A GOETHITE PROCESS MODELING METHOD BY ASYNCHRONOUS FUZZY COGNITIVE NETWORK BASED ON AN IMPROVED CONSTRAINED CHICKEN SWARM OPTIMIZATION ALGORITHM

JUNJIE PENG, NING CHEN, JIAYANG DAI* AND WEIHUA GUI
School of Automation, Central South University
Changsha 410083, China

(Communicated by Biao Luo)

ABSTRACT. In order to solve the problem that the mechanism model of non-linear system with uncertainty is difficult to establish, a modeling method of nonlinear system based on Asynchronous Fuzzy Cognitive Network (AFCN) is proposed. This method combines fuzzy cognitive network with time-lag system, and extends the node state values and weights of fuzzy cognitive network to the time interval, which enhances the adaptability of the model. At the same time an improved constrained chicken swarm optimization algorithm (ICCSOA) is proposed to identify model parameters of AFCN. A lag matrix corresponding to the actual measured values of the system lag of the nodes in the AFCN model is introduced, and a correction term including the difference between the measured values and the predicted values of the system is added to the model parameter updating mechanism. The simulation experiment results of goethite process system shows this modeling method can be used to model complex systems with uncertainties or partial missing data. The control model based on the established system model can make correct control decisions. ICCSOA has the advantages of fast convergence speed and accurate learning results, whose global search ability and convergence accuracy are higher than those of CSO algorithm, which can be widely used to the modeling of uncertain systems.

1. Introduction. Zinc is an important non-ferrous metal. At present, most zinc smelting enterprises adopt the method of direct leaching zinc with high-iron zinc sulfide concentrate, which can effectively reduce sulfur dioxide emissions and improve the recovery rate of leaching slag[9]. Among them, zinc sulfate solution generated from high-iron zinc sulfide concentrate can enrich a large number of iron ions. However, the excessive concentration of iron ions in the solution would lead to overmuch impurity of finished zinc ingot, as well as the rising power consumption in the electrolysis process and the unstable production. Therefore, goethite process for iron-removal is an important step in zinc smelting.

2010 Mathematics Subject Classification. 90C26, 90C70, 90C59.

Key words and phrases. Fuzzy cognitive network, time-lag system, chicken swarm optimization algorithm, terminal constraint.

This research is supported by the Program of National Science Foundation of China, grant number 61673339 and the Program of National Science Foundation of Hunan Province, grant number 2017JJ2329.

* Corresponding author: Jiayang Dai.
The difficulty of modeling goethite process lays in its complicate mechanism, the process has random uncertainties of dynamic data and information. In addition, the goethite process has large time-lag\cite{2, 3, 4, 15}. Fuzzy Cognitive Map (FCM)\cite{12} is an intelligent method can deal with the dynamic data and information with stochastic uncertainty. The method can also express the inherent characteristics and patterns of the uncertain system as a model that can be easily solved and described by natural language. FCM is a soft computing method combining neural network with fuzzy logic\cite{16, 17, 22}, which is a directed graph composed of nodes, arcs and weights. The node in cognitive map is called concept node, which represents a characteristic of a describing system, such as entity, action, behavior, cause, result, tendency, trend and index. The signed and weighted arcs connecting the two conceptual nodes represent the causal relationship among these concepts. The range of values of conceptual states is \([0,1]\), and the range of weights is \([-1,1]\). FCM has good dynamic characteristics and learning ability. It has strong advantages in modeling, process analysis and decision-making\cite{20}. Anninou et al. used FCM to present a mathematical model of human Parkinson’s syndrome by analyzing the related factors that may lead to human Parkinson’s syndrome\cite{1}. Fatahi analyzed the relationship between personality and emotion, and established a mathematical model based on FCM, which was of great significance to the development of humanized system\cite{11}. Kreinovich simplified the multivariate neural network into a unitary neural network using Miller criterion, which was proved that only experience and subjective consciousness can make FCM describe the system effective\cite{13}. Obiedat et al. combined dynamic system with neural network to establish an FCM model of social ecosystem. The results showed that the method can provide appropriate suggestions for decision makers\cite{19}. Christen used FCM as a tool to analyze the experience of agricultural system management in Scotland and found the problem of poor agricultural system\cite{5}. Zhang proposed a T-S modeling method based on FCM for large-scale non-linear systems and verified it in the reversing system\cite{28}. Traditional FCM modeling method has strong dependence on expert knowledge through offline simulation of system process. If the expert’s knowledge is not accurate enough, the system FCM model cannot correctly reflect the actual state of the current system. In order to make more accurate control and decision for real-time systems\cite{14}, a FCM-based Fuzzy Cognitive Network (FCN) modeling method was proposed\cite{6, 7, 8}. FCN consists of a fuzzy cognitive map, an update mechanism based on actual system feedback and the storage of knowledge obtained from the whole operation process. FCN is only used at the initial point or completely divorced from expert experience, which can overcome the shortcomings of traditional FCM modeling methods that rely on expert experience heavily.

For the complex multiple input multiple output system (MIMO system), the lags of different input variables may be different. Yet, the traditional fuzzy cognitive network is suitable for the modeling of process with the same input variable lag, while the unsynchronized fuzzy cognitive network can fully reflect the time lags of each input variable in the system. Therefore, for the goethite system whose different input variables have different time lags, an asynchronous fuzzy cognitive network (AFCN) is proposed for modeling.

Past FCN modeling methods cannot accurately describe systems with different time-lags. The parameter setting of FCN model greatly affects the accuracy of the model. The weight learning method of FCN model can constantly modify the parameters of the model through a certain updating mechanism, so as to improve
the accuracy of FCN model. Wojciech et al. applied genetic algorithm to weight
calculation[23]. According to the principle of survival of the fittest, individuals with
the strongest adaptability to the environment were selected to cross and mutate, and
the mutation process produced a new generation of population with stronger
adaptability. The first disadvantage of genetic algorithm is that it is easy to pre-
mature and has poor local optimization ability. Second, it is unavoidable to search
the same feasible solution many times, which also affects its operational efficiency.
Parsopoulo et al.[21] proposed a learning method based on particle swarm optimi-
ization (PSO), whose basic idea was to simulate the predatory behavior of birds.
PSO algorithm is a local search algorithm with high efficiency, but it has some
shortcomings such as with low precision and easy divergence. If the parameters
such as acceleration coefficient and maximum velocity are too large, the particle
swarm may miss the optimal solution and the algorithm cannot converge. In the
case of convergence, the particles tend to lose diversity and easily fall into the local
optimal solution because all solutions approach the optimal solution at the same
time. Chicken swarm algorithm was a swarm intelligence algorithm based on search
behavior of chicken flock proposed by Meng et al. in 2014[18], which divided the
chicken flock into several subgroups. Each subgroup consisted of a cock, several
hens and chickens. Different kinds of chickens followed different moving rules. Indi-
viduals in each subgroup searched for food around the cock in the subgroup,
while the cock moved randomly. Chicken swarm optimization is a global optimiza-
tion algorithm with high convergence accuracy and good robustness. Yu applied
chicken swarm algorithm to hybrid localization of deep mine monitoring based on
Wireless Sensor Network[27]. Djaafar extended the chicken swarm optimization
(CSO) to solve multi-objective optimization problems and address the integration
of the archive population that guides the chicken swarm towards the Pareto optimal
solutions[10]. Dh.Wu builds a Markov chain model of CSO with finite homogeneous
Markov chain by using stochastic process theory, and analyses some properties of
Markov chain[24]. Wang combines chicken swarm algorithm with chaos theory for
wind power interval prediction[25]. However, it converges slowly and easily falls
into local optimum when solving multi-extremum optimization problems. Form the
above, general optimization algorithms have weak global search ability and are easy
to fall into local optimum.

