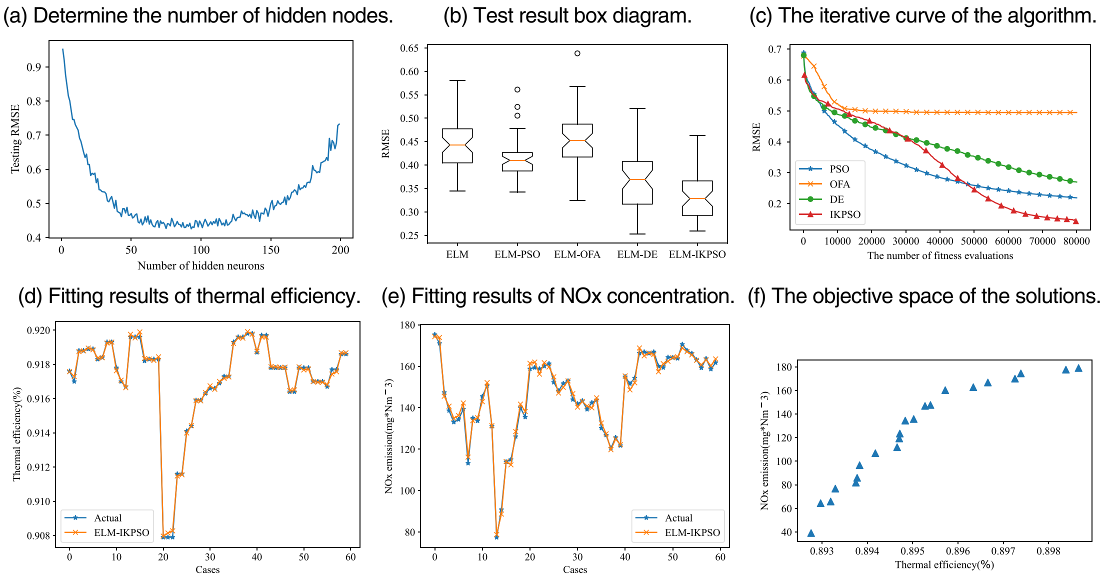

Figure 3. *Figures of the experimental results.*

4. Experiments and result analysis

4.1. Experiment simulation and parameter setting

The experimental data come from a 300 MWe CFBB in China. There are a total of 20 operational parameters and 2 target parameters. A simplified CFBB model is shown in Fig. 2. Therefore, the number of input layer nodes of the ELM is 20, and the number of output layer nodes is 2. The number of hidden layer nodes is set to 80 by parameter verification, and the verification curve is shown in Fig. 3(a).

In order to verify the optimization ability of the proposed IKPSO algorithm, it is compared with PSO [24], KPSO [23], OFA [26], and DE [13] on the CEC2017 benchmark functions [27] in Section 4.2. All experiments are conducted 51 times independently, the dimension of the test functions is 100, the total

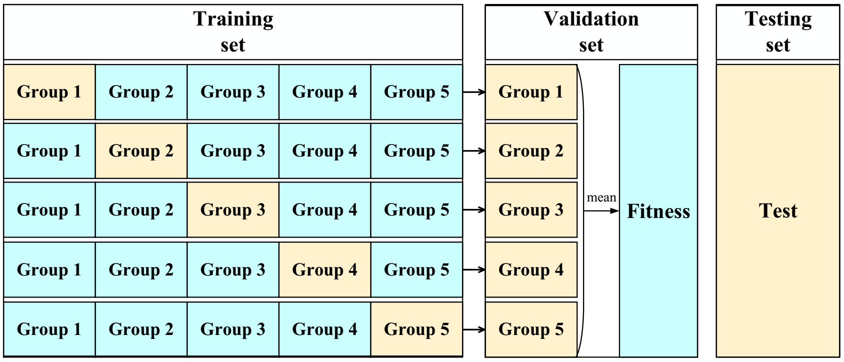

Figure 4. *Simplified CFBB model.*

number of evaluations is 1000000, and the population size is set to 100. The experimental results are verified by the Wilcoxon rank-sum test [28].

