Available online at www.sciencedirect.com

SciVerse ScienceDirect

Journal of the Franklin Institute ▮ (▮▮▮▮) ▮▮▮–▮▮▮

ELSEVIER

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

Modeling of pH neutralization process using fuzzy recurrent neural network and DNA based NSGA-II

Xiao chen*, Anke Xue, Dongliang Peng, Yunfei Guo

Department of Automation, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China

Received 31 August 2012; received in revised form 7 December 2012; accepted 18 March 2013

Abstract

In this paper, the Takagi–Sugeno fuzzy recurrent neural network (T–S FRNN) is applied to model a pH neutralization process. Since the accuracy and complexity of the network are two contradictory criteria for the T–S FRNN model, a DNA based NSGA-II is proposed to optimize the parameters of the model. In the DNA based NSGA-II, each individual is encoded with one nucleotide base sequence, modified DNA based crossover and mutation operators are designed to improve the searching ability of the algorithm, and crowding tournament selection is applied based on the Pareto-optimal fitness and the crowding distance. The study on the performance of test functions shows that the DNA based NSGA-II outperforms NSGA-II in the quality of the obtained Pareto-optimal solution. To verify the effectiveness of the established T–S FRNN model for the pH neutralization process, it is compared with two T–S FRNN models optimized with other methods. Comparison results show that the model optimized by DNA based NSGA-II is more accurate and the complexity of the network is acceptable.

1. Introduction

The control of pH neutralization process is common in many chemical processes such as biotechnology processing and waste treatment [1–4]. However, high performance and robust pH control can be difficult to achieve due to the inherent nonlinearity characteristics of the process. In recent years, several nonlinear control strategies have been employed into the control of pH neutralization process [2–4]. Although these pH control techniques are different in many ways, most of them require a dynamic model of the process. Hence, in order to get high control

*Corresponding author. Tel.: +86 15068102725.
E-mail addresses: chenxiao@hdu.edu.cn, micxchen@sohu.com (X. chen).

0016-0032/$32.00 © 2013 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jfranklin.2013.03.014

performance, a dynamic model which can adequately address the characteristics of the process is important. However, since the pH process exhibits severe nonlinearity, the process model is hard to obtain.

In the previous study, intelligent techniques including neural networks and fuzzy system have been used to model and control the complex nonlinear systems for their excellent ability to approximate the nonlinear system [5,6]. Abdalla and Hawileh [7] used a Radial Basis Function (RBF) artificial neural network to predict the low-cycle fatigue life of steel reinforcing bars. Switched circuits are modeled based on wavelet decomposition and neural network [8]. Li et al. [9] designed a fuzzy H_∞ controller for a quarter-vehicle suspension model with actuator delay and fault. Recently, the synergy of these two paradigms has given rise to a rapidly emerging field, neuro-fuzzy network [10–12]. It intends to combine the semantic transparency of rule-based fuzzy systems with the learning capability of neural networks. Since it can capture the advantages of fuzzy logic and neural networks, there are many applications of the neuro-fuzzy network in the identification of industrial nonlinear systems. Salahshoor et al. [13] designed an adaptive neuro-fuzzy modeling approach to identify an online affine-type model. Sadeghian and Fatehi [14] used the locally linear neuro-fuzzy technique to identify and predict the process fault in a cement rotary kiln. Scherer [15] proposed a relational neuro-fuzzy system for machine learning systems. Among all different types of the neuro-fuzzy models, Takagi–Sugeno (T–S) fuzzy recurrent neural network (FRNN) stands out due to its internal dynamic network structure and has been employed in the modeling of the complex nonlinear system [16,17]. Baruch et al. [18] applied hierarchical fuzzy-neural multi-model and T–S rules with recurrent neural procedural consequent part for system identification, state estimation and adaptive control of complex nonlinear plants. Nie et al. [19] identified the pH neutralization process with fuzzy-neural approaches.

