IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2020, accepted July 11, 2020, date of publication July 14, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009248

A Collaborative Compound Neural Network Model for Soil Heavy Metal Content Prediction

WENQI CAO AND CONG ZHANG
School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China

Corresponding author: Cong Zhang (hb_wh_zc@163.com)

This work was supported in part by the Major Technical Innovation Projects of Hubei Province under Grant 2018ABA099, in part by the National Science Fund for Youth of Hubei Province of China under Grant 2018CFB408, in part by the Natural Science Foundation of Hubei Province of China under Grant 2015CFA061, and in part by the National Nature Science Foundation of China under Grant 61272278.

ABSTRACT The prediction of soil heavy metal content is an important part of the management of soil heavy metal pollution, but it is often ignored. At present, there are few studies on the prediction of soil heavy metal content, and it is an urgent problem to choose an efficient method for soil heavy metal content prediction. In this paper, a collaborative compound neural network model (CCNN) was put forward to predict the soil heavy metal content, this model uses wavelet neural network (WNN) as the basic prediction model, and at the same time proposes a parallel bird swarm algorithm (PBSA) to solve the parameter optimization problem of WNN, based on the bird swarm algorithm (BSA), the PBSA not only increases the gathering behavior of individual, but also adopts sine transformation based on fitness difference ratio to carry out the following behavior of beggars to improve the global optimization ability, besides that, the acceptance criterion is used to compare the fitness of individuals after updating to avoid falling into a local optimum. Soil heavy metal content data from Yinchuan city of Ningxia and six new urban areas in Wuhan, China are used to make prediction experiments respectively, through compare with support vector machine (SVM), radial basis function neural network (RBFNN), WNN and bird swarm algorithm optimizes wavelet neural network (BSA-WNN), the experimental results demonstrate that the predicted value of the CCNN is closer to the actual value and has better prediction performance.

INDEX TERMS Soil heavy metal content prediction, collaborative compound neural network model, parallel bird swarm algorithm, wavelet neural network.

I. INTRODUCTION

The heavy metals in the soil are usually not broken down by soil microorganisms, its long-term accumulation will pollute the environment and also enter the human body along with the food chain, which directly endangering human health [1]. Especially for the past several years, with the continuous development of industry, the heavy metal pollutants discharged to soil are also increasing, how to reduce the content of heavy metal pollutants in soil has become one of the problems that need to be urgently solved [2]. In the process of soil heavy metal pollution control, due to some conditions, it is difficult to grasp the detailed soil heavy metal content data, so predicting soil heavy metal content has become an effective way to solve this problem [3].

There are many methods for data prediction, the traditional methods include multiple linear regression (MLR) [4], support vector machine (SVM) [5], etc., they have been applied to data prediction in many fields. In recent years, as researchers have more and more research on artificial neural network, it has been widely used in data prediction, there are many types of neural networks available for it, for example, back propagation neural network (BPNN) [6], fuzzy neural network (FNN) [7], radial basis function neural network (RBFNN) [8] and general regression neural network (GRNN) [9], etc., compared with traditional prediction methods, artificial neural networks have been proven to have higher prediction accuracy on nonlinear problems [10], [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Tu Ngoc Nguyen.

Wavelet neural network (WNN) is an artificial neural network proposed on the basis of breakthroughs in wavelet analysis and research, it combines wavelet analysis with artificial neural network [12], compared with traditional neural network models like BPNN, it has the advantages of fast convergence speed and strong nonlinear approximation ability [13]. WNN has been applied to data prediction in various industries, for example, Chen *et al.* applied the WNN to short-term power load forecasting and the numerical testing showed that this method provides accurate predictions [14]. Wen et al employed WNN to propose a prediction model to estimate the gross calorific value of coals, the experimental results proved that this model has a better prediction performance [15]. However, during the training process, the parameters of WNN are difficult to determine, which often results in the unstable prediction performance on it [16].

Researchers usually use intelligent optimization algorithms to optimize the parameters of WNN to improve its prediction performance, Xiao *et al.* utilized a new improved cuckoo search algorithm to optimize the initial weights and the parameters of dilation and translation in WNN, the error indicator is significantly lower than before in the results [17], Wang *et al.* adopted WNN with an improved fruit fly optimization algorithm (IFOA) for the melt index prediction in the industrial propylene polymerization, the WNN–IFOA model can obtain the least predicting errors compared with other existing models [18]. In the last few years, there have been many intelligent optimization algorithms proposed, such as particle swarm optimization (PSO) [19], differential evolution (DE) [20], firefly algorithm (FA) [21], covariance matrix adaptation evolution strategy (CMA-ES) [22], coyote optimization algorithm (COA) [23], bird swarm algorithm (BSA) [24], etc., the BSA was proposed by Meng et al in 2016 and compared with other swarm intelligence algorithm like PSO and DE, it can have good diversity and efficiently avoid prematurity, when solving the objective optimization problem, its convergence speed and accuracy is better than PSO and DE, too. But the same as other meta-heuristic algorithms, with the complexity of the problem to be optimized increases, it is easy to fall into the local optimal.

The purpose of this article is to explore an efficient method for the prediction of soil heavy metal content. Among the data prediction methods, WNN has been proved to have higher prediction accuracy than traditional methods like MLR and neural networks like BPNN, but it is rarely applied to the prediction of soil heavy metal content, so WNN is used as the basic prediction model. However, in view of the difficulty of determining the parameters in WNN, the optimization algorithm is used to optimize the parameters in it, compared with the existing optimization algorithms such as PSO and DE, the effectiveness of BSA in handling optimization problems has been proved, but it is difficult to balance its exploration and exploitation. Therefore, this paper first makes improvements to BSA and proposes a parallel bird swarm algorithm (PBSA) on the basis of the BSA, the gathering behavior is added before the bird flying and foraging behavior, that is to divide the population into multiple small groups and learn the information from the each groups to expand the global search range, at the same time, the sine transformation based on the fitness difference ratio is increased to the position update formula of the begging behavior, and after each position update, use the acceptance criterion to judge whether to accept this update, so as to improve the effectiveness of the internal information exchange in the population. Then the PBSA is applied to the optimization of the dilation and translation parameters and the weights in WNN, which is combined into the collaborative compound neural network model (CCNN). The simulation experiment results demonstrate that this model can achieve better prediction performance on the prediction of soil heavy metal content.

The rest of this paper is laid out as follows: A brief introduction of WNN and BSA are describe in section II. Section III and IV present the details of PBSA and CCNN respectively. The experimental results and analysis of PBSA on several benchmark functions and CCNN in soil heavy metal content prediction are shown in section V. Section VI gives the conclusion.

II. BASIC THEORY

A. WAVELET NEURAL NETWORK

The wavelet neural network (WNN) was first introduced by Zhang and Benveniste [25]. It has the same structure as the artificial neural network, but the nonlinear activation function of its neurons is the wavelet basis function, this feature makes it have strong self-learning ability and good local microscopic characteristics of wavelet transform, so it has better nonlinear fitting ability and fast convergence speed [26]–[28]. The structure of WNN is shown in Fig. 1.

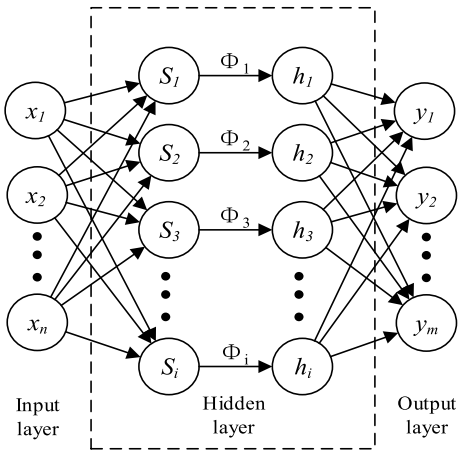

FIGURE 1. The structure of WNN.

Assume that the number of nodes in the input layer, hidden layer, and output layer are n, i, and m, respectively, the input of the data at the *j-th* node of the hidden layer is as follows:

$$S_j = \sum_{k=1}^{n} \omega_{kj} x_k, \quad j = 1, 2, , \ldots, i \tag{1}$$

where x_k is the *k-th* sample input data and ω_{kj} is the weight of hidden layer.