In Section 4.3, the proposed ELM based on IKPSO optimization (ELM-IKPSO) is compared with ELM [10], ELM-PSO [16], ELM-OFA [26], and ELM-DE [13], respectively. The optimized ELM has the same parameters. Because the number of hidden layer nodes is set to 80, according to the fitness function setting in Section 3.4, the individual dimension to be optimized is $\alpha = 1680$. The total evaluation times of PSO, DE, OFA, and IKPSO are 80,000, and the population size is 100. The experiments are run independently 51 times and simulated on Python 3.9. In order to prevent model overfitting, the training data set is divided into a training set and a validation set, and the RMSE of the model on the validation set is taken as the optimization goal. The structure diagram of the data set division is shown in Fig. 4. Finally, the statistical test is performed by the Wilcoxon rank-sum test [28], and the confidence level α [29] is set to 0.05. Finally, the proposed method is used for multi-objective optimization on the established model to improve boiler thermal efficiency and reduce NOx emissions.

4.2. CEC2017 benchmark functions validation

In order to test the optimization ability of the algorithm using the Cauchy distribution, several evolutionary algorithms are compared on the CEC2017 benchmark functions. KPSO is the original algorithm using a Gaussian distribution function, and IKPSO is the improved algorithm using a Cauchy distribution function. Detailed experimental results are shown in Table I. The best results are shown in bold. It can be seen from the experimental results that the optimization ability of the algorithm is improved after using the Cauchy distribution function to generate the offspring, and IKPSO is superior to KPSO in all 30 test functions. The reason is that the use of Cauchy distribution to generate offspring can make the population find potential regions for development more quickly and make full use of computing resources. From the comparison of KPSO and IKPSO with PSO, it can be concluded that the algorithm using the Kalman filter principle improves the optimization ability of PSO. In addition, in comparison with other types of evolutionary algorithms, OFA and DE, IKPSO is superior to the compared algorithms in 23 and 20 functions, respectively. In summary, IKPSO is superior to the compared algorithms, showing stronger optimization ability and generalization performance.

4.3. Model establishment and analysis

The detailed results of the modeling algorithms are shown in Table II. As shown in Table II, a comparison of the outcome metrics for ELM, ELM-PSO, ELM-OFA, ELM-DE, and ELM-IKPSO is listed. For each algorithm, the best result is shown in bold font. For the training data, ELM-IKPSO has the smallest performance metric, significantly outperforming the compared algorithms. For the test data, it