As a neuro-fuzzy network, accuracy and complexity of the T–S FRNN are two factors that should be considered when we evaluate the performance of the network. However, the accuracy and the complexity of the network tend to be contradictory in nature. Moreover, different practical condition needs different trade-off between those two characteristics. Thus, how to find a network which can attain the desired trade-off is important. Usually, the precision of the T–S FRNN depends on the decision of the network parameters like the center point and width of the membership function while the complexity is usually affected by other network parameters like the number of the fuzzy rules. Hence, through optimization those network parameters, the decision of network could be represented as a multi-objective problem and be solved by multi-objective optimal algorithms (MOEAs). Among many MOEAs, Non-Dominated Sorting Genetic Algorithm (NSGA-II) proposed by Deb et al. [20] is the most well-known algorithm and has been applied into many fuzzy rule based systems [21,22]. It is well known that global searching ability is essential for NSGA-II because it makes the algorithm explore the working space and find a set of solutions which can be a good approximation of the Pareto optimal front. Thus, in order to improve the performance of NSGA-II, this paper presents a new DNA (Deoxyribonucleic acid) based NSGA-II. This idea is inspired by the recent study of DNA based genetic algorithm (GA) which introduces mechanism of the biological DNA into GA. With the specific encoding method and genetic operators, the global searching speed of traditional GA can be largely improved [23–26].

Hence, inspired by the success of DNA based GA, a DNA based NSGA-II is proposed to optimize the parameters of the T–S FRNN for the modeling of the pH neutralization process. In this study, every individual in the DNA based NSGA-II represents a premise part of one network whose corresponding consequent part is obtained by the recursive least square (RLS) algorithm. The individual is evaluated on the base of Pareto-dominance relation and crowding measure. Then modified genetic operators are applied to improve the global search ability of the algorithm. Elitism is also implemented in the algorithm based on the Pareto-dominance relation and the crowding measure. To validate the performance of the constructed model, it is compared with other reported models based on the train data and the test data.

2. T–S fuzzy neural network model

A SISO T–S FRNN is shown in Fig. 1.

In Fig. 1, it can be seen that the network is divided into two major parts: the premise part and the consequent part. In T–S FRNN, fuzzy rule R^j is described as follows:

$$R^j: \text{ if } x_1(k) \text{ is } A_j, \text{ then } y_j(k+1) = B_j^T X(k), \quad j = 1, 2, \cdots, M$$

where $X(k) = [1, u(k-d), \cdots, u(k-d-m+1), y(k-1), \cdots y(k-n)]$, u is the input of the actual system, y is the output of the actual system, y_j is the Rule j output of the T–S FRNN, B_j is the parameter of the consequent part, and $j = 1, 2, \cdots, M$, M is the number of the fuzzy rules, A_j is fuzzy set. In this paper, we set $x_1(k) = y(k)$ here.

The premise part of the T–S FRNN can be divided into four layers.

Layer 1: Each node in layer 1 transmits input values to layer 2.

Layer 2: Each node in layer 2 represents a Gaussian membership function:

$$\mu_j(x_1) = \exp\left[-\frac{\|x_1 - c_j\|^2}{\sigma_j^2}\right], j = 1, 2, \cdots, M \tag{1}$$

where μ_j is the jth Gaussian membership function, c_j is the center of the jth Gaussian membership function, and σ_j is the width of the jth Gaussian membership function.

Layer 3: Each node indicates a possible IF-part for fuzzy rules.

$$\alpha_j = \mu_j, \quad j = 1, 2, \cdots M \tag{2}$$

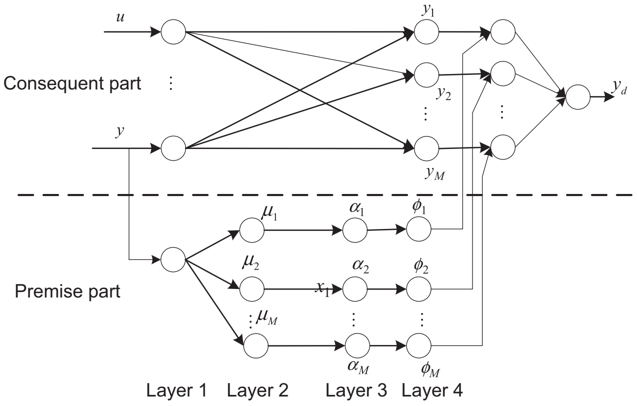

Fig. 1. Schematic diagram of SISO T–S FRNN.