Aiming at the problems of chicken swarm algorithm, this paper proposes an
improved constrained chicken swarm algorithm (ICCSO), which improves the basic
chicken swarm algorithm on two aspects of constrained function and evolutionary
mechanism. The simulation results of the standard test functions show that the
algorithm outperforms the basic chicken swarm algorithm and the improved chicken
swarm algorithm in convergence speed and global search ability.

On the basis of FCM and FCN, this paper proposes an Asynchronous Fuzzy Cog-
nitive Network (AFCN) modeling method, and applies the improved constrained
chicken swarm algorithm proposed in this paper to learning network weights, which
can be effectively used to model complex systems with uncertainties or partial miss-
ing data. This paper is arranged as follows, The second part introduces the AFCN
modeling mechanism and its characteristics for goethite process. The third part
proposes an improved constrained chicken swarm algorithm. The fourth part veri-
fies the effectiveness of the proposed modeling method by using a control strategy
of goethite process. The last part presents the conclusion.
2. Asynchronous fuzzy cognitive network modeling based on goethite process.

2.1. Goethite process. The main facilities of the goethite process are five continuous stirred tank reactor (CSTR) that connected in series, as shown in Figure 1. The liquid in reactor 1# is mostly zinc sulfate solution from the previous process, and the minority is the reflux containing seed of crystal of goethite. The liquid will flow downstream to 2# until 5# reactor in turn. Calcine (main component is ZnO) is added at the top of reactors 1# -4#, while oxygen is injected into of reactors 1# -5# from their bottom. The complex oxidation, hydrolysis and neutralization reactions of Fe$^{2+}$, Fe$^{3+}$, and H$^+$ are occurred in reactors, resulting in gradual iron-removal from zinc sulfate solution in form of goethite. The outlet solution of 5# reactor is deposited in the thickener, whose supernatant is sent to the next process. Liquid from the bottom of the thickener is mostly sent to the filter press to remove the goethite solid, and part of it is the reflux in reactor 1#.

![Figure 1. Process flow chart of goethite process in a smelting enterprise](image)

The main chemical reactions in the reactors can be simplified by the following three chemical equations:

Oxidation reaction:

$$4Fe^{2+} + 4H^+ + O_2 \rightarrow 4Fe^{3+} + 2H_2O$$  \hspace{1cm} (1)

Hydrolysis reaction:

$$Fe^{3+} + 2H_2O \leftrightarrow FeOOH + 3H^+$$  \hspace{1cm} (2)

Neutralization reaction:

$$2H^+ + ZnO \rightarrow Zn^{2+} + H_2O$$  \hspace{1cm} (3)

After iron-removal, the solution needs to meet the process requirements in Eq. (4). Practically, the goethite process is affected by many factors, such as temperature, concentration of iron ion, pH value, oxygen flow, pressure, reaction time, stirring intensity and so on. Among them, concentration of iron ion and pH value play decisive roles in the production of goethite. In goethite process, concentration
of iron ion is affected by the oxidation rate of $\text{Fe}^{2+}$. The rapid oxidation reaction means excessive content of $\text{Fe}^{3+}$ in a short time, leading to unmoral hydrolysis of it. On the contrary, if the oxidation rate is too slow, the concentration of iron ion after iron-removal cannot satisfy the process requirement. On the other hand, low pH value will reduce the efficiency of iron-removal, while high pH value will precipitate other components unnecessarily. Especially when the concentration of $\text{Fe}^{3+}$ and pH value in the solution are high, $\text{Fe}^{3+}$ will precipitate rapidly and form colloids.

$$\begin{cases}
c_{\text{Fe}^{2+}} + c_{\text{Fe}^{3+}} & \leq 1\text{g/L} \\
c_{\text{Fe}^{2+}} & < 0.5\text{g/L} \\
3.5 & \leq \text{pH} \leq 4.0
\end{cases} \quad (4)$$

The goethite process is actually happened in all five reactors, and the concentrations of ions are gradually reduced to the required ranges. However, taking $\text{Fe}^{2+}$ concentration as an example, the too gentle or too steep decreasing gradient of $\text{Fe}^{2+}$ concentration in one reactor would have certain effect on the $\text{Fe}^{2+}$-related reactions of the next reactor, eventually failing to meet the process requirement Therefore, the decreasing range of $\text{Fe}^{2+}$ concentration in each reactor needs to be within a reasonable range.

In the actual chemical reactions, the contact of reactants cannot generate products immediately; the time between the reaction start and products appear is called time-lag, and there are different time-lags in different chemical reactions. From chemical reaction kinetics, the oxidation rate of $\text{Fe}^{2+}$, which goes fastest in the whole goethite process, is mainly related to its own concentration and the concentration of dissolved oxygen in solution. The hydrolysis rate of $\text{Fe}^{3+}$ depends only on its own concentration. Practically, the $\text{Fe}^{3+}$ concentration is relatively low in the goethite process, making the $\text{Fe}^{3+}$ hydrolyzes in low speed. When modeling the goethite process by FCN, the difference between oxidation rate and hydrolysis rate makes state renewal of the network is not synchronous. Therefore, the traditional FCN modeling method cannot accurately simulate the dynamic changes of the actual goethite system. A new FCN modeling method suitable for complex goethite process needs to be developed in view of the particularity of goethite process, making full use of process mechanism, expert knowledge and historical data.