Table I. *Experimental results on the CEC2017 test functions.*

Problem	PSO	KPSO	OFA	DE	IKPSO
F1	1.0018e + 11 (1.47e + 10) −	6.1020e + 8 (7.43e + 8) −	**4.6755e + 3 (1.51e + 3)** =	3.5800e + 5 (1.02e + 5) −	6.7939e + 3 (6.23e + 3)
F2	3.3635e + 145 (1.59e + 146) −	5.5273e + 90 (2.71e + 91) −	1.4049e + 118 (2.71e + 118) −	8.0378e + 94 (3.35e + 95) −	**4.7460e + 51 (2.32e + 52)**
F3	3.4951e + 5 (5.63e + 4) −	1.9574e + 5 (3.03e + 4) −	4.9939e + 5 (2.99e + 4) −	4.7845e + 5 (5.00e + 4) −	**7.6400e + 2 (3.25e + 2)**
F4	1.3009e + 4 (3.35e + 3) −	3.6696e + 2 (6.27e + 1) −	2.5367e + 2 (4.51e + 1) =	**2.3425e + 2 (2.25e + 1)** +	2.7816e + 2 (4.23e + 1)
F5	1.0433e + 3 (7.01e + 1) −	8.3949e + 2 (1.11e + 2) −	1.0464e + 3 (2.97e + 1) −	9.1251e + 2 (2.58e + 1) −	**3.3379e + 2 (9.52e + 1)**
F6	7.9519e + 1 (6.76e + 0) −	6.8869e + 1 (4.92e + 0) −	2.1571e + 1 (3.13e + 0) −	6.7668e + 0 (1.09e + 0) −	**5.4069e + 0 (1.39e + 0)**
F7	2.6112e + 3 (2.30e + 2) −	3.5315e + 3 (3.80e + 2) −	1.1787e + 3 (2.10e + 1) −	1.0151e + 3 (1.91e + 1) −	**3.6440e + 2 (4.23e + 1)**
F8	1.1689e + 3 (6.74e + 1) −	9.0035e + 2 (9.48e + 1) −	1.0512e + 3 (1.98e + 1) −	9.1501e + 2 (2.00e + 1) −	**2.9957e + 2 (5.59e + 1)**
F9	3.7394e + 4 (4.99e + 3) −	5.1764e + 4 (7.53e + 3) −	2.5680e + 4 (5.35e + 3) −	**1.1677e + 3 (3.82e + 2)** =	2.1319e + 3 (1.77e + 3)
F10	2.1049e + 4 (1.21e + 3) −	1.5810e + 4 (1.46e + 3) −	3.0191e + 4 (5.79e + 2) −	3.0379e + 4 (4.74e + 2) −	**1.2142e + 4 (1.45e + 3)**
F11	9.0133e + 4 (2.11e + 4) −	2.3378e + 3 (1.37e + 3) −	2.4218e + 4 (4.21e + 3) −	1.8119e + 3 (1.98e + 2) −	**1.2458e + 3 (2.17e + 2)**
F12	1.6518e + 10 (6.46e + 9) −	1.3137e + 8 (6.51e + 7) −	7.7714e + 7 (4.04e + 7) −	**6.6567e + 6 (2.11e + 6)** +	3.0226e + 7 (1.18e + 7)
F13	6.1210e + 8 (7.54e + 8) −	5.5023e + 6 (1.12e + 7) −	**3.3181e + 3 (1.60e + 3)** +	1.0563e + 5 (2.77e + 4) −	7.5666e + 4 (2.89e + 4)
F14	2.1975e + 6 (1.62e + 6) −	6.2107e + 5 (2.89e + 5) −	3.0444e + 6 (6.82e + 5) −	**1.1729e + 3 (2.15e + 2)** +	1.0022e + 5 (4.11e + 4)
F15	4.0332e + 6 (8.08e + 6) −	1.0320e + 7 (4.54e + 7) −	**1.1419e + 3 (5.92e + 2)** +	3.1703e + 3 (4.34e + 2) +	6.3926e + 4 (2.59e + 4)
F16	7.4919e + 3 (1.14e + 3) −	5.5574e + 3 (8.76e + 2) −	8.6763e + 3 (3.65e + 2) −	8.6063e + 3 (3.89e + 2) −	**2.5769e + 3 (5.72e + 2)**
F17	6.3117e + 3 (1.37e + 3) −	4.4148e + 3 (5.05e + 2) −	5.4039e + 3 (2.34e + 2) −	5.3598e + 3 (2.89e + 2) −	**2.3680e + 3 (4.78e + 2)**
F18	3.0409e + 6 (2.95e + 6) −	9.7127e + 5 (3.99e + 5) −	5.3564e + 6 (1.03e + 6) −	6.5721e + 5 (1.49e + 5) −	**2.6727e + 5 (9.27e + 4)**
F19	1.7753e + 8 (2.47e + 8) −	5.2892e + 6 (1.80e + 6) −	**8.2265e + 2 (5.84e + 2)** +	6.5361e + 3 (1.57e + 3) +	2.9670e + 4 (8.17e + 3)
F20	4.0234e + 3 (6.06e + 2) −	3.4251e + 3 (4.31e + 2) −	5.0638e + 3 (3.24e + 2) −	5.3276e + 3 (2.41e + 2) −	**2.2435e + 3 (5.19e + 2)**
F21	1.7706e + 3 (1.64e + 2) −	1.4819e + 3 (1.45e + 2) −	1.2605e + 3 (3.84e + 1) −	1.1385e + 3 (2.72e + 1) −	**5.0094e + 2 (4.77e + 1)**
F22	2.2722e + 4 (1.52e + 3) −	1.8039e + 4 (1.47e + 3) −	3.0257e + 4 (4.09e + 3) −	3.1246e + 4 (4.75e + 2) −	**1.4597e + 4 (1.33e + 3)**
F23	3.0745e + 3 (3.06e + 2) −	2.6570e + 3 (2.00e + 2) −	1.5333e + 3 (2.67e + 1) −	1.4883e + 3 (3.12e + 1) −	**8.5151e + 2 (6.15e + 1)**
F24	4.5409e + 3 (5.72e + 2) −	4.7312e + 3 (3.77e + 2) −	1.8708e + 3 (2.36e + 1) −	1.7964e + 3 (2.80e + 1) −	**1.2095e + 3 (5.31e + 1)**
F25	7.5473e + 3 (1.11e + 3) −	9.3390e + 2 (9.36e + 1) −	8.3379e + 2 (3.57e + 1) −	**7.1820e + 2 (4.64e + 1)** +	7.9853e + 2 (4.71e + 1)
F26	2.8916e + 4 (2.87e + 3) −	3.0295e + 4 (3.42e + 3) −	9.8554e + 3 (5.17e + 3) =	1.1477e + 4 (1.29e + 3) −	**6.6992e + 3 (1.54e + 3)**
F27	2.6787e + 3 (6.16e + 2) −	3.6094e + 3 (8.39e + 2) −	1.0786e + 3 (6.38e + 1) −	7.6726e + 2 (5.37e + 1) =	**7.4665e + 2 (4.79e + 1)**
F28	1.1085e + 4 (1.97e + 3) −	7.0589e + 2 (6.23e + 1) −	6.6229e + 2 (2.34e + 1) −	6.8811e + 2 (1.59e + 2) =	**6.1186e + 2 (2.96e + 1)**
F29	1.1226e + 4 (2.48e + 3) −	7.0144e + 3 (7.70e + 2) −	6.5877e + 3 (2.17e + 2) −	6.7393e + 3 (3.41e + 2) −	**2.8097e + 3 (4.69e + 2)**
F30	1.1043e + 9 (6.56e + 8) −	2.9254e + 7 (1.27e + 7) −	2.5165e + 5 (3.14e + 5) +	**4.8213e + 4 (1.36e + 4)** +	5.6326e + 5 (2.49e + 5)
+/-/=	0/30/0	0/30/0	4/23/3	7/20/3	