The above input performs wavelet-based telescopic translation changes in the hidden layer, and the calculation formulas are as follows:

$$\phi(x) = \cos(1.75x) * e^{-x^2/2} \tag{2}$$

$$h_j = \phi[(S_j - b_j)/a_j] \tag{3}$$

where $\phi(x)$ is wavelet basis function, b_j is the translation factor and a_j is the dilation factor, h_j is the output of *j-th* node of the hidden layer.

The final output of the data at the *t-th* node of the output layer is given by:

$$y_t = \varphi\left(\sum_{j=1}^{i} \omega_{jt} h_j\right), \quad t = 1, 2, \ldots, m \tag{4}$$

where ω_{jt} is weight of output layer, φ is the activation function.

It can be seen from the above calculation process that there are four main parameters in WNN: ω_{kj}, b_j, a_j, ω_{jt}, and the setting of their values directly affects the performance of the network, it is usually hard to set these parameters to more appropriate values in experiments, therefore, finding a suitable intelligent optimization algorithm to optimize these parameters is often of critical importance.

B. BIRD SWARM ALGORITHM

Bird swarm algorithm (BSA) is a swarm intelligence algorithm, which imitates the behavior of bird swarm like foraging behavior, vigilance behavior and flight behavior in the nature to realize the optimization process of the target problem [29], [30].

In the procedure of BSA, the individual uses flight interval FQ to determine whether to conduct flight behavior or forage behavior, when is flight behavior, the individual needs to be a producer or scrounger to update their position. The formula for updating the position of producer and scrounger are as follows:

$$x_i^{t+1} = x_i^t + x_i^t * randn(0, 1) \tag{5}$$

$$x_i^{t+1} = x_i^t + \left(x_k^t - x_i^t\right) * FL * rand(0, 1) \tag{6}$$

where x_i^{t+1} is the position of the individual i after the *t-th* update, x_i^t is the position of the individual i before the *t-th* update, $randn(0, 1)$ denotes Gaussian distributed random number with mean 0 and standard deviation 1, x_k^t is the position of the individual $k(k \neq i)$ before the *t-th* update, FL indicates that the scrounger follows the producer's search for food and the range is [0, 2], $rand(0, 1)$ denotes independent uniformly distributed numbers in (0, 1).

When the individual is forage behavior, it chooses to forage for food or keep vigilance according by the preset condition. The updating formula of the individual's position when foraging for food is as follows:

$$x_i^{t+1} = x_i^t + \left(p_i - x_i^t\right) * C * rand(0, 1) + \left(g_t - x_i^t\right) * S * rand(0, 1) \tag{7}$$

where p_i is the best previous position of the *i-th* individual and g_t is the best previous position shared by all individuals, C and S are cognitive coefficients and social accelerated coefficients respectively.

When the individual keeps vigilance, it moves to the center of the population, and the position update formula is as follows:

$$x_i^{t+1} = x_i^t + A1 * \left(mean - x_i^t\right) * rand(0, 1) + A2 * \left(p_k - x_i^t\right) * rand(-1, 1) \tag{8}$$

$$A1 = a_1 * \exp\left(-\frac{pFit_i}{sumFit + \epsilon} * N\right) \tag{9}$$

$$A2 = a_2 * \exp\left(-\frac{pFit_i - pFit_k}{|pFit_i - pFit_k| + \epsilon} * N * \frac{pFit_k}{sumFit + \epsilon}\right) \tag{10}$$

where $mean$ represents the average position of the entire population, $rand(-1, 1)$ is independent uniformly distributed numbers in $(-1, 1)$, p_k is the best previous position of the *k-th* individual, a_1 and a_2 are two positive constants in [0, 2], $pFit_i$ denotes the *i-th* individual's best fitness value, $sumFit$ is the sum of the population's best fitness value, ϵ is used to avoid zero-division error and N is the total number of individuals, $A1$ and $A2$ respectively represent the influence degree of population's average position and *k-th* individual's position on *i-th* individual when *i-th* individual moves.

Through the position update of the above four behaviors, individuals in the population search for the best position within the target range to achieve the process of optimizing. Although the efficiency of BSA in dealing with optimization problems has been proven, but like most meta-heuristic algorithms, as the complexity of the problem increases, the convergence accuracy and speed of the algorithm decreases, finding a better way to balance exploration and exploitation is the direction of improving the algorithm at present [21], [31].

III. PARALLEL BIRD SWARM ALGORITHM

In the bird swarm algorithm (BSA), due to the individual's position movement is mainly toward the best position in the population, it is easy to cause the diversity of the population to decrease and fall into the local optimum. In addition, when the individual is moving as a scrounger, the FL value is usually a preset fixed value, which causes the scrounger to learn too much or not enough when learning the position of other individuals easily, too much learning may lead to the over-moving and miss a better position, learning not enough may induce the convergence speed of the algorithm to decrease. In response to these two problems, the parallel bird swarm algorithm (PBSA) proposes the gathering behavior, sine transformation position update method based on fitness difference ratio, and the acceptance criterion on the basis of BSA for improvement.

A. GATHERING BEHAVIOR

In the natural world, when birds are flying or foraging, individuals in the swarm can be divided into multiple small

groups according to the distance between them. The behavior of individuals is easily affected by small groups, and the individuals of each small group are composed of random, when flying or foraging again, the position of each individual in the swarm changed, and the individual composition of the small group also changed. Inspired by this phenomenon, PBSA increased the gathering behavior for the individuals in bird swarm.

The gathering behavior of the individuals occurs before each flying and foraging behavior. The implementation of this steps are as follows:

1) Randomly select several individuals as the central point of the small group, the number of points is the number of small groups;

2) Each individual calculates the spatial distance from different center points, and selects the closest group to join in;

3) In the subsequent position movement, each individual performs position learning on other individual in the small group. For example, the position update formula during foraging behavior is modified to:

$$x_i^{t+1} = x_i^t + \left(p_i - x_i^t\right) * C * rand\ (0, 1) + \left(g_s - x_i^t\right) * S * rand\ (0, 1) \quad (11)$$

where g_s is the best previous position of the small group, which can be obtained by comparing the best previous position of each individual in the small group, and the modification of the position update formula of other behaviors are the same.

4) After updating the position of individuals, the individual composition of each small group is re-organized, that is, the individual is randomly determined as the central point again, and the new small group is constituted according to the spatial distance.

B. SINE TRANSFORMATION POSITION UPDATE METHOD BASED ON FITNESS DIFFERENCE RATIO

When the individual moves the position as a scrounger, *FL* means scrounger follows the producer to find food, the larger the value, the more the position moves and vice versa. In the BSA, *FL* is set to a fixed value, and in the actual learning process, the fitness of each learned individual is different, and the amount of individual learning should also change accordingly, so it is not reasonable to set a fixed value of *FL*. Sine transformation position update method based on fitness difference ratio takes sine transformation as the variation method for *FL*, then, separately calculates the fitness difference between the learning individual and the learned individual, and the fitness difference between the best individual and the worst individual, the ratio of these two is used as the independent variable of *FL*. The *FL*'s calculation formula is set as follows:

$$\mathrm{FL}_i = (FL_{max} - FL_{min}) + FL_{min} * \sin\left(\frac{\pi\ |f_k - f_i|}{2\ (f_{max} - f_{min})}\right) \quad (12)$$

where FL_{max} and FL_{min} are constant number between 0 and 2, f_{max} and f_{min} are the maximum fitness value and the minimum fitness value, respectively, f_k is the fitness value of the learned individual and f_i is the fitness value of the individual i.

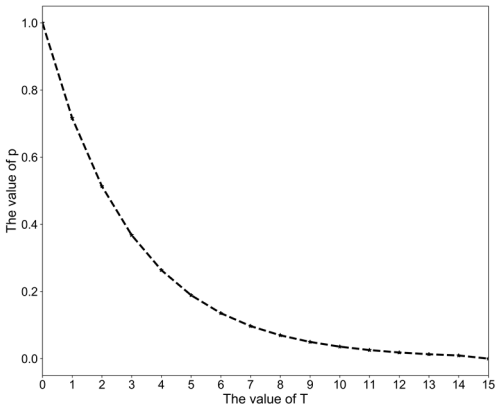

FIGURE 2. The change process of *FL* with fitness difference ratio.