2.2. Asynchronous fuzzy cognitive network (AFCN). Asynchronous Fuzzy Cognitive Network (AFCN) is an extension of FCN to reflect the actual system characteristic, which is a system model combining fuzzy logic and neural network. A non-linear system can be represented by AFCN as a directed graph with nodes, weights, system feedback and system time-lag. When the directed graph takes different initial values and weights, it corresponds to different operating conditions of the system. $C = \{C_1, C_2, ..., C_n\}$ is used to represent the concept set of vertices of a digraph, and each node represents some characteristics of the system, such as variables, states, events, targets, etc. $A^k_i$ is used to represent the state of node $C_i$ at time $k$. Its value is converted from the actual value of the system, and its range is $[0, 1]$. $W_{ij}$ denotes the causal effect between node $C_i$ and $C_j$, and the range is $[-1, 1]$. if $W_{ij} > 0$, then the state of the result concept node $C_j$ varies positively with the state value of the cause concept node $C_i$ Conversely, if $W_{ij} < 0$, the state of $C_j$ varies inversely with the state value of $C_i$; When $W_{ij} = 0$, the state of $C_j$ is not related to the state value of $C_i$.

In AFCN model, nodes can be divided into control nodes and state nodes. The control node is the control variable of the system, which is not affected by other
nodes of the system and only related to the input variable. The state node represents the state of the system at a certain time, which is the state variable of the system. The state value of the control node is updated to:

\[ A_i^{k,AFCN} = A_i^{system} \]  

(5)

\( A_i^{k,AFCN} \) is the calculated value of AFCN when node \( C_i \) at time \( k \), and \( A_i^{system} \) is the actual system value of node \( C_i \), which is measured online or given in advance.

The relationship between the dynamic behaviors of the system is stored in the causal relationship between the network structure of the cognitive map and the interaction between nodes. In the model, the state value of a node at a certain time is affected by its own state value at that time and the value of the node that has causal relationship with it. In a time-lag system, the state value of a node at a certain time is not only related to its state value at the previous time, but also to the state value of the related nodes at several times before current time, meaning the state value of a node is

\[
A_j^k = f \left( A_j^{k-1} + \sum_{i=1, i\neq j}^n A_i^{k-1} W_{ij}^1 + \sum_{i=1, i\neq j}^n A_i^{k-2} W_{ij}^2 + \ldots + \sum_{i=1, i\neq j}^n A_i^{k-x} W_{ij}^x \right)
\]

(6)

among them, \( A_j^k \) is the state value of node \( C_j \) at time \( k \), \( A_j^{k-1} \) is the state value of node \( C_j \) at time \( k-1 \), \( A_i^{k-1} \) is the state value of node \( C_i \) at time \( k-1 \), \( A_i^{k-x} \) is the state value of node \( C_i \) at time \( k-x \), \( W_{ij}^x \) is the weight value of node \( C_i \)'s influence on node \( C_j \) at time \( k-x \), \( f \) is the threshold function, and \( W \) is an \( n \times n \) matrix.

Eq.(6) can be simplified to:

\[
A_j^k = f \left( A_j^{k-1} + \sum_{i=1, i\neq j}^n \sum_{m=k-G_{ij}}^{k-1} A_i^m W_{ij}^m \right)
\]

(7)

\[
\Gamma = \begin{bmatrix}
  \Gamma_{11} & \Gamma_{21} & \cdots & \Gamma_{n1} \\
  \Gamma_{21} & \Gamma_{22} & \cdots & \Gamma_{n2} \\
  \vdots & \vdots & \ddots & \vdots \\
  \Gamma_{n1} & \Gamma_{n2} & \cdots & \Gamma_{nn}
\end{bmatrix}
\]

(8)

among them, \( n \) is the number of nodes in the system model, \( A_j^k \) and \( A_j^{k-1} \) are the state values of node \( j \) at time \( k \) and time \( k-1 \), respectively. \( A_i^m \) is the state value of node \( i \) at time \( m \), \( W_{ij}^m \) is the weight value between node \( i \) and node \( j \) at time \( m \), and \( m \leq k-1 \). \( \Gamma_{ij} \) is the time required for node \( i \) to influence node \( j \), namely time-lag. \( \Gamma \) is defined as \( n \times n \) system time-lag matrix, in which each element represents the time required for one node to influence another node in the system model. \( f \) represents a transition function to ensure that the state value is converted to \([0,1]\). The transition function is \( f = 1/(1 + e^{-cx}) \) in this paper.

In order to obtain time-lag between nodes in AFCN, a method of analyzing the molar mass changes of ions in unit time involved in the oxidation, hydrolysis and neutralization reactions is adopted in this paper. It is assumed that the time required to consume 4 \textit{mol} of ferrous ion in the oxidation reaction is \( t \), and the reaction reaches equilibrium state at the same time. According to the law of mass conservation, in the oxidation shown in Eq. (1), it is necessary to consume 4 \textit{moles} of ferrous ion and 1 \textit{mol} of oxygen for producing 4 \textit{moles} of ferric ion. Similarly, from Eq. (2), the 4 \textit{moles} of ferric ion produced by the oxidation reaction will produce 12
moles of hydrogen ion ideally. Considering the 4 moles of hydrogen ion consumed by the oxidation and the 12 moles of hydrogen ion produced by the hydrolysis, there are 8 moles of hydrogen ion remained in the solution. In the neutralization reaction shown in Eq. (3), 4 moles of ZnO is required to consume the 8 moles of hydrogen ion. In summary, it takes time $4t$ to consume $4\text{moles}$ of oxygen and time $t$ to consume 4 moles of ZnO, it takes time $t$ get 4 moles of ferric ion and time $0.5t$ to get 4 moles of hydrogen ion. Therefore, the molar mass ratio of the substances involved in the chemical reaction is the time-lag between the nodes, such as the time required for the ferrous ion represented by $C_3$ to affect the hydrogen ion represented by $C_5$ is 2, whereas the time required for $C_5$ to influence $C_3$ is 0.5, meaning the time-lag is 2 and 0.5, respectively.

Combining with the established AFCN model of the 1# reactor of the goethite process, the time-lag matrix between the nodes in the model of this system is as follows:

$$
\Gamma = \begin{bmatrix}
0 & 0 & 8 & 8 & 16 \\
0 & 0 & 2 & 2 & 4 \\
0 & 0 & 0 & 2 & 4 \\
0 & 0 & 2 & 0 & 4 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
$$

among them, there is no lag between the node and itself, so the time-lag is 0 The stable node in the system is not affected by other nodes, hence, the time-lag between the node and other nodes is also 0.

Based on the analysis of the mechanism of goethite process, the reactions mainly involves the concentration change of ferrous ion, ferric ion hydrogen ion, and the addition of oxygen and calcine. The concentrations of ferrous ion, ferric ion and hydrogen ion interact with each other, and are also affected by oxygen and calcine. Oxygen and calcine are the control variables, which remain unchanged during the whole goethite process and are not affected by themselves and other nodes. Therefore, the AFCN model of 1# reactor for goethite process as shown in Figure 2 is established.