Table II. *Detailed RMSE results.*

	Algorithm	Minimum	Maximum	Mean	Variance	Wilcoxon rank-sum test
Train set	ELM	1.652e-01	2.764e-01	2.020e-01	2.054e-02	+
	ELM-PSO	1.140e-01	1.649e-01	1.345e-01	1.097e-02	+
	ELM-DE	8.683e-02	1.579e-01	1.162e-01	1.855e-02	+
	ELM-OFA	1.337e-01	1.747e-01	1.510e-01	1.069e-02	+
	ELM-IKPSO	**8.203e-02**	**1.315e-01**	**9.892e-02**	**9.306e-03**	
Test set	ELM	3.446e-01	5.800e-01	4.407e-01	4.968e-02	+
	ELM-PSO	2.969e-01	5.883e-01	4.054e-01	6.265e-02	+
	ELM-DE	**2.525e-01**	5.204e-01	3.691e-01	5.915e-02	+
	ELM-OFA	3.248e-01	6.383e-01	4.534e-01	6.136e-02	+
	ELM-IKPSO	2.587e-01	**4.632e-01**	**3.319e-01**	**4.607e-02**	

can be seen that although ELM-DE is similar to ELM-IKPSO in the minimum value, it is worse than ELM-IKPSO in both mean and variance, which proves that ELM-IKPSO has better robustness and generalization ability. Therefore, ELM-IKPSO has good generalization performance, and the model built by ELM-IKPSO is efficient. Figure 3(b) plots the boxplot of the five algorithms on the test set. From the information in the box diagram, ELM-IKPSO achieves better statistical results. In addition, the four optimized models are more obvious than the original ELM algorithm, which proves the importance of the optimized ELM.

Figure 3(c) shows the iterative convergence diagram of different evolutionary algorithms in the optimization process, where the horizontal axis is the number of fitness function evaluations, and the vertical axis is the mean of the validation set results. It can be seen from the convergence curve that the OFA converges too fast, and it is easy to converge to the local optimum in the CFBB modeling problem. DE and PSO are better than OFA. After using the Kalman filter principle to correct PSO, the extra number of fitness evaluations will be consumed in each iteration to find more potential regions and prevent convergence to the local optimum. From the perspective of the search process, the IKPSO algorithm is superior to the compared algorithms. Figure 3(d) and (e) shows the fitting effect on the test set. Figure 3(d) shows the thermal efficiency fitting effect, and Fig. 3(e) shows the NOx fitting effect. The blue star marks the actual data. In addition, it can be seen from the two simulation graphs that the output values of ELM-IKPSO are close to the actual data. The model built by ELM-IKPSO is efficient.