Suppose that the value of FL_{max} is 2 and the value of FL_{min} is 1, the change process of *FL* with the fitness difference ratio is shown in Fig. 2. At this time, there are two cases: one is that the fitness value of the learning individual is lower than that of the learned individual, then *FL* increases as the fitness value of the learning individual increases, and the amount of the individual moving to a better position also increases, by this way to a better position; the other is that the fitness value of the learned individual is higher than that of the learned individual, then *FL* increases with the decrease of the fitness value of the learned individual, and the amount of individual exploration into the surrounding area also increases, so as to find more better position to avoid falling into a local optimum.

C. ACCEPTANCE CRITERION

After each position update, the individual may have a lower fitness value than before, at this point, whether or not to accept the update is directly affect the convergence speed of the algorithm. In this method, the fitness value comparison link is added after each individual position update, and an acceptance criterion is proposed as the criterion for the individual to accept the location update, in this criterion, the probability p is used to determine whether the individual accepts the update process. The specific calculation formula of p is as follows:

$$p = \begin{cases} \exp\left(-\frac{T}{3}\right), & f_{new} - f_{old} \le 0; \\ 1, & f_{new} - f_{old} > 0 \end{cases} \quad (13)$$

where f_{new} is the fitness of individual after position update, f_{old} is the fitness of individual before position update, T is the times of sustained decrease in fitness value and its initial value is 0, when the fitness value decreases after the position

update continuously, use the following formula to update T:

$$T = T + 1 \tag{14}$$

Fig. 3 shows the curve of p changing with T, When the position is moved multiple times in a row with the fitness value decreases or remains unchanged, the value of T increase, at the beginning, the number of consecutive times is small, it does not mean that the position has fallen into the local optimum and there is still a large probability to accept the position update, but as the number of consecutive times increases, the possibility of falling into local optimality also increases, so the possibility of accepting position update is decreases to zero. When the individual does not accept the position update, the individual will return to the position before the update and perform the position update under different behaviors again, unless there is a situation where the fitness value increases after the position update, the individual will directly accept the updated position, and the T value will be initialized to 0.

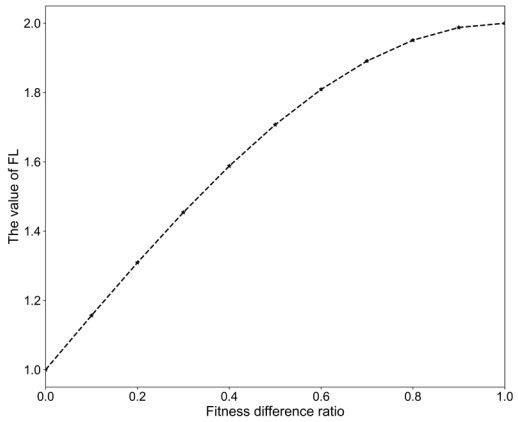

FIGURE 3. The probability to accept the position update.

IV. COLLABORATIVE COMPOUND NEURAL NETWORK MODEL

The collaborative compound neural network model (CCNN) mainly sets the weights of the WNN hidden layer and output layer, translation factor and dilation factor as the individual position of the population in PBSA, then use the training data through the WNN under different parameters to calculate the mean square error between the output value and the actual value as the fitness function, finally, the best position found is used as the initial WNN parameter value for network training. The specific steps can be summarized as the pseudo code shown in Table 1 and the flowchart of CCNN is shown in Fig. 4.

According to Fig. 4 and Table 1, the procedure of PBSA optimizes WNN parameters and trains WNN model need the highest computational overhead of executing CCNN, the computational complexity of other steps is relatively simple and can be ignored. When using PBSA optimizes WNN parameters, the three steps of generating individuals, updating individual positions, and calculating fitness are relatively complex, assuming that the population is N and the number of iterations is M, then the computational complexity of the three programs is the same, namely $O(MN)$, so the computational complexity of PBSA is $O(MN)$. When training WNN, the main steps are forward calculation and error back propagation to adjust parameters, assuming that the number of iterations is M, then they have the same computational complexity of $O(M)$. To summarise, the overall computational complexity of CCNN is $O(MN)$.

V. EXPERIMENTS AND ANALYSIS

A. VALIDATION OF PARALLEL BIRD SWARM ALGORITHM

In order to verify the effectiveness of the proposed PBSA, six benchmark problems are selected to compare the four algorithms of PBSA, BSA, FA, CMA-ES and COA, the theoretical optimal solutions of the six benchmark problems are 0, of which the first three are unimodal and the last three are multimodal. Their formulas are as follows:

1) Schwefels P2.22 (domain : $[-10, 10]$, $d= 20$)

$$f_1(x) = \sum_{j=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \tag{15}$$

2) Quadric (domain : $[-100, 100]$, $d= 20$)

$$f_2(x) = \sum_{j=1}^{n} (\sum_{i=1}^{j} x_i^2) \tag{16}$$

3) Rosenbrock's (domain : $[-10, 10]$, $d= 20$)

$$f_3(x) = \sum_{i=1}^{n-1} [100 \left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2] \tag{17}$$

4) Rastrigin (domain : $[-5.12, 5.12]$, $d= 20$)

$$f_4(x) = \sum_{j=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10] \tag{18}$$

5) Ackley (domain : $[-32, 32]$, $d= 20$)

$$f_5(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) \tag{19}$$

6) Girewank (domain : $[-600, 600]$, $d= 20$)

$$f_6(x) = \sum_{j=1}^{n} \frac{x_i^2}{4000} + \prod_{i=1}^{n} \left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{20}$$

All the experiments using PyCharm professional 2018 are performed on a computer with 1.8 GHz dual-core processor and 8.0 GB of RAM in macOS Catalina operating system. The parameter settings for PBSA, BSA, FA, CMA-ES and COA are shown in Table 2, and these five algorithms iterate 100 times under the six benchmark problems respectively.

TABLE 1. Pseudo code of CCNN.

Collaborative compound neural network model (CCNN)
Input: N: the number of individuals; g: the number of small group; FQ: the flight interval
P: the probability of foraging for food; $C, S, a_1, a_2, FL_{max}, FL_{min}$: six constant parameters
n, i, m: the number of nodes in the input layer, hidden layer and output layer
Initialize the parameters in WNN as individual of population in PBSA; $t = 0$
Calculate the fitness of each individual, renew the best previous position of the each individual and population
While (stop condition in not reached) do
Divide the population into multiple small groups through gathering behavior
If $(t\%FQ \neq 0)$
If $rand(0,1) < P$
Birds forage for food
Else
Bids keep vigilance
End if
Else
If individual is producer
Producing
Else
Scrounging
End if
End if
Calculate the new fitness, use acceptance criterion to determine whether to accept position update
Renew the best previous position of the each individual and population; $t = t + 1$
End while; Set the best previous position of population as the initial parameters of WNN
While (stop condition in not reached) do
WNN forward calculation
Error back propagation to adjust parameters
End while
Output: the model after trained

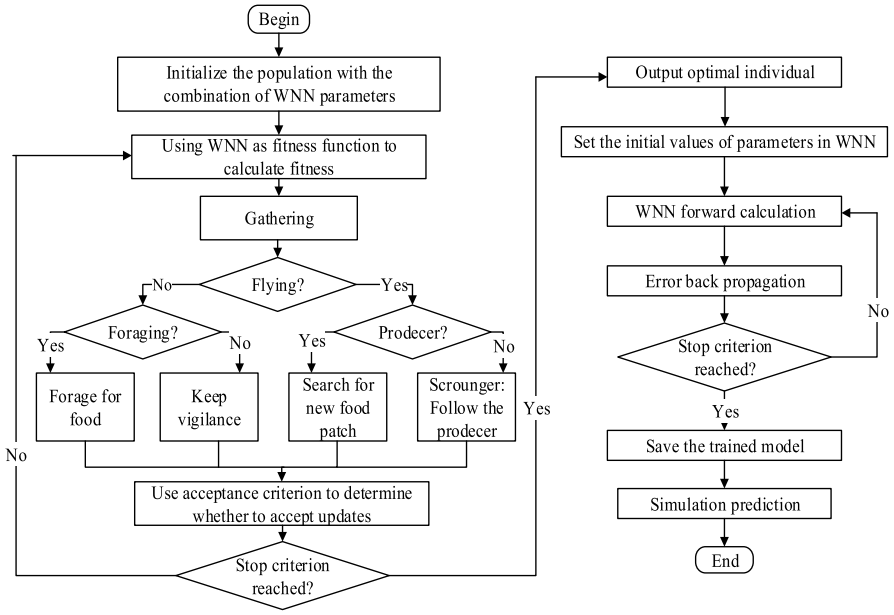

FIGURE 4. The flowchart of CCNN.