Layer 4: The nodes in this layer realize the normalization. The output of each node is

$$\phi_j = \frac{\alpha_j}{\sum_{l=1}^{M} \alpha_l} \tag{3}$$

The consequent part can be described as recurrent neurons shown in Fig. 2.

Through defuzzier, the final output of the T–S FRNN is

$$y_d(k+1) = \sum_{j=1}^{M} \phi_j B_j^T X(k) \tag{4}$$

From the above description, we can find that there are several parameters need to be determined: c_j, σ_j and B_j. In this paper, we use DNA based NSGA-II to optimize the premise part parameters c_j and σ_j while the parameters in the consequent part are determined by the recursive least square (RLS) algorithm.

2.1. DNA based NSGA-II

2.1.1. Optimal objectives

As mentioned above, accuracy and complexity are two conflicting factors of one network. Generally, the accuracy of the network can be evaluated by the error between the actual system output and the model predicted output. Here, we define the error as the following form:

$$E_i = \sum_{j=1}^{Ns} (y - y_d)^2 \tag{5}$$

As for the complexity of the T–S FRNN model, it is determined by three factors: the number of the input of the premise part of the fuzzy rules, the number of the fuzzy rules and the schematic diagram of the consequent part. Since the first and the third factors are both predefined and fixed, the complexity of the network is determined by the number of the fuzzy rules.

Based on the above discussion, we define the optimal objectives as below:

$$f_1 = E \tag{6}$$

$$f_2 = M \tag{7}$$

2.1.2. Decoding method

In the view of modern biology, DNA is the major genetic material for life and encodes plentiful genetic information. The basic subunits of DNA are nucleotides. Due to their different chemical structure, nucleotides can be classified into four bases: adenine (A), guanine (G), cytosine (C), and thymine (T). These nucleotide bases can be bonded based on the Watson–Crick

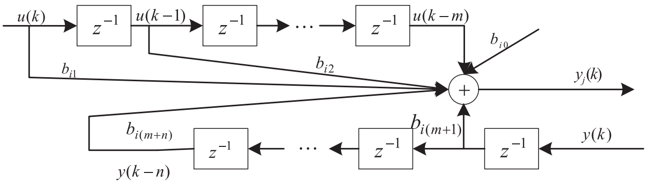

Fig. 2. Schematic diagram of dynamic neuron.

Please cite this article as: X. chen, et al., Modeling of pH neutralization process using fuzzy recurrent neural network and DNA based NSGA-II, Journal of the Franklin Institute. (2013), http://dx.doi.org/10.1016/j.jfranklin.2013.03.014

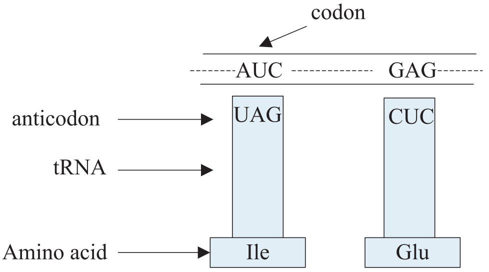

Fig. 3. Genetic information transmission.

complementary principle that A or G of one strand is always bonded to T or C of anther strand. Through this complementary property, a codon specified by three bases can be bonded with the specific anticodon consisting of three complementary bases of the codon on transfer RNA (tRNA), and assists subsequent transmission of genetic information in the formation of a specific amino acid, as shown in Fig. 3.

Based on the biological DNA structures, we can design a DNA encoding method for practical problems. In this work, every variable x_i is represented as a string consisting of a combination of four nucleotide bases, A, G, C, T. This means we have a four-letter alphabet $\sum \{A,G,C,T\}$ to encode potential solutions. Since such string cannot be possessed by digital computers, these bases are encoded with digital numbers. Here, integers 0, 1, 2, and 3 are adopted to encode the bases since they could represent the characteristics of bases, such as structure, function group and complementary relationship. And the mapping from nucleotide base to the digital integer is 0123/CGAT, which means that C is encoded with 0, G with 1, A with 2, and T with 3.