**Figure 2. AFCN model of 1# reactor for goethite process**

The nodes in the model are expressed as follows: $C_1$ is the oxygen flow rate, $C_2$ is the addition amount of calcine, $C_3$ indicates the concentration of ferrous ion in solution, $C_4$ is the concentration of ferric ion in solution Likewise, $C_5$ represents the concentration of hydrogen ion in solution. Among them, the oxygen flow rate and the amount of calcine added directly affect the value of other variables in the goethite process Therefore, $C_1$ and $C_2$ are the control nodes, i.e. the state of $C_1$ and
C_2 is not affected by other nodes, but remains unchanged and affects other nodes. The concentrations of ferrous ion, ferric ion and hydrogen ion in the solution reflect the reactions in the reactor hence, C_3, C_4 and C_5 are state nodes, representing the state of the system at a certain time. C_1, C_2, C_3, C_4 and C_5 are all normalized in interval [0, 1].

In the AFCN model of the 1# reactor in goethite process shown in Figure 2, the state node C_3, C_4, C_5 cannot be updated synchronously. Therefore, it is necessary to deeply analyze the relationship between the state value of each node and the time variation in the goethite process. On this basis, the asynchronous fuzzy cognitive network model of the process is transformed into a synchronous fuzzy cognitive network model through Eq.(7) and Eq.(8).

The aforementioned AFCN model can be used as an independent open-loop controller on the actual system. However, due to the incompleteness of the initial knowledge based on expert experience, and the unexpected interference during the operation of the system, the system will ultimately fail to reach the expected value. Therefore, in order to avoid the above situation and realize the close connection between the model and the actual system, it is necessary to adjust the weights between nodes according to the actual value of the system. That is, the weight updating is needed. The purpose of using feedback mechanism in AFCN control method is to adjust the weights between nodes in the model by weight updating, making the corresponding model more consistent with the current state of the system.

3. Weight learning based on improved constrained chicken swarm optimization.

3.1. Weight optimization of AFCN. On the premise of the established AFCN model, the weights in the model need to be identified. Definition of objective function F(W) is,

\[ F(W) = \sum_{j=1}^{n} \left( |A_{j}^{\text{min}} - A_{j}^{k}|^2 + |A_{j}^{\text{max}} - A_{j}^{k}|^2 \right) \] (10a)

\[ W = \begin{bmatrix} W_{11} & \cdots & \cdots & \cdots & W_{1n} \\ \vdots & \ddots & & & \vdots \\ \vdots & & W_{ij} & \ddots & \vdots \\ \vdots & & & \ddots & \vdots \\ W_{n1} & \cdots & \cdots & \cdots & W_{nn} \end{bmatrix} \] (10b)

Each element W_{ij} in matrix W means the influence of node C_i on node C_j, A_j^k is the state value of the output node C_j at time k which can be obtained by Eq.(3), and it satisfies the following condition,

\[ A_{j}^{\text{min}} \leq A_{j} \leq A_{j}^{\text{max}}, j = 1, 2, ..., n \] (11)

When all the output nodes are in the expected range and the objective function F(W) is globally minimized, AFCN is in stable state. In this case, the matrix W can be obtained by the algorithm proposed in this paper.
3.2. Basic chicken swarm optimization algorithm. After the structure of AFCN is determined, if there are \( A \) weights in the weight matrix, \( B \) initial values are generated randomly for each weight to optimize. To get the optimal weights as soon as possible, the chicken swarm optimization algorithm is used. In the basic chicken swarm optimization algorithm, some individuals with the best fitness (i.e., objective function) are regarded as cocks, some individuals with the worst fitness as chickens, and the remaining individuals as hens. The hen chooses the cock randomly, and the mother-child relationship between the hen and the chicken is also established randomly. Once the relationships are established, the hierarchy, dominance and mother-child relationships remain unchanged until several generations later. Individuals in each subgroup look for food around the cock, which moves randomly and updates the group through iteration to obtain the optimal weight.

When solving the optimization problem, each individual in the chicken flock corresponds to a solution of the optimization problem. Assuming that all individuals in the flock are \( N \), the number of cocks, hens, chickens and mother hens are \( N_R \), \( N_H \), \( N_C \) and \( N_M \), respectively. \( W_{p,q}(p \in [1,N], q=1,D) \) denotes the position of individual \( p \) in the D-dimensional search space at the \( t \)th iteration.

In the chicken flock, the location updating equations of cocks, hens and chickens are different. Cocks are the most adaptable group of individuals in the chicken flock. They look for food in a wider space. The corresponding location update equation is as follow,

\[
W_{p,q}\left(t+1\right) = W_{p,q}(t) \ast \left[1 + \text{randn}(0, \sigma^2)\right]
\]

\[
\sigma^2 = \begin{cases} 
1, & \text{if } h_p \leq h_v \\
\exp\left(\frac{h_v - h_p}{|h_p| + \varepsilon}\right), & \text{other}
\end{cases}
\]

\text{randn}(0, \sigma^2) \text{ is a Gaussian distribution with mean 0 and variance } \sigma^2 \text{. } \varepsilon \text{ is a very small positive constant. } v(v \in [1,N], v \neq p) \text{ is the cock index selected randomly from all noncocks And } h_v \text{ represents the corresponding fitness value of cock } v.

The equation for updating the position of hen is as follows:

\[
W_{p,q}\left(t+1\right) = W_{p,q}(t) + S_1 \ast \text{rand} \ast \left(W_{r1,q}(t) - W_{p,q}(t)\right) + S_2 \ast \text{rand} \ast \left(W_{r2,q}(t) - W_{p,q}(t)\right)
\]

\[
S_1 = \exp\left(\frac{h_p - h_{r1}}{|h_i| + \varepsilon}\right)
\]

\[
S_2 = \exp(h_{r2} - h_p)
\]

\text{rand is a random number between } [0,1]\text{. } r_1 \text{ is the corresponding cock of henpin the subgroup } r_2 \text{ is a randomly selected individual cock or hen, and } r_1 \neq r_2.

The equation for updating the position of chicken is

\[
W_{p,q}\left(t+1\right) = W_{p,q}(t) + FL \ast \left(W_{m,q}(t) - W_{p,q}(t)\right)
\]

\( m \) is the hen corresponding to the chicken \( FL \) (\( FL \in [0,2] \)) is the coefficient between the follower and the followed individual, indicating the tightness of the chicken following its corresponding hen in food searching.