The final optimization results of the five algorithms under the six benchmark problems are shown in Table 3. From Table 3, we can see that whether it is unimodal or multimodal, the optimization results of BSA and PBSA are much closer to 0 than FA, CMA-ES and COA, among the optimization results for f_4 and f_6, both BSA and PBSA reached the theoretical optimal solution, a possible reason is that, compared with the other three algorithms, BSA and PBSA have

TABLE 2. Parameter settings.

Algorithm	Parameters
CMA-ES	$\lambda = 30, \sigma^{(0)} = 0.5$
COA	$N_p = 20, N_c = 5$
FA	$m = 30, \alpha = 0.2, \gamma = 1, \beta_0 = 1$
BSA	$m = 30, C = S = 1.5, a_1 = a_2 = 1$ $FQ = 3, FL \in [0.5,0.9], P \in [0.8,1]$
PBSA	$m = 30, C = S = 1.5, a_1 = a_2 = 1,$ $FQ = 3, FL_{max}$ =0.9,FL_{min} = 0.5, $P \in [0.8,1]$

carried out the learning of the internal position information from the population while randomly moving to the external position of the population, which can help it to find the existence of a better position faster. Apart from this, when comparing PBSA and BSA, it can be seen that although both of them have reached the theoretical optimal solution on f_4 and f_6, the mean value and the standard deviation of PBSA are best than BSA, PBSA gets the best results on 2 out of 3 unimodal benchmark problems and gets the best results on all multimodal benchmark problems, this may be due to the fact that PBSA added acceptance criterion on the basis of BSA, which avoids the emergence of lower fitness values after the location update. It can be known from the above results that PBSA has higher convergence accuracy when comparing the final optimization results of the five algorithms.

In order to analyze the significance of the results obtained, the non-parametric statistical significance tests of Frideman and Nemenyi post-hoc have been used, after calculating the results in Table 3, it can be got that the value of τ_F is 34.13 and the value of p_{value} is 1.94e-04, which means that there is significant difference in all cases for an α level of 0.05, then use the Nemenyi Post-hoc to calculate the critical difference domain, the performance comparison results of the five algorithms are displayed using the Frideman test chart in Fig. 5. As shown in Fig. 5, the horizontal line represents the critical difference domain of each algorithm, and the value corresponding to the point in the middle of the horizontal line represents the average ordinal value of each algorithm, the horizontal line of PBSA does not overlap with the horizontal line of COA and FA and the average ordinal value of PBSA is smaller than COA and FA, indicating that these three algorithms are significantly different and the performance of PBSA is better than COA and FA. The horizontal line of PBSA overlaps with the horizontal line of BSA and CMA-ES, indicating that the performance of these three is relatively close, so it needs to be judged by the average ordinal value, when comparing the average ordinal value, it can be seen that the average ordinal value of PBSA is smaller than other two, which means the performance of PBSA is better than the other two.

Under the six benchmark problems, the change of the best fitness value of these five algorithms in each iteration are shown in Fig. 6 and Fig. 7. Comparing the curves in two figures, it can be seen that under six functions, the best fitness

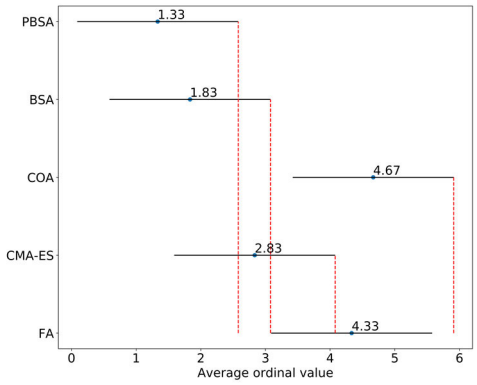

FIGURE 5. The result of Frideman test and Nemenyi post-hoc test.

values of FA and COA in each iteration are much lower than BSA and PBSA, indicating that their convergence speed and convergence accuracy are worse than BSA and PBSA. When optimizing f_1 and f_3, the convergence speed of CMA-ES in the early stage of the iteration is slower than that of BSA and PBSA, but the best fitness value in the later stage of the iteration is closer to BSA and PBSA, this means that when processing unimodal benchmark, the performance of CMA-ES is close to that of BSA and PBSA, but when processing multimodal benchmark, the performance of CMA-ES is not as good as that of BSA and PBSA. Except for the change curves representing BSA and PBSA under f_3 is very close, the change curves representing PBSA under the functions are lower than that of BSA, it means that the convergence speed and stability of PBSA are better than BSA, this might because that, in PBSA, once the position of the individual falls into the local optimum, the population will regroup the small group through gathering behavior, and this behavior helps the individual of to modify the learning direction and expand the search range to jump out of the local optimum.

Apart from the above comparison, Table 4 shows the runtime of the five algorithms iterated 100 times on different benchmarks. When the benchmark problem is unimodal, the runtime of the algorithm is generally short, and when the benchmark problem is multimodal, the calculated dimension increases and the runtime of the algorithm increases, among them, the runtime of the five algorithms is sorted from long to short approximately as BSA, CMA-ES, PBSA, COA and FA. The runtime of PBSA is slightly longer than that of BSA, this should be due to the fact that compared with BSA, PBSA has increased gathering behavior and fitness value com-parison, but this time difference is acceptable.

From the above results, it can be concluded that PBSA not only get the better convergence accuracy but also the convergence speed than other four algorithms, the efficiency of PBSA to deal with benchmark problems is verified.

B. SOIL HEAVY METAL CONTENT PREDICTION

To make the experimental results more credible, two sets of different data are used to carry out prediction experiments of heavy metal content in soil.

TABLE 3. Result comparison on 6 benchmark functions.

	Algorithm	Best	Mean	SD		Algorithm	Best	Mean	SD
f_1	FA	2.58E-02	5.69E+01	2.15E+01	f_4	FA	2.18E-04	2.43E+01	2.13E+00
	CMA-ES	3.71E-11	2.18E+04	1.17E+05		CMA-ES	0.00E+00	8.55E+01	1.50E+02
	COA	1.19E+01	2.01E+01	4.27E+00		COA	5.68E+01	1.00E+02	2.57E+01
	BSA	1.47E-15	2.04E-12	2.57E-12		BSA	0.00E+00	2.18E-10	7.56E-10
	PBSA	5.56E-17	2.76E-14	5.91E-14		PBSA	0.00E+00	2.72E-15	8.23E-15
f_2	FA	9.70E-02	6.73E+05	1.06E+06	f_5	FA	1.99E+01	2.09E+01	4.14E-01
	CMA-ES	4.29E-10	1.49E+06	5.04E+06		CMA-ES	2.88E-06	6.16E+00	7.19E+00
	COA	1.78E+04	8.33E+04	4.53E+04		COA	1.22E+01	1.70E+01	2.22E+00
	BSA	1.09E-16	1.46E-07	4.45E-07		BSA	3.15E-14	1.36E-08	4.89E-08
	PBSA	6.70E-24	2.07E-11	7.40E-11		PBSA	4.44E-16	1.35E-11	3.17E-11
f_3	FA	7.73E+02	2.76E+06	5.11E+06	f_6	FA	2.62E+02	5.90E+02	1.28E+02
	CMA-ES	2.03E+01	2.28E+06	4.97E+06		CMA-ES	7.94E-11	1.25E+02	3.40E+02
	COA	1.99E+03	1.56E+05	1.32E+05		COA	4.64E+01	7.79E+01	2.99E+01
	BSA	1.89E+01	1.89E+01	1.10E-01		BSA	0.00E+00	7.03E-17	2.46E-16
	PBSA	1.89E+01	1.90E+01	1.71E-01		PBSA	0.00E+00	0.00E+00	0.00E+00