Then, every variable x_i is represented as an integer string of length l. The lower limit $x_{\min\ i}$ is represented by the decoded integer 0, and the upper limit $x_{\max\ i}$ is represented by the decoded integer 4^l-1. And the length of one individual is $L=n\times l$. Based on this DNA encoding method, we can introduce features of the biological DNA into the NSGA-II and develop DNA based NSGA-II.

2.1.3. Crossover operator

There are two types of crossover operation in the DNA based NSGA-II. These crossover operations are different from the traditional crossover that the parents exchange the anticodon of the selected segment instead of the original segment itself. The details of the crossover operations are described as below.

2.1.3.1. Crossover A. Like the traditional crossover, in this operation, we randomly select two individuals from the population as parents. Then, every parent is separated into n parts, and two parts in the same location of two parents are mated as a pair of sub-parents. Such separation is convenient for the population with different length individuals. For example, in this paper, each individual represents the premise part of one network that may contain different fuzzy rules, thus the number of unknown parameters is different and then the length of each individual could be different too. After the sub-parents are chosen, we select two crossover points at each sub-parent and the segments between the crossover points are defined as codons. Different from the codon in biological DNA, the number of the bases in a codon of the DNA based NSGA-II is not fixed. The number of the bases and the location of codon are both assigned randomly. Based on the Watson–Crick complementary principle, the anticodon of each codon is obtained. Finally, the

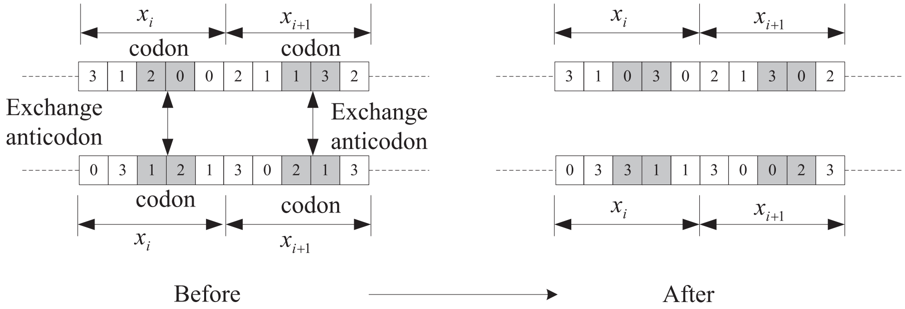

Fig. 4. An example of crossover A.

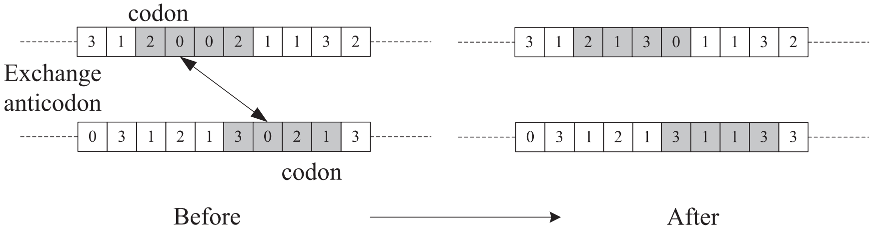

Fig. 5. An example of crossover B.

sub-parents exchange the aniticodon instead of the original segment. This operation is adopted with probability p_{c1}. An example is shown in Fig. 4.

2.1.3.2. Crossover B. In this operation, we first randomly choose two individuals in the population as parents. Then in each parent, we randomly select some consecutive bases as codon. The number of the bases is randomly assigned. Notice that, unlike crossover A, the location of two codons in each parent may not be same. Then, based on the Watson–Crick complementary principle, the anticodons consist of bases which are complementary to the codons are achieved. Finally, two parents exchange anticodons and two new individuals are obtained. This operation is adopted with probability p_{c1}. An example is shown in Fig. 5.