3.3. Improved constrained chicken swarm optimization algorithm. The basic chicken swarm algorithm converges slowly and easily falls into local optimum. This paper improves basic algorithm from aspects of constraint function and evolutionary mechanism.
### 3.3.1. Improvement on constrains of chicken swarm optimization algorithm.

In the basic chicken swarm algorithm, if the individual is out of the search range, the search is restarted from the nearest boundary, i.e.

$$W_{t_{p,q}} = \begin{cases} \text{Lb}_q, W_{t_{p,q}} < \text{Lb}_q \\ \text{Ub}_q, W_{t_{p,q}} > \text{Ub}_q \end{cases}$$  \hspace{2cm} (18)

$Lb_q$ and $Ub_q$ represent the lower and upper boundaries of $q$-dimension.

Chicken swarm algorithm replaces constraints with the corresponding upper and lower bounds, resulting in unsatisfactory convergence speed. To improve its convergence speed, the random movement of the corresponding values of the best subgroup in the current chicken flock is proposed to replace the constraints:

- if $W_{t_{p,q}} < Lb_q$ or $W_{t_{p,q}} > Ub_q$ then $temp = W_{t_{h,q}} + \lambda_1 \text{randn}(0,1)$
- if $Lb_q \leq temp \leq Ub_q$ then $W_{t_{p,q}} = temp$ or $W_{t_{p,q}} = W_{t_{h,q}}$

Among them, $\lambda_1 = \begin{cases} 0.2 \ast (W_{t_{h,j}} - W_{t_{i,j}}), i \leq (N - N_c) \\ 0.2 \ast (W_{t_{h,j}} - W_{t_{g,j}}), i > (N - N_c) \end{cases}$  \hspace{2cm} (19)

$\lambda_1$ is the step size, randn(0,1) is standard normal distribution. $W_{t_{h,q}}$ is randomly selected from the current optimal $N_{well}(N_{well}=N_{R})$ individuals, and $W_{t_{g,q}}$ is randomly selected from the current less optimal $N_{good}(N_{good}=N-N_{C})$ individuals.

### 3.3.2. Improvement on evolution algorithm.

According to the location update equation, the individual in each subgroup searches for food around the cock, and the cock moves randomly. If the cock is trapped in the local optimum, the whole subgroup may also be trapped in it. In addition, there is no information exchange among the cocks, so the convergence speed is slow. In order to improve the convergence speed and global search ability of the algorithm and increase the traction effect of the optimal individual of the whole flock on the cock and hen, the corresponding position update formula of the improved cock is as follows:

$$W_{t_{p,q}}^{t+1} = W_{t_{p,q}}^t \ast [1 + \text{randn}(0, \sigma^2)] + \lambda_2 \ast (W_{t_{best,q}}^t - W_{t_{p,q}}^t)$$  \hspace{2cm} (20)

$$\lambda_2 = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \ast [2 - \exp(ln 2 \ast t / M)]$$  \hspace{2cm} (21)

$\lambda_2$ is the adaptive adjustment factor, $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum of the adjustment factor respectively. $M$ is the maximum number of iterations.

The updated equation of the hen’s position is as follows:

$$W_{t_{p,q}}^{t+1} = W_{t_{p,q}}^t + S_1 \ast \text{rand} \ast (W_{t_{r1,q}}^t - W_{t_{p,q}}^t) + S_2 \ast \text{rand} \ast (W_{t_{r2,q}}^t - W_{t_{r1,q}}^t) + \lambda_2 \ast (W_{t_{best,j,q}}^t - W_{t_{p,q}}^t)$$  \hspace{2cm} (22)

The steps of the improved algorithm are

1. Initialization: The initial parameters as $N$, $N_R$, $N_H$, $N_C$, $N_M$ and $M$ are determined. The initial positions of each individual in the solution space are randomly generated (when $t=0$). After the corresponding fitness are calculated, the best current position $E_{best}$ of the individual and the best overall position $P_{best}$ of the chicken flock are initialized;

2. If the remainder of the iteration order divided by the number of pre-determined iterations $G$ is 1, then the fitness values of chickens should be ranked, and the hierarchy, dominance and mother-child relationship of chickens should be established;

3. Update the position of cock, hen and chicken and calculate their fitness by Eq. (20), (22), (17) respectively;
(4) Update the current best position of the individual and the global best position of the chicken flock;

(5) Step (2) to (4) is an iteration process. After step(4) is completed, whether the maximum number of iterations has been reached should be determined. If it has been reached, the iteration will be terminated, otherwise, the iteration should continue.

3.4. Application of ICCSO algorithm for AFCN system in goethite process. It is assumed that the initial weights given by experts in the model are accurate, i.e., the initial weights $W_0$ of node interconnection are invariable. When the initial weight satisfies $\sqrt{\sum_{i=1}^{n} (||W_i||)^2} < 4$, the model achieves the condition of “fixed point” equilibrium state. The weights of the AFCN model obtained by iteration of Eq. (5) converge to the fixed point $W_{end}$.

At a certain time the concentration of Fe$^{2+}$ is 11.43 g/L, the concentration of Fe$^{3+}$ is 2.39 g/L and the pH value is 1.6 according to expert experience and historical data, the flow rate of O$_2$ is 30 m$^3$/h and the addition of calcine is 0.375 t/h. The state of the moment is normalized by the transition function $f = 1/(1+e^{-cx})$. After normalizing the initial state value at a certain time in the system, the state value $A$ in the AFCN model at that time can be obtained:

$$A_{\text{start}} = [0.667, 0.125, 0.58, 0.895, 0.833]$$ (23)

According to expert knowledge, the initial weight matrix of the AFCN for goethite process is:

$$W = \begin{bmatrix}
0 & 0 & -0.5621 & 0.494 & -0.268 \\
0 & 0 & 0.2656 & -0.1134 & -0.1768 \\
0 & 0 & 0 & -0.1134 & -0.2167 \\
0 & 0 & -0.084 & 0 & -0.2242 \\
0 & 0 & -0.2213 & 0.233 & 0 \\
\end{bmatrix}$$ (24)

4. Simulation and analysis.

4.1. Analysis on simulation of ICCSO algorithm. Seven test functions listed in Table 1 are used to analyze and verify the performance of the proposed ICCSO. The simulation results are compared with the improved chicken swarm algorithm ICSO[26] and the basic chicken swarm algorithm CSO. The test functions are adopt in the 2010 IEEE Congress on Evolutionary Computation.