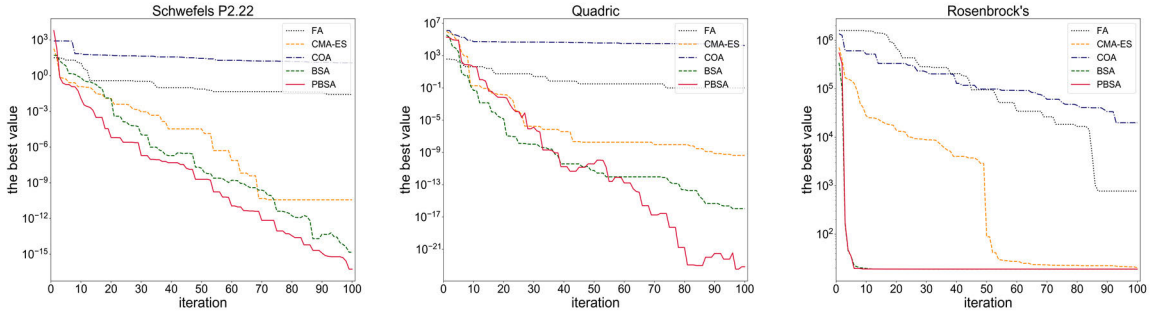

FIGURE 6. The change of best fitness value under $f_1 f_2$ and f_3.

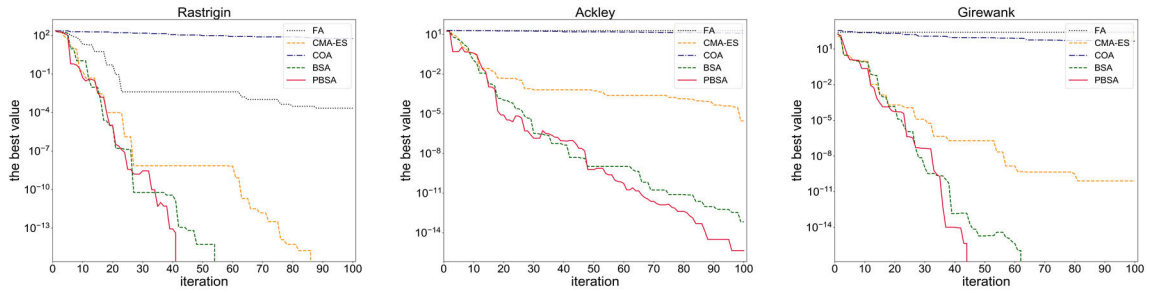

FIGURE 7. The change of best fitness value under $f_4 f_5$ and f_6.

The first set of data is selected from Yinchuan city of Ningxia, China and is the content of heavy metals in the surface soil which includes Co, Cr, Cs, Mg, Pb and Ti [32]. The heavy metal content of Co is to be predicted and other five heavy metal are the input data. Table 5 statistically analyzes the heavy metal content in these data.

The other set of data is selected from farmland soil heavy metal content in six new urban areas of Wuhan, China, this data was collected by Wuhan Academy of Agricultural Sciences and the soil heavy metal content included in this data are As, Cd, Cr, Cu, Ni, Pb, Zn and Hg, the statistical analysis of these data is shown in Table 6. In this experiment, Cr was selected as the heavy metal to be predicted, in order to determine the five types of input data for prediction, the correlation analysis between other heavy metal content and Cr was conducted, Pearson coefficient is used as the evaluation standard in this analysis, and the calculation results are shown in Table 7. The five heavy metals with the highest Pearson coefficient in Table 7 are As, Cd, Ni, Pb and Zn, which are used as the input data for the prediction of soil heavy metal content.

In the experiment of soil heavy metal content prediction, the hold-out validation has been adopted. There is a total of 96 sets in the first set of data, 76 sets of data are randomly chosen as training data and 20 sets of data are randomly chosen as test data. The other set of data has a total of 1000 sets,

TABLE 4. The runtime of five algorithms to iterate 100 times on six benchmark problems.

	Algorithms	Runtime (s)		Algorithms	Runtime (s)
f_1	FA	5.2791	f_2	FA	8.5596
	CMA-ES	0.2331		CMA-ES	0.3574
	COA	1.1145		COA	2.0414
	BSA	0.1548		BSA	0.2852
	PBSA	0.2286		PBSA	0.3912
f_3	FA	6.9841	f_4	FA	11.4157
	CMA-ES	0.3166		CMA-ES	0.6754
	COA	1.3285		COA	1.6883
	BSA	0.2494		BSA	0.4075
	PBSA	0.5422		PBSA	0.7256
f_5	FA	11.1711	f_6	FA	11.3572
	CMA-ES	0.4946		CMA-ES	0.5561
	COA	1.6151		COA	1.7197
	BSA	0.3993		BSA	0.3551
	PBSA	0.4832		PBSA	0.6998

TABLE 5. The statistical analysis of the heavy metal content in data 1.

Heavy metal	Minimum (mg/kg)	Maximum (mg/kg)	Mean (mg/kg)	Standard Deviation
Co	16.70	108.40	37.24	17.43
Cr	66.20	143.80	109.07	12.79
Cs	0.10	42.00	17.36	9.69
Mg	0.98	3.25	2.13	0.41
Pb	12.80	49.10	24.99	5.41
Ti	1189.00	2441.00	2040.23	341.14

TABLE 6. The statistical analysis of the heavy metal content in data2.

Heavy metal	Minimum (mg/kg)	Maximum (mg/kg)	Mean (mg/kg)	Standard Deviation
As	0.24	82.07	10.15	6.00
Cd	0.01	4.94	0.21	0.39
Cr	11.13	171.21	57.47	24.66
Cu	2.16	159.36	26.20	14.06
Ni	3.32	77.67	28.23	12.05
Pb	1.96	83.30	19.47	8.60
Zn	15.16	293.73	71.17	29.29
Hg	0.01	2.37	0.14	0.17

TABLE 7. The correlation results of the heavy metal and Cr.

Heavy metal	Pearson coefficient
As	0.593931
Cd	0.323458
Cu	0.147539
Ni	0.665215
Pb	0.635627
Zn	0.422606
Hg	0.041006

800 sets of data are randomly selected to use for training and 200 sets of data are randomly selected to use for test. Under these two sets of data, first the models are trained on the training data, then, the test data is used to make predictions by the trained models, and the analysis of the results are based on this. In addition to this, the data needs to be normalized before experiment and choose the maximum and minimum normalization as the normalization method.

During the experiment, four methods of SVM, RBFNN, WNN and bird swarm algorithm optimize WNN (BSA-WNN) were selected to establish the model and compare the experimental results with CCNN. In order to ensure the fairness of the experiment, all the heavy metal content prediction experiments are carried out in the same experimental environment, and it is the same as the experiment of PBSA. Except for SVM, the number of iterations for training other models is 100, at the same time, RBFNN and WNN use the same parameter initialization method and the input layer nodes, hidden layer nodes and output layer nodes of all neural network models are set to 5, 8, and 1. The parameter settings of these models are shown in Table 8.

TABLE 8. Parameter settings.