Finally, the established model is multi-objective optimized to increase the boiler combustion efficiency and reduce the NOx emission concentration. These two objectives are conflicting, and the improvement of thermal efficiency is accompanied by an increase in NOx emission concentration. Figure 3(f) is the optimized 22 groups of mutually non-dominated solutions. If NOx concentration needs to be controlled below 80 $mg^{*}Nm^{-3}$, decision manager can choose from the solution sets.

5. Conclusions

This paper proposed an extreme learning model based on IKPSO to establish the combustion model of the CFBB. Aiming at some shortcomings of the existing KPSOs, the update method of Kalman gain was improved to make it more suitable for the evolution process of the population. At the same time, in the process of searching, the population was adaptively grouped hierarchically, and the individuals in the convergent state were corrected. Compared with the other three algorithms, the model optimized by the proposed algorithm showed better performance and generalization ability. Finally, multi-objective optimization was carried out on the model, and a set of widely distributed non-dominated solutions was obtained. Decision managers could choose the appropriate operation scheme according to the solutions.

In future work, the proposed model will combine online learning methods and selectively update the model when new data are obtained. We will also consider optimizing the model using time series forecasting methods. In addition, the algorithm will be further improved to obtain more combustion optimization solutions.

Author contributions. Ke Chen and Hao Guo conceived and designed the study. Jing Liang, Kunjie Yu, and Caitong Yue conducted data gathering. Xia Li participated in the design of the study and performed the statistical analysis. Ke Chen and Hao Guo wrote the article. All authors read and approved the final manuscript.

Financial support. This work was supported by the National Natural Science Foundation of China (61806179, 61876169, 61922072, 61976237, 61673404, 62106230, 62006069, and 62206255), China Postdoctoral Science Foundation (2021T140616, 2021M692920, 2022M712878 and 2022TQ0298), Key R&D and Promotion Projects in Henan Province (192102210098, 212102210510), and Henan Postdoctoral Foundation (202003019).

Conflicts of interest. The authors declare no conflicts of interest exist.

References

[1] X. S. Gao, D. G. Xu, Y. Wang, H. H. Pan and W. M. Shen, “Multifunctional robot to maintain boiler water-cooling tubes,” *Robotica* **27**(6), 941–948 (2009).

[2] H. P. Jagtap, A. K. Bewoor, F. Pathan and R. Kumar, “Application of Particle Swarm Optimization Method to Availability Optimization of Thermal Power Plants,” **In:** *Nature-Inspired Optimization in Advanced Manufacturing Processes and Systems* (CRC Press, 2020) pp. 97–112.

[3] S. Santra, R. Sakthivel, Y. Shi and K. Mathiyalagan, “Dissipative sampled-data controller design for singular networked cascade control systems,” *J. Frankl. Inst.* **353**(14), 3386–3406 (2016).

[4] D. Adams, D. H. Oh, D. W. Kim, C. H. Lee and M. Oh, “Prediction of SOx–NOx emission from a coal-fired CFB power plant with machine learning: Plant data learned by deep neural network and least square support vector machine,” *J. Clean. Prod.* **270**, 122310 (2020).

[5] K. K. Pandey and D. R. Parhi, “Trajectory planning and the target search by the mobile robot in an environment using a behavior-based neural network approach,” *Robotica* **38**(9), 1627–1641 (2020).

[6] B. A. Garro, K. Rodríguez and R. A. Vázquez, “Classification of dna microarrays using artificial neural networks and abc algorithm,” *Appl. Soft Comput.* **38**(July), 548–560 (2016).

[7] S. R. Shahamiri and S. S. B. Salim, “Real-time frequency-based noise-robust automatic speech recognition using multi-nets artificial neural networks: a multi-views multi-learners approach,” *Neurocomputing* **129**, 199–207 (2014).

[8] K. Bouhoune, K. Yazid, M. S. Boucherit and A. Chériti, “Hybrid control of the three phase induction machine using artificial neural networks and fuzzy logic,” *Appl. Soft. Comput.* **55**(2), 289–301 (2017).

[9] N. Kim, B. Jeong and K. Park, “A novel methodology to explore the periodic gait of a biped walker under uncertainty using a machine learning algorithm,” *Robotica* **40**(1), 120–135 (2022).

[10] G. B. Huang, Q. Y. Zhu and C. K. Siew, “Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks,” **In:** *IEEE International Joint Conference on Neural Networks 2004*, vol. 2 (2004) pp. 985–990.