In this experiment, the population size $N$ of ICCSO algorithm is 100, the dimension $D$ is 10, and the maximum iteration number $M$ is 1000. $N_R$, $N_H$, $N_C$ and $N_M$ are 0.2N, 0.6N, 0.1N and 0.1N respectively. FL is a random number between [0.4, 1]. $G = 10$, $\lambda_{min} = 0.8$, $\lambda_{max} = 1.0$. Each test function runs 50 times independently to eliminate randomness. The calculated optimum value, worst value, average value, standard deviation are compared with the calculated results of CSO and ICSO. The results are shown in Table 2.

As can be seen from Table 2, ICCSO algorithm achieves the more satisfactory results for five standard test functions, especially for the problems with many extreme points. For example, because of the existence of many extreme points in F2 function, both CSO algorithm and ICSO algorithm fall into local optimum, while ICCSO algorithm avoids falling into local optimum and achieves high convergence.
Table 1. Standard test functions for testing algorithm performance.

| Test Function            | Expression                                | Symbol | Range of values   | Value of optimal solution |
|--------------------------|-------------------------------------------|--------|-------------------|---------------------------|
| Sphere                   | \( f(x) = \sum_{i=1}^{n} x_i^2 \)         | F1     | [-100,100]        | 0                         |
| Rosenbrock               | \( f(x) = -a \cdot \exp(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) \) \( - \exp[\frac{1}{n} \sum_{i=1}^{n} \cos(c x_i)] + a + e \) | F2     | [-30,30]          | 0                         |
| High Conditioned Elliptic| \( f(x) = \sum_{i=1}^{n} a_i \cdot x_i^2 \) | F3     | [-100,100]        | 0                         |
| Bent Cigar               | \( f(x) = x_1^2 + \sum_{i=2}^{n} a x_i^2 \) | F4     | [-100,100]        | 0                         |
| Discus                   | \( f(x) = a x_1^2 + \sum_{i=2}^{n} x_i^2 \) | F5     | [-100,100]        | 0                         |
| Rotated hyper-ellipsoid  | \( f(x) = \sum_{i=1}^{n} \sum_{j=1}^{i} x_j^2 \) | F6     | [-100,100]        | 0                         |
| Rotated rastrigin        | \( f(x) = \sum_{i=2}^{n} (x_i^2 - a \cdot \cos 2\pi x_i + a) \) | F7     | [-100,100]        | 0                         |

Accuracy. The simulation results show that ICCSO algorithm outperforms CSO algorithm and ICSO algorithm in global search ability and convergence speed.

4.2. Analysis on simulation of CSO algorithm in goethite system with AFCN. When the time-lag is not 0, the final weights obtained by Eqs.(7) and (10) are updated online by CSO algorithm:

\[
W = \begin{bmatrix}
0 & 0 & -0.4483 & -0.1742 & -0.3944 \\
0 & 0 & 0.2516 & -0.3584 & -0.2092 \\
0 & 0 & 0 & -0.0692 & -0.2774 \\
0 & 0 & 0.1033 & 0 & 0.1557 \\
0 & 0 & -0.1772 & -0.6807 & 0
\end{bmatrix} \tag{25}
\]

The weight matrix W of Eq.(25) is substituted into the AFCN model, then the model state can be update independently. After 11 iterations, the AFCN model reaches the equilibrium point. At this time, the values of each node state are:

\[
A_{equil} = [0.6667 \ 0.125 \ 0.3786 \ 0.3588 \ 0.443] \tag{26}
\]

The specific process is shown in Figure 3.

The change of Fe\(^{2+}\) at the third node is mainly observed. From the simulation result of CSO algorithm in Eq.(26), it is seen that when the ACFCN model with time-lag reaches steady state, \( A_3 = 0.371 \), which is not close to the actual value \( A_3^{destination} = 0.36 \). It can be seen that CSO cannot accurately simulate the actual working conditions of the system.
Table 2. Comparison of test results of algorithms

| Title Symbol | Algorithms | Optimal value | Worst value | Average value | Standard deviation | Stable step |
|--------------|------------|---------------|-------------|---------------|--------------------|-------------|
|              |            | Symbol        |             |               |                    |             |
| F1           | CSO        | 1.8488e-133  | 2.8082e-123 | 6.0362e-125  | 3.9681e-124        | 30          |
|              | ICSO       | 7.0233e-133  | 6.4356e-125 | 3.1959e-126  | 1.1357e-125        | 26          |
|              | ICCSO      | 6.9244e-182  | 2.0424e-163 | 5.0084e-165  | 3.2075e-164        | 24          |
| F2           | CSO        | 6.1449        | 7.97        | 6.9651        | 0.3031             | 33          |
|              | ICSO       | 5.9715        | 7.2163      | 6.7664        | 0.3324             | 26          |
|              | ICCSO      | 2.1398e-07    | 4.9115e-05  | 6.6865e-06    | 8.2512e-06         | 27          |
| F3           | CSO        | 6.7415e-127   | 2.1961e-117 | 7.328e-119    | 3.3125e-118        | 30          |
|              | ICSO       | 9.006e-128    | 1.4093e-118 | 3.3872e-119   | 1.9906e-118        | 26          |
|              | ICCSO      | 8.9815e-177   | 1.2078e-160 | 2.4737e-162   | 1.7073e-161        | 24          |
| F4           | CSO        | 3.1473e-127   | 1.0419e-117 | 2.9859e-119   | 1.4796e-118        | 30          |
|              | ICSO       | 9.4786e-127   | 5.5614e-120 | 2.5308e-120   | 8.8336e-120        | 26          |
|              | ICCSO      | 5.148e-178    | 5.2831e-176 | 1.1419e-176   | 7.372e-176         | 24          |
| F5           | CSO        | 6.1475e-131   | 9.9848e-123 | 5.2391e-124   | 1.7265e-123        | 30          |
|              | ICSO       | 9.8894e-132   | 2.2241e-125 | 4.6208e-125   | 3.1431e-124        | 26          |
|              | ICCSO      | 4.9026e-181   | 1.0179e-166 | 5.8305e-168   | 2.5381e-167        | 24          |
| F6           | CSO        | 5.4115e-127   | 1.5173e-109 | 6.328e-109    | 3.3225e-107        | 30          |
|              | ICSO       | 8.016e-138    | 1.9214e-140 | 4.6672e-140   | 2.1066e-139        | 26          |
|              | ICCSO      | 9.148e-165    | 1.3078e-187 | 2.6728e-187   | 1.7953e-187        | 24          |
| F7           | CSO        | 4.1923e-172   | 1.0419e-117 | 2.1659e-119   | 1.4707e-118        | 30          |
|              | ICSO       | 9.4554e-125   | 4.6634e-122 | 2.5325e-122   | 7.7543e-122        | 26          |
|              | ICCSO      | 6.1579e-148   | 5.2635e-177 | 1.9719e-172   | 6.3823e-172        | 24          |