2.1.4. Mutation operation

Mutation is another important operation in the DNA based NSGA-II. In this paper, three types of mutation are used with different probabilities. The details of the crossover operations are described as below.

2.1.4.1. Inverse-Anticodon operator (IA operator). In the IA operator, an individual mutates through replacing the codon with the inversed corresponding anticodon. First, the individual is also separated into n parts and some consecutive bases are chosen as a codon in each part. Then, the inverse corresponding anticodon is obtained by inverting the order of the bases order in the corresponding anticodon. Finally, the codon is replaced by its inverse anticodon. For example, in Fig. 6, if the codon is selected as 112 (GGA), its anticodon will be 003 (CCT) and its inverse anticodon is 300 (TCC). Then, the bases 112 are replaced with 300. IA operator occurs with the probability p_{IA}.

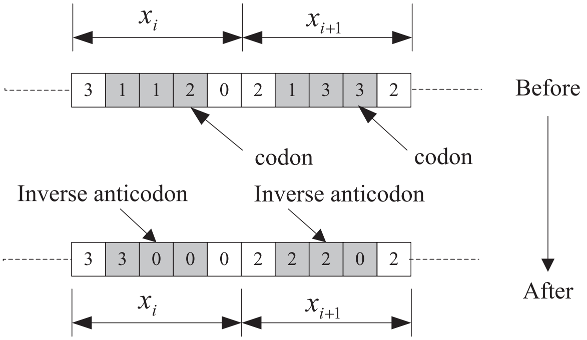

Fig. 6. An example of IA operator.

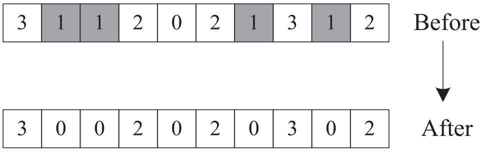

Fig. 7. An example of MM operator.

2.1.4.2. Maximum–Minimum operator (MM operator). In the MM operation, the chromosome mutates by replacing the frequently used base with the rarely used base in the current chromosome. Notice that, like crossover B, chromosome does not require to be separated into several parts in the MM operation. One example of MM operator is shown in Fig. 7. In Fig. 7, it is clear that base G (1) is the most frequently used base in the chromosome, while base C (0) is the least frequently used base. Then, MM operator replaces every base G with base C. MM operator occurs with the probability p_{MM}.

2.1.4.3. Mutation operator. Mutation is a background operator which produces spontaneous random changes in the chromosomes. In the DNA–GA, every base in the individual can be replaced by one of another three bases with the probability p_m. Fig. 8 gives an example of mutation where base A is replaced with base C.

2.1.5. Fitness assignment and selection

In the DNA based NSGA-II, the fitness assignment and selection of NSGA-II is adopted. For each individual in the population, it is compared with every other individual and checked whether it could be marked as dominated. Otherwise, the individual is marked as non-dominated. Here, all the non-dominated individuals are assigned rank 1. Then the remaining individuals are compared again and the non-dominated individuals are assigned rank 2. Through repeating the above process, each individual is designated a rank level.

After that, the crowded tournament selection is implemented based on the crowding distance of each individual. Here, the crowding distance of one individual is defined as the average distance of two individuals on either side of this individual along each of the objective function. In the selection operation, we first randomly select two individuals from the population, and pick

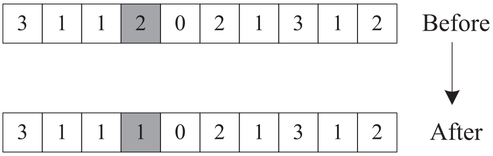

Fig. 8. An example of mutation operator.

up the one with higher rank if they belong to different rank. Otherwise, the one with better crowding distance would be selected.

2.1.6. *Procedure of DNA based NSGA-II*

Based on the above description, the procedure of the DNA based NSGA-II can be summarized as shown in Fig. 9.