[11] N. J. Guliyev and V. E. Ismailov, “On the approximation by single hidden layer feedforward neural networks with fixed weights,” *Neural Netw.* **98**(4), 296–304 (2018).

[12] G. B. Huang, Q. Y. Zhu and C. K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing* **70**(1–3), 489–501 (2006).

[13] Q. Y. Zhu, A. K. Qin, P. N. Suganthan and G. B. Huang, “Evolutionary extreme learning machine,” *Patt. Recognit.* **38**(10), 1759–1763 (2005).

[14] J. W. Cao, Z. P. Lin and G. B. Huang, “Self-adaptive evolutionary extreme learning machine,” *Neural Process. Lett.* **36**(3), 285–305 (2012).

[15] T. Matias, F. Souza, R. Araújo and C. H. Antunes, “Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine,” *Neurocomputing* **129**(16), 428–436 (2014).

[16] F. Han, H. F. Yao and Q. H. Ling, “An improved evolutionary extreme learning machine based on particle swarm optimization,” *Neurocomputing* **116**(1), 87–93 (2013).

[17] Y. F. Xu, G. C. Chen and J. S. Yu, “The Kalman Particle Swarm Optimization Algorithm and Its Application in Soft-Sensor of Acrylonitrile Yield,” **In:** *International Conference on Natural Computation (2006)*, pp. 176–179.

[18] V. Pano and P. R. Ouyang, “Gain tuning of position domain pid control using particle swarm optimization,” *Robotica* **34**(6), 1351–1366 (2016).

[19] Y. L. Wu, G. Liu, X. P. Guo, Y. H. Shi and L. X. Xie, “A self-adaptive chaos and kalman filter-based particle swarm optimization for economic dispatch problem,” *Soft Comput.* **21**(12), 3353–3365 (2017).

[20] P. L. Gilabert, R. N. Braithwaite and G. Montoro, “Beyond the moore-penrose inverse: strategies for the estimation of digital predistortion linearization parameters,” *IEEE Microw. Mag.* **21**(12), 34–46 (2020).

[21] K. Chen, Q. Lv, Y. Lu and Y. Dou, "Robust regularized extreme learning machine for regression using iteratively reweighted least squares," *Neurocomputing* **230**(1), 345–358 (2017).

[22] Y. Mo, Z. Song, H. Li and Z. Jiang, "A hierarchical safety control strategy for exoskeleton robot based on maximum correntropy kalman filter and bounding box," *Robotica* **37**(12), 2165–2175 (2019).

[23] P. Peng, C. Chen and Y. Yang, "Particle swarm optimization based on hybrid Kalman filter and particle filter," *J. Shanghai Jiaotong Univ. (Sci.)* **25**(6), 681–688 (2020).

[24] S. Sengupta, S. Basak and R. A. Peters, "Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives," *Mach. Learn. Knowl. Extract*. **1**(1), 157–191 (2018).

[25] J. Liang, H. Guo, K. J. Yu, B. Y. Qu, C. T. Yue and K. J. Qiao, "An Improved Composite Differential Evolutionary Algorithm with Self-Adaptive Mutation Strategy for Identifying Photovoltaic Model Parameters," **In:** *5th Asian Conference on Artificial Intelligence Technology (ACAIT)* (2021) pp. 591–599.

[26] G. Y. Zhu and W. B. Zhang, "Optimal foraging algorithm for global optimization," *Appl. Soft Comput.* **51**(5), 294–313 (2017).

[27] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu and P. N. Suganthan, Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization, Technical report, Zhengzhou University (2016).

[28] J. Derrac, S. García, D. Molina and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Lect. Notes Comput. Sci.* **1**(1), 3–18 (2011).

[29] J. Liang, L. Y. Zhang, K. J. Yu, B. Y. Qu, C. T. Yue and K. J. Qiao, "A Differential Evolution Based Self-Adaptive Multi-Task Evolutionary Algorithm," **In:** *5th Asian Conference on Artificial Intelligence Technology (ACAIT)* (2021) pp. 150–156.

Cite this article: J. Liang, H. Guo, K. Chen, K. Yu, C. Yue and X. Li (2023). "An improved Kalman particle swarm optimization for modeling and optimizing of boiler combustion characteristics", Robotica **41**, 1087–1097. https://doi.org/10.1017/S026357472200145X