4.3. Analysis on simulation of ICCSO algorithm in goethite system with AFCN. When the time-lag is not zero, the final weights obtained by using Eqs.(7) and (10) and online learning and updating with ICCSO algorithm are obtained:

\[
W = \begin{bmatrix}
0 & 0 & -0.4494 & -0.1782 & -0.3999 \\
0 & 0 & 0.2666 & -0.35 & -0.1976 \\
0 & 0 & -0.0691 & -0.27 & 0 \\
0 & 0 & 0.1046 & 0 & 0.1579 \\
0 & 0 & -0.1706 & -0.6898 & 0
\end{bmatrix}
\]
The weight matrix $W$ of Eq.(27) is substituted into the AFCN model, and the state of the model is updated independently. After 9 iterations, the AFCN model reaches the equilibrium point, at which time the state of each node is taken as the value:

$$A_{equil} = [0.6667 \ 0.125 \ 0.3559 \ 0.3451 \ 0.4449]$$ \hspace{1cm} (28)

The specific process is shown in Figure 4.

It can be clearly seen that the simulation result of ICCSO algorithm is $A_3 = 0.3559$, which is very close to the actual working condition value $A_{destination} = 0.36$. The simulation results well reflect the change process of iron ion concentration in the goethite process. It can be seen that the AFCN model using ICCSO algorithm to determine the weights can reflect the expected stabilization target of concentration under the condition of system stability, and the control effect is very ideal. This method is suitable for industrial system modeling with time-lag.

4.4. Simulation experiment analysis of ICCSO algorithm in goethite precipitation system with FCN. When using FCN model without time lag, the final weights obtained by using Eqs.(7) and (10) and on-line learning and updating with ICCSO algorithm are obtained:

$$W = \begin{bmatrix}
0 & 0 & -1 & -0.3842 & -0.7244 \\
0 & 0 & -0.2516 & -0.4084 & -0.2392 \\
0 & 0 & 0 & -0.5012 & -0.4774 \\
0 & 0 & -0.0841 & 0 & 0.0613 \\
0 & 0 & -0.472 & -1 & 0
\end{bmatrix}$$ \hspace{1cm} (29)

The weight matrix $W$ of eq.(29) is substituted into the FCN model, and the state of the model is updated independently. After 9 iterations, the FCN model
reaches the equilibrium point, at this moment, the state value of each node is taken as follows:

\[
A^{equil} = [0.6667 \ 0.125 \ 0.3816 \ 0.3798 \ 0.4497]
\] (30)

The specific process is shown in Figure 5.

It can be clearly seen that the simulation result of ICCSO algorithm is \( A^3 \) = 0.3816, which is not close to the actual working condition value \( A^3_{\text{destination}} = 0.36 \). Apparently, the FCN model which uses ICCSO to determine the weight that can not reflect the actual working conditions of the system. Furthermore, its control effect is not satisfactory.

4.5. **Comparison on simulations of ICCSO, GA and PSOA in goethite precipitation system.** As shown in Figure 6, the simulation validates the final results of Fe\(^{2+}\). \( A_3 \) is obtained by using ICCSO, GA and PSOA weight learning algorithms respectively. From the results, it can be seen that ICCSO is stable at step 9, GA is stable at step 10, and PSOA is stable at step 13. And the final result is closer to the real value of the system. This horizontal comparison shows the superiority of ICCSO in optimization algorithm. The results reveals that the algorithm being proposed has superiority both in accuracy and efficiency, also it is more stable. The comparison of results indicates the advantage of ICCSO.

4.6. **Various errors verify the stability of AFCN modeling in goethite system.** There was only one kind of working condition of goethite precipitation system which is used above. To enhance the persuasion and prove stability and practicability, various working conditions were added. According to the distribution of Fe\(^{2+}\) concentration, Fe\(^{3+}\) concentration and pH in a smelter, Fe\(^{2+}\) concentration can be divided into three types: small (S), medium (M) and large (B); Fe\(^{3+}\) concentration...
Figure 5. FCN simulation after ICCSO learning

Figure 6. Comparison of ICCSO, GA and PSOA
can be divided into two types: small (S) and large (B); and pH into two types: small (S) and large (B). So there are 12 combinations. Based on these working conditions, the important index of iron sinking process is export Fe\(^{2+}\) concentration, namely \(A_3\). Its root mean square error (RMSE), mean absolute error (MAE), maximum absolute error (MAX) and standard deviation (SD) are analyzed as shown in Table 4.

| Working conditions | \(A_3\) | RMSE  | MAE   | MAX   | SD    |
|-------------------|--------|-------|-------|-------|-------|
| SSS               | 0.1400 | 0.1421| 0.0013| 0.0023| 0.0011|
| SSB               | 0.1500 | 0.1490|       |       |       |
| SBS               | 0.1518 | 0.1533|       |       |       |
| SBB               | 0.1177 | 0.1199|       |       |       |
| MSS               | 0.2900 | 0.2906|       |       |       |
| MSB               | 0.2694 | 0.2689|       |       |       |
| MBS               | 0.3659 | 0.3659|       |       |       |
| MBB               | 0.0100 | 0.0120|       |       |       |
| BSS               | 0.1620 | 0.1627|       |       |       |
| BSB               | 0.1997 | 0.2062|       |       |       |
| BBS               | 0.3400 | 0.3407|       |       |       |
| BBB               | 0.3290 | 0.3189|       |       |       |

It can be seen that the deviation between the simulation results of the AFCN model established by the method in this paper and the actual target values of the system is very small, which indicates that the AFCN model can be applied to the goethite precipitation system stably.

5. **Conclusions.** A modeling method based on asynchronous fuzzy cognitive network model for nonlinear systems were proposed. By combining the modeling method of fuzzy cognitive network with the asynchronous time-lag system, the node state and weight of the fuzzy cognitive network were extended to the time interval. The simulation results showed that the proposed modeling method had wide applicability for uncertain and time-lag systems.

For the first time, the basic chicken swarm algorithm is improved on two aspects of constraint function and evolutionary mechanism. The basic chicken swarm algorithm was improved from aspects of constraint function and evolutionary mechanism for the first time. The improved chicken swarm algorithm was used to identify the model parameters of AFCN. Compared with CSO algorithm, the experimental results showed that the learning results of the proposed ICCSO in this paper were
accurate and independent from the initial values set by experts. It could also avoid the local optimum that general optimization algorithms like GA and PSOA would fall into. Additionally, the global search ability, convergence accuracy and speed of ICCSO algorithm were higher than those of CSO algorithm.

**Funding:** This research was funded by the Program of National Science Foundation of China, grant number 61673339 and the Program of National Science Foundation of Hunan Province, grant number 2017JJ2329.