Model	Parameters
SVM	$c = 1, gamma = 0.1$
RBFNN	$learning\ rate = 0.01, ite = 100$
WNN	$learning\ rate = 0.01, ite = 100$
BSA-WNN	$learning\ rate = 0.01, ite = 100$ $m = 20, C = S = 1.5, a_1 = a_2 = 1$ $FQ = 5, FL \in [0.5,0.9], P \in [0.8,1]$
CCNN	$learning\ rate = 0.01, ite = 100$ $m = 20, C = S = 1.5, a_1 = a_2 = 1,$ $FQ = 5, FL_{max} = 0.9, FL_{min} = 0.5,$ $P \in [0.8,1]$

Fig. 8 and Fig. 9 show the comparison between the predicted value and the actual value on the training data and the test data when using dataset 1 for prediction and Fig. 10 and Fig. 11 show the comparison of that when using dataset 2 for prediction. When comparing the prediction results of training data and test data in the same dataset, it can be clearly seen that no matter in which dataset, the fit of the curve representing the predicted value and the actual value in the test data prediction is higher than the training data prediction, this is mainly because the training dataset is usually many times larger than the test dataset, so it is more difficult to fit the curve representing the predicted value to the curve representing the actual value. Since dataset 2 is larger than dataset 1, the data distribution between them is inconsistent, therefore, when the model is used to train different data, the training effect of the model is also different, it is the reason why when comparing Fig. 8 with Fig. 10 and Fig. 9 with Fig. 11, the curves in them show different degrees of fit. Besides this, when comparing the degree of curve fit between the predicted value and the true value in different models of each picture, the same result is obtained, that is the distance between the curve representing the predicted value and the curve representing the actual value in SVM and RBFNN is significantly greater than that of the other three models, in the comparison of the other three models, the distance between

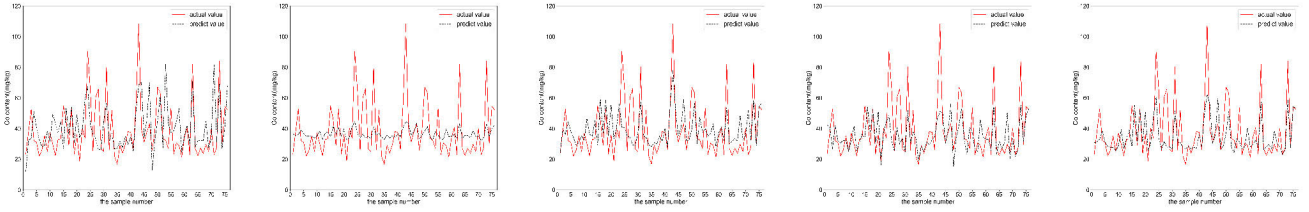

FIGURE 8. The prediction results of SVM, RBFNN, WNN, BSA-WNN and CCNN in training data of dataset 1.

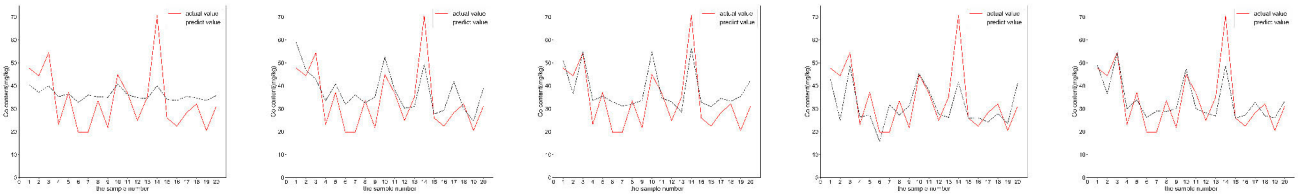

FIGURE 9. The prediction results of SVM, RBFNN, WNN, BSA-WNN and CCNN in test data of dataset 1.

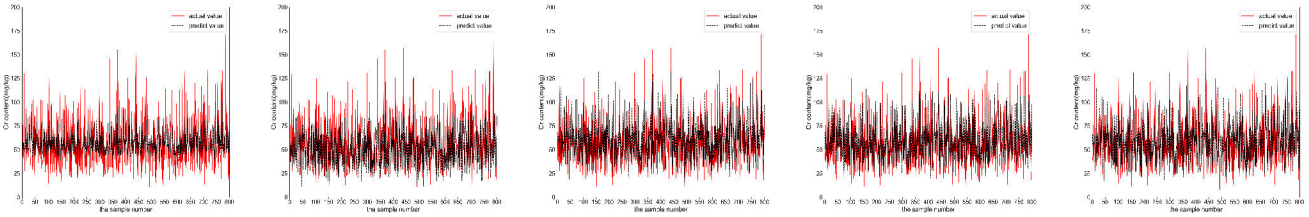

FIGURE 10. The prediction results of SVM, RBFNN, WNN, BSA-WNN and CCNN in training data of dataset 2.

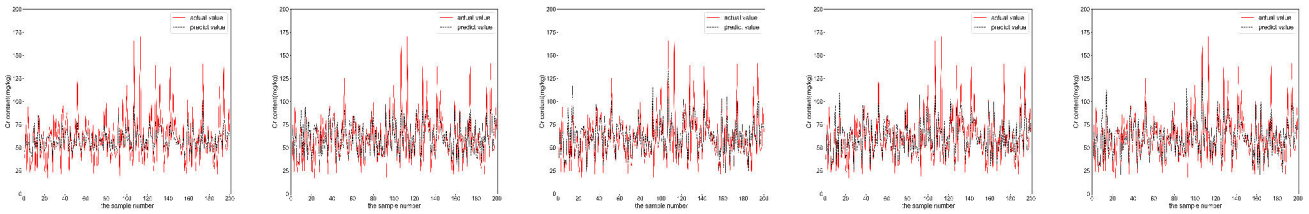

FIGURE 11. The prediction results of SVM, RBFNN, WNN, BSA-WNN and CCNN in test data of dataset 2.

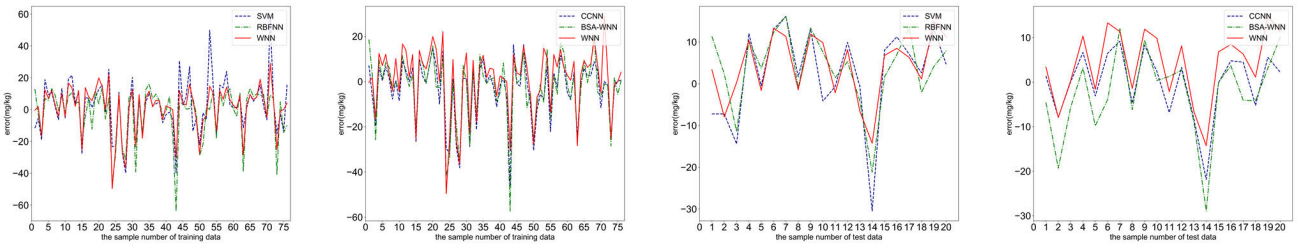

FIGURE 12. The error between predicted value and actual value of dataset 1.

the predicted value and the actual value of BSA-WNN and CCNN is smaller than that of WNN, and the overlap of the curves in BSA-WNN and CCNN is also increased, apart from this, the curve overlap of CCNN is greater than that of BSA-WNN.

The error after calculation between predicted value and actual value of each model on two datasets are shown in Fig. 12 and Fig. 13, Fig. 12 shows the error of 76 groups of training data and 20 groups of test data and Fig. 13 shows the error of 800 groups of training data and 200 groups of

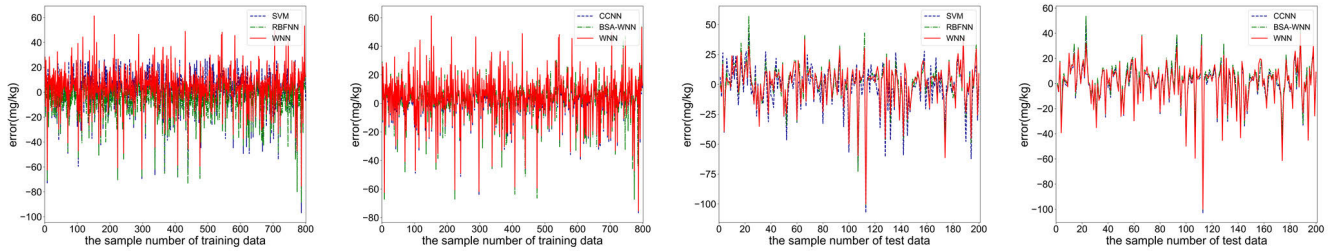

FIGURE 13. The error between predicted value and actual value of dataset 2.

test data. The same as the comparison between the predicted value and the actual value, due to the different datasets, the error values in Fig. 12 and Fig. 13 are also different. When observing these two figures, it can be seen that the error curve representing WNN does not completely cover the curves representing SVM and RBFNN, whether using training data or test data, and at most inflection points, the curves representing SVM and RBFNN are more prominent than the curves representing WNN, which indicates that the error of SVM and RBFNN is greater than WNN. Similarly, when comparing the error curves representing CCNN, BSA-WNN and WNN, it can be seen that although the curve fluctuates, the position of the error curve representing CCNN is always between the two curves representing WNN and BSA-WNN, and it is closer to the line with error value 0, which indicates that the error of CCNN is the smallest among the five models.