3. Computational tests

To illustrate the effectiveness of the proposed DNA based NSGA-II, we use four typical multi-objective optimal functions as test problems to test the performance of the algorithm compared with the performance of NSGA-II. All the Pareto-optimal fronts of the four problems are non-consecutive, non-convex and deceptive which are difficult for the optimal algorithm. The details of the functions are shown in Table 1.

Generally, there are two types of indices to evaluate the solutions of the multi-objective problems: convergence to the Pareto-optimum and the diversity of the Pareto-optimum. In this section, we use three performance indices to describe the convergence of the algorithm and the diversity and the contribution of the solutions.

(1) Distance error (DE): DE is used to represent the distance between the optimized pareto-optimal front (PF_{op}) and the known pareto-optimal front (PF_{kn}).

$$DE = \left(\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{1/2} \tag{8}$$

where n_{PF} is the number of the solution in the PF_{op} and d_i is the minimum distance among the distances between the solution in PF_{op} and the solution in PF_{kn}. A smaller DE means that the optimized front is more convergent to the real pareto-optimal front.

(2) Even spacing (ES): ES is used to represent the uniformity of the distribution of the solutions in PF_{op}

$$ES = \frac{\left[\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} (d_i' - \overline{d}')^2 \right]^{1/2}}{\overline{d}'}, \quad \overline{d}' = \frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i' \tag{9}$$

where d_i' is the Euclidean distance between the ith solution and its nearest solution in PF_{op}. Smaller ES means that the distribution of the solution in the Pareto-optimal front is more even.

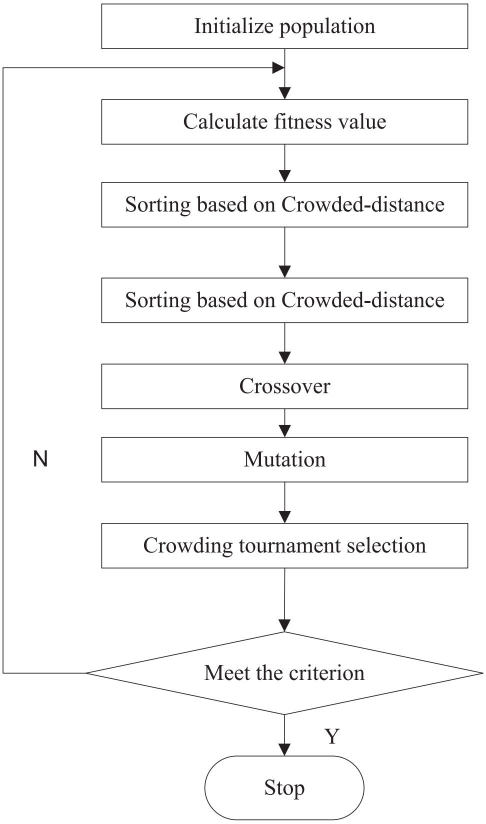
Initialize population
Calculate fitness value
Sorting based on Crowded-distance
Sorting based on Crowded-distance
Crossover
N
Mutation
Crowding tournament selection
Meet the criterion
Y
Stop

Fig. 9. Procedure of DNA based NSGA-II.

(3) Maximum Spread (MS): MS represents the degree that PF_{kn} is covered by PF_{op}.

$$MS = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left\{\frac{[\min(f_i^{\max}, F_i^{\max}) - \max(f_i^{\min}, F_i^{\min})]}{F_i^{\max} - F_i^{\min}}\right\}^2} \tag{10}$$

where $f_i^{\max}$ and $f_i^{\min}$ is the maximum and minimum of the ith objective function in PF_{op} separately, and $F_i^{\max}$ and $F_i^{\min}$ is the maximum and minimum of the ith objective function in PF_{kn} separately. If MS is 1, the optimized Pareto-optimal front can completely cover all the optimum of the objective functions.

We use DNA based NSGA-II and NSGA-II to optimize the four test problems, and obtain the corresponding Pareto-optimal front when the algorithm evolves to 1000 generation which is shown from Figs. 10–13. The indices of two algorithms of four test functions are shown in Table 2.