**Conflicts of Interest:** The authors declare no conflict of interest.

**REFERENCES**

[1] A. P. Antigoni and P. P. Groumpos, Modeling of parkinson’s disease using fuzzy cognitive maps and non-linear hebbian learning, *International Journal on Artificial Intelligence Tools*, 23 (2014), 1450010.

[2] N. Chen, J. Y. Dai, X. J. Zhou, Q. Q. Yang and W. H. Gui, Distributed model predictive control of iron precipitation process by goethite based on dual iterative method, *International Journal of Control Automation and Systems*, 17 (2019), 1233–1245.

[3] N. Chen, J. Y. Dai, W. H. Gui, Y. Q. Guo and J. Q. Zhou, A hybrid prediction model with a selectively updating strategy for iron removal process in zinc hydrometallurgy, *Science China Information Sciences*, 63 (2020), 119205.

[4] N. Chen, Y. Fan, W. H. Gui, C. H. Yang and Z. H. Jiang, Hybrid modeling and control of iron precipitation by goethite process, *Chinese Journal of Nonferrous Metals*, 24 (2014), 254–261.

[5] B. Christen, C. Kjeldsen, T. Dalgaard and J. Martin-Ortega, Can fuzzy cognitive mapping help in agricultural policy design and communication?, *Land Use Policy*, 45 (2015), 64–75.

[6] N. Chen, J. Q. Zhou, J. J. Peng, W. H. Gui and J. Y. Dai, Modeling of goethite iron precipitation process based on time-delay fuzzy gray cognitive network, *Journal of Central South University*, 26 (2019), 63–74.

[7] N. Chen, J. J. Peng, L. Wang, Y. Q. Guo and W. H. Gui, Fuzzy grey cognitive networks modeling and its application, *Acta Automatica Sinica*, 44 (2018), 1227–1236.

[8] N. Chen, L. Wang, J. J. Peng, B. Liu and W. H. Gui, Improved nonlinear Hebbian learning algorithm based on fuzzy cognitive networks model, *Control Theory and Applications*, 33 (2016), 1277–1280.

[9] Y. G. Deng, Q. Y. Chen, Z. L. Yin and P. M. Zhang, Iron removal from zine leaching solution by goethite method, *Non-ferrous Metal*, 62 (2014), 80–84.

[10] Z. Djaafar, A. Yahia and N. Farid, Multi-objective chicken swarm optimization: A novel algorithm for solving multi-objective optimization problems, *Computers and Industrial Engineering*, 129 (2019), 377–391.

[11] S. Fatahi and H. Moradi, A fuzzy cognitive map model to calculate a user’s desirability based on personality in e-learning environments, *Computers in Human Behavior*, 63 (2016), 272–281.

[12] B. Kosko, Fuzzy cognitive maps, *International Journal of Man-Machine Studies*, 24 (1986), 65–75.

[13] V. Kreinovich and C. D. Styllos, Why fuzzy cognitive maps are efficient, *International Journal of Computers Communications & Control*, 10 (2015), 825–833.

[14] T. Kottas, D. Stimoniaris and D. Tsiamitros, New operation scheme and control of Smart Grids using Fuzzy Cognitive Networks, *PowerTech, 2015 IEEE Eindhoven*, 63 (2015), 1–5.

[15] D. B. Li and J. M. Jiang, Present situation and development trend of zinc smelting technology at home and abroad, *China Metal Bulletin*, 6 (2015), 41–44.

[16] P. C. Marchal, J. G. García and J. G. Ortega, Application of fuzzy cognitive maps and run-to-run control to a decision support system for global set-point determination, *IEEE Transactions on Systems Man & Cybernetics Systems*, 47 (2017), 2256–2267.

[17] A. Mourhir, E. I. Papageorgiou, K. Kokkinos and T. Rachidi, Exploring precision farming scenarios using Fuzzy Cognitive Maps, *Sustainability*, 9 7 (2017), 1241.

[18] X. B. Meng, Y. Liu and X. Z. Gao, A new bio-inspired algorithm: Chicken swarm optimization, *Proc of International Conference in Swarm of Intelligence, Cham: Springer*, (2014), 86–94.

[19] M. Obiedat and S. Samarasinghe, A novel semi-quantitative Fuzzy Cognitive Map model for complex systems for addressing challenging participatory real life problems, *Applied Soft Computing*, 48 (2016), 91–110.
[20] E. I. Papageorgiou, K. D. Aggelopoulou and T. A. Gemtos, Yield prediction in apples using Fuzzy Cognitive Map learning approach, *Computers & Electronics in Agriculture*, 91 (2013), 19–29.

[21] K. E. Parsopoulos, E. I. Papagergiou, P. P. Groumpos and M. N. Vrahatis, A first study of fuzzy cognitive maps learning using particle swarm optimization, *Proceedings of IEEE Congress on Evolutionary Computation 2003*, (2003), 1440–1447.

[22] J. Solana-Gutiérrez, G. Rincón, C. Alonso and D. García-de-Jalón, Using fuzzy cognitive maps for predicting river management responses: A case study of the Esla River basin, Spain, *Ecological Modelling*, 360 (2017), 260–269.

[23] W. Stach, L. Kurgan, W. Pedrycz and M. Reformat, Genetic learning of fuzzy cognitive maps, *Fuzzy Sets and Systems*, 153 (2005), 371–401.

[24] D. H. Wu, S. P. Xu and F. Kong, Convergence analysis and improvement of the chicken swarm optimization algorithm, *IEEE Access*, 4 (2019), 9400–9412.

[25] B. Wang, W. Li, X. H. Chen and H. H. Chen, Improved chicken swarm algorithms based on chaos theory and its application in wind power interval prediction, *Mathematical Problems in Engineering*, (2019), Art. ID 1240717, 10 pp.

[26] Z. Q. Wu, D. Q. Yu and X. H. Kang, Application of improved chicken swarm optimization for MPPT in photovoltaic system, *Optimal Control Applications and Method*, 39 (2018), 1029–1042.

[27] X. W. Yu, L. X. Zhou and X. Y. Li, A novel hybrid localization scheme for deep mine based on wheel graph and chicken swarm optimization, *Computer Networks*, 154 (2019), 73–78.

[28] Y. L. Zhang, *Modeling and Control of Dynamic System Based on Fuzzy Cognitive Maps*, Dalian University of Technology, 2012.

Received June 2019; revised August 2019.

E-mail address: yuzoudiyi@126.com
E-mail address: ningchen@csu.edu.cn
E-mail address: daijiayang@csu.edu.cn
E-mail address: gwh@csu.edu.cn