Besides comparing the error between the predicted value and the true value, the three error indicators of mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) are used to evaluate the prediction performance of the five models, the calculation formula are as follows.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{21}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{22}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{23}$$

where $\hat{y}_i$ is the predicted value and y_i is the actual value. Table 9 gives the calculation results of the five models under three error indicators in dataset 1 and dataset 2.

As can be seen from Table 9, when the three error indicators are calculated on dataset 1, the calculation results of the five models on the training data are generally greater than the calculation results on the test data, but when calculating on dataset 2, the calculation results is opposite to that on dataset 1, a possible reason is that, the model is mainly trained by using the training data, and the test data is used for the test of prediction, the data volume of dataset 2 is large and the calculation results of these five indicators are calculated based on the total amount of data, so the calculation results in the

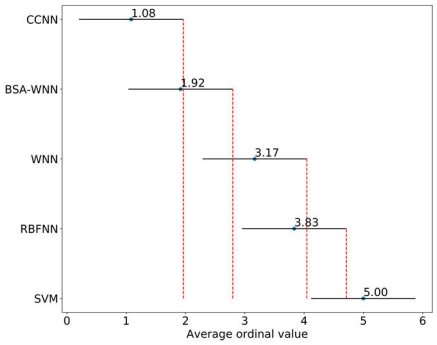

FIGURE 14. The result of Frideman test and Nemenyi post-hoc test.

training data are better than that in the test data, but dataset 1 has only 96 sets of data in total, the data volume of training data and test data are relatively small and close, therefore, the calculation results of the five indicators are more affected by the data volume, which leads to the calculation result in test data is better than that in the training data.

Apart from that, no matter in which dataset, whether it is training data or test data, it can be clearly seen that the three error indicators of SVM have the largest value, it might because that compared to neural networks, the prediction performance of traditional method SVM have lower prediction performance. When comparing WNN and RBFNN, the three error indicator values of the two are relatively close, WNN gets the smaller value on 10 out of 12 error indicator values, which shows that prediction performance of WNN is better than RBFNN. The three error indicator values of PBSA and BSA is smaller than WNN and PBSA gets the smallest error indicator values, this should be because when using BSA and PBSA to optimize the parameters in WNN, the initial parameters of WNN are better than the unoptimized ones, and the model can obtain better prediction performance when the model is trained under the same number of iterations, moreover, it has been proved that the optimization capability of PBSA is better than that of BSA in experiment 1, so the prediction performance of CCNN is better than BSA-WNN.

Same as experiment 1, the non-parametric statistical significance tests of Frideman and Nemenyi post-hoc are used to analyze the significance of the results obtained, the value of τ_F is 244.48 and the value of p_{value} is $2.54e$-09 after calculating the results in Table 9, it means that there is significant difference in all cases for an α level of 0.05, then use the

TABLE 9. The calculated value of error indicators in dataset 1 and dataset 2.

Dataset 1	Model	MAE (mg/kg)	RMSE (mg/kg)	MAPE (%)	Dataset 2	Model	MAE (mg/kg)	RMSE (mg/kg)	MAPE (%)
Training data	SVM	12.816	17.083	36.38	Training data	SVM	12.905	17.793	28.81
	RBFNN	11.751	16.339	32.43		RBFNN	12.302	17.176	24.29
	WNN	11.159	14.684	30.04		WNN	11.621	15.982	25.16
	BSA-WNN	9.968	14.648	23.74		BSA-WNN	11.143	15.514	23.15
	CCNN	9.461	13.771	21.77		CCNN	10.983	15.528	23.12
Test data	SVM	8.9031	11.296	30.67	Test data	SVM	13.964	20.262	27.69
	RBFNN	7.8967	9.6201	26.71		RBFNN	12.889	18.578	25.11
	WNN	7.5704	8.8208	27.81		WNN	12.576	17.817	24.21
	BSA-WNN	7.0729	7.7577	20.96		BSA-WNN	12.398	17.469	23.21
	CCNN	5.6373	7.2459	18.45		CCNN	11.892	17.391	22.39

Nemenyi Post-hoc to calculate the critical difference domain, the performance comparison results of the five algorithms are displayed using the Frideman test chart in Fig. 14. From Fig. 14, it can be seen that the horizontal line representing the CCNN critical difference domain does not overlap with the horizontal line representing the critical difference domain of WNN, RBFNN and SVM, and the average ordinal value of CCNN is smaller than that of WNN, RBFNN and SVM, which indicates that the prediction performance of these four models are significantly different and CCNN has a better prediction performance than WNN, RBFNN and SVM. The horizontal line representing the critical difference domain of CCNN overlaps with the horizontal line representing that of BSA-WNN and the average ordinal value of PBSA is smaller than BSA-WNN, it means that the prediction performance of CCNN is better than BSA-WNN.

TABLE 10. The runtime of five models for training.

Model	Runtime in dataset1(s)	Runtime in dataset2(s)
SVM	3.84	4.75
RBFNN	19.03	231.61
WNN	28.87	265.93
BSA-WNN	190.51	2163.72
CCNN	225.91	2285.62

In addition to the above comparisons, Table 10 gives the time required to train each model. In Table 10, it is easy to see that the runtime of each model is closely related to the amount of training data, since the data volume of dataset 2 is much larger than that of dataset 1, the runtime of the five models when using dataset 2 is much longer than that when using dataset 1. Compared the runtime required by each model when using the same datasets, it can be seen that the runtime required by SVM is the shortest, this may be because SVM only needs to perform one non-linear calculation, while the neural network model needs to perform multiple calculations. The runtime required by RBFNN is relatively close to that of WNN, indicating that the computational complexity of these two is basically the same. The time required for CCNN and BSA-WNN to run is much longer than WNN, of which CCNN has the longest runtime, this should be because compared to WNN, CCNN and BSA-WNN both have added a parameter optimization procedure, and the computational complexity of this procedure is higher than that of WNN, furthermore, the analysis in experiment 1 can explain that PBSA has a longer runtime than BSA, so the runtime of CCNN is longest.

From the above results, it can be concluded that compared with the other four models, CCNN has better prediction performance when predicting soil heavy metal content. However, owing to the fact that CCNN adds a parameter optimization procedure on the basis of WNN and parameter optimization is mainly achieved by PBSA, the computational complexity of PBSA and WNN are $O(MN)$ and $O(M)$ respectively, so the computational complexity of CCNN is higher when compared with WNN, resulting in much longer runtime, which is also a problem that needs to be further resolved.

VI. CONCLUSION

Based on wavelet neural network (WNN), a collaborative compound neural network model (CCNN) was proposed to make soil heavy metal content prediction. In particular, on the basis of BSA, the gathering behavior, the sine transformation position update method based on fitness difference ratio and acceptance criterion are introduced, and a new parallel bird swarm algorithm (PBSA) is put forward, which improves the internal information utilization rate of the population while enhances the algorithm's global search capability, then apply PBSA to optimize the parameters of WNN to ensure its prediction efficiency. Compared the optimization results of PBSA with BSA, COA, CMA-ES and FA on six benchmark problems and carry out the experiment of soil heavy metal content prediction in two sets of data with several models including CCNN, BSA-WNN, WNN, RBFNN and SVM, the effectiveness of PBSA is verified and the experimental results show that the prediction performance of CCNN is better than other models. Future work would include optimizing the structure of CCNN to reduce the time it takes to run and make its prediction performance better.

REFERENCES

[1] Z. Li, Z. Ma, T. J. van der Kuijp, Z. Yuan, and L. Huang, "A review of soil heavy metal pollution from mines in China: Pollution and health risk assessment," *Sci. Total Environ.*, vols. 468–469, pp. 843–853, Jan. 2014, doi: 10.1016/j.scitotenv.2013.08.090.

[2] Q. Yang, Z. Li, X. Lu, Q. Duan, L. Huang, and J. Bi, "A review of soil heavy metal pollution from industrial and agricultural regions in China: Pollution and risk assessment," *Sci. Total Environ.*, vol. 642, pp. 690–700, Nov. 2018, doi: 10.1016/j.scitotenv.2018.06.068.

[3] X. Jia, B. Hu, B. P. Marchant, L. Zhou, Z. Shi, and Y. Zhu, "A methodological framework for identifying potential sources of soil heavy metal pollution based on machine learning: A case study in the yangtze delta, China," *Environ. Pollut.*, vol. 250, pp. 601–609, Jul. 2019, doi: 10.1016/j.envpol.2019.04.047.

[4] G. Liang, J. Xu, and L. Liu, "QSPR analysis for melting point of fatty acids using genetic algorithm based multiple linear regression (GA-MLR)," *Fluid Phase Equilibria*, vol. 353, pp. 15–21, Sep. 2013, doi: 10.1016/j.fluid.2013.06.008.

[5] W. Xie, L. Yu, S. Xu, and S. Wang, "A new method for crude oil price forecasting based on support vector machines," in *Proc. Int. Conf. Comput. Sci.* New York, NY, USA: Springer-Verlag, 2006, pp. 444–451, doi: 10.1007/11758549_63.

[6] Z. Liao and J. Wang, "Forecasting model of global stock index by stochastic time effective neural network," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 834–841, Jan. 2010, doi: 10.1016/j.eswa.2009.05.086.

[7] C. L. P. Chen, Y.-J. Liu, and G.-X. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 583–593, May 2014, doi: 10.1109/TCYB.2013.2262935.

[8] M. R. Mustafa, R. B. Rezaur, H. Rahardjo, and M. H. Isa, "Prediction of pore-water pressure using radial basis function neural network," *Eng. Geol.*, vols. 135–136, pp. 40–47, May 2012, doi: 10.1016/j.enggeo.2012.02.008.

[9] Y. Wang and H. Peng, "Underwater acoustic source localization using generalized regression neural network," *J. Acoust. Soc. Amer.*, vol. 143, no. 4, pp. 2321–2331, Apr. 2018, doi: 10.1121/1.5032311.

[10] L. Tian, H. Liu, F. Luan, and Y. Gao, "QSPR study on the prediction of ionization potential of various organic compounds by heuristic method and radial basis function neural network," in *Proc. 7th Int. Conf. Natural Comput.*, Jul. 2011, doi: 10.1109/ICNC.2011.6022125.

[11] S. Zhang, H. Jiang, Y. Yin, W. Xiao, and B. Zhao, "The prediction of the gas utilization ratio based on TS fuzzy neural network and particle swarm optimization," *Sensors*, vol. 18, no. 2, p. 625, Feb. 2018, doi: 10.3390/s18020625.

[12] J. Adamowski and H. F. Chan, "A wavelet neural network conjunction model for groundwater level forecasting," *J. Hydrol.*, vol. 407, nos. 1–4, pp. 28–40, Sep. 2011.

[13] B. Xia, D. Cui, Z. Sun, Z. Lao, R. Zhang, W. Wang, W. Sun, Y. Lai, and M. Wang, "State of charge estimation of lithium-ion batteries using optimized Levenberg-Marquardt wavelet neural network," *Energy*, vol. 153, pp. 694–705, Jun. 2018, doi: 10.1016/j.energy.2018.04.085.

[14] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland, and S. J. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 322–330, Feb. 2010, doi: 10.1109/tpwrs.2009.2030426.

[15] X. Wen, S. Jian, and J. Wang, "Prediction models of calorific value of coal based on wavelet neural networks," *Fuel*, vol. 199, pp. 512–522, Jul. 2017, doi: 10.1016/j.fuel.2017.03.012.

[16] R. R. Molkdaragh, S. Jafarmadar, S. Khalilaria, and H. S. Saraee, "Prediction of the performance and exhaust emissions of a compression ignition engine using a wavelet neural network with a stochastic gradient algorithm," *Energy*, vol. 142, pp. 1128–1138, Jan. 2018, doi: 10.1016/j.energy.2017.09.006.

[17] L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, "Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting," *Appl. Energy*, vol. 198, pp. 203–222, Jul. 2017.

[18] W. Wang, M. Zhang, and X. Liu, "Improved fruit fly optimization algorithm optimized wavelet neural network for statistical data modeling for industrial polypropylene melt index prediction," *J. Chemometrics*, vol. 29, no. 9, pp. 506–513, Sep. 2015.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov. 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[20] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011, doi: 10.1109/TEVC.2010.2059031.

[21] A. Hackl, C. Magele, and W. Renhart, "Extended firefly algorithm for multimodal optimization," in *Proc. 19th Int. Symp. Electr. App. Technol. (SIELA)*, May 2016, pp. 169–178, doi: 10.1109/SIELA.2016.7543010.

[22] N. Hansen, "Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed," in *Proc. 11th Annu. Conf. Companion Genetic Evol. Comput. Conf. (GECCO)*, Jul. 2009, pp. 2389–2395, doi: 10.1145/1570256.1570333.

[23] J. Pierezan and L. Dos Santos Coelho, "Coyote optimization algorithm: A new Metaheuristic for global optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, doi: 10.1109/CEC.2018.8477769.

[24] X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, "A new bio-inspired optimisation algorithm: Bird swarm algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 28, no. 4, pp. 673–687, Jul. 2016, doi: 10.1080/0952813X.2015.1042530.

[25] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 889–898, 1992, doi: 10.1109/72.165591.

[26] L. Huang and J. Wang, "Global crude oil price prediction and synchronization based accuracy evaluation using random wavelet neural network," *Energy*, vol. 151, pp. 875–888, May 2018, doi: 10.1016/j.energy.2018.03.099.

[27] M. R. G. Razin and B. Voosoghi, "Wavelet neural networks using particle swarm optimization training in modeling regional ionospheric total electron content," *J. Atmos. Solar-Terr. Phys.*, vol. 149, pp. 21–30, Nov. 2016.

[28] F. Jiang, L. Dong, and Q. Dai, "Electrical resistivity imaging inversion: An ISFLA trained kernel principal component wavelet neural network approach," *Neural Netw.*, vol. 104, pp. 114–123, Aug. 2018, doi: 10.1016/j.neunet.2018.04.012.

[29] M. Ahmad, N. Javaid, I. A. Niaz, S. Shafiq, O. U. Rehman, and H. M. Hussain, "Application of bird swarm algorithm for solution of optimal power flow problems," in *Proc. Conf. Complex, Intell., Softw. Intensive Syst.* Cham, Switzerland: Springer, 2018, pp. 280–291, doi: 10.1007/978-3-319-93659-8_25.

[30] C. Zeng, C. Peng, K. Wang, Y. Zhang, and M. Zhang, "Multi-objective operation optimization of micro grid based on bird swarm algorithm," *Power Syst. Protection Control*, vol. 44, pp. 117–122, Jul. 2016, doi: 10.7667/PSPC151381.

[31] X. Wang, Y. Deng, and H. Duan, "Edge-based target detection for unmanned aerial vehicles using competitive bird swarm algorithm," *Aerosp. Sci. Technol.*, vol. 78, pp. 708–720, Jul. 2018, doi: 10.1016/j.ast.2018.04.047.

[32] Z. Mingxin, "Data set of heavy metal content in surface soil of Yinchuan City, Ningxia," *Global Change Res. Data Repository*, vol. 2, pp. 198–204, Apr. 2018, doi: 10.3974/geodb.2018.04.10.V1.

WENQI CAO received the B.S. degree from the School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan, China, in 2018. He is currently pursuing the master's degree in mathematics and computer science with Wuhan Polytechnic University. His research interest includes artificial intelligence technology and its application.

CONG ZHANG received the bachelor's degree in automation engineering from the Huazhong University of Science and Technology, in 1993, the master's degree in computer application technology from the Wuhan University of Technology, in 1999, and the Ph.D. degree in computer application technology from Wuhan University, in 2010. He is currently a Professor with the School of Mathematics and Computer Science, Wuhan Polytechnic University. His research interests include multimedia signal processing, multimedia communication system theory and application, and pattern recognition.