The degree that the obtained solutions could converge to the known Pareto-optimal front is the most important for the multi-objective optimization methods. From the description of the Pareto-optimal solution, if the final solutions obtained by one method are closer to the known Pareto-optimal front compared with the solutions obtained by other methods, it means this method possesses better performance than the compared methods. Then, the other factors about the solutions' quality would be compared, like the indices in Table 2. Thus, we would first analyze

Please cite this article as: X. chen, et al., Modeling of pH neutralization process using fuzzy recurrent neural network and DNA based NSGA-II, Journal of the Franklin Institute. (2013), http://dx.doi.org/10.1016/j.jfranklin.2013.03.014

Table 1
Test functions.

DEB	$\min f_1(\mathbf{x}) = x_1$ $\min f_2(\mathbf{x}) = g/f_1(\mathbf{x})$ $g = 2 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\}$	$x_i \in [0.1, 1]$ $i = 1, 2$
KUR	$\min f_1(\mathbf{x}) = \sum_{i=1}^{2}\left[1 - 10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})\right]$ $\min f_2(\mathbf{x}) = \sum_{i=1}^{3}\left[\lvert x_i\rvert^{0.8} + 5\sin(x_i^3)\right]$	$x_i \in [-5, 5]$ $i = 1, 2, 3$
ZDT3	$\min f_1(\mathbf{x}) = x_1$ $\min f_2(\mathbf{x}) = g\left[1 - (x_1/g)^{1/2} - x_1\sin(10\pi x_1)/g\right]$ $g = 1 + 9\left(\sum_{i=2}^{n} x_i/(n-1)\right)$	$x_i \in [0, 1]$ $i = 1, 2, \cdots, 30$
ZDT6	$\min f_1(\mathbf{x}) = x_1$ $\min f_2(\mathbf{x}) = g[1 - (x_1/g)^{1/2}]$ $g = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, \cdots, 10$

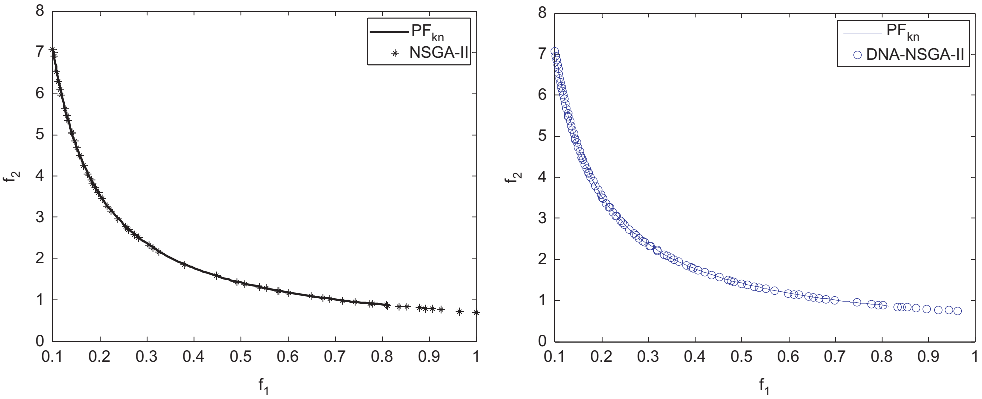

Fig. 10. Pareto-optimal front obtained by NSGA-II and DNA-NSGA-II for DEB.

the final optimal front obtained by DNA based NSGA-II and NSGA-II through Figs. 10–13. In test problem DEB as shown in Fig. 10, both methods perform well that the obtained optimal fronts can converge to the known Pareto-optimal front. They also perform similarly in test problem ZDT6 that most of the obtained solutions belong to the known Pareto-optimum while some ones are not as shown in Fig. 13. However, DNA based NSGA-II shows its advantage in KUR and ZDT3 that both of its optimal fronts can converge to the known Pareto-optimal front while the optimal fronts of NSGA-II not. That difference is more obvious in test problem ZDT3 shown in Fig. 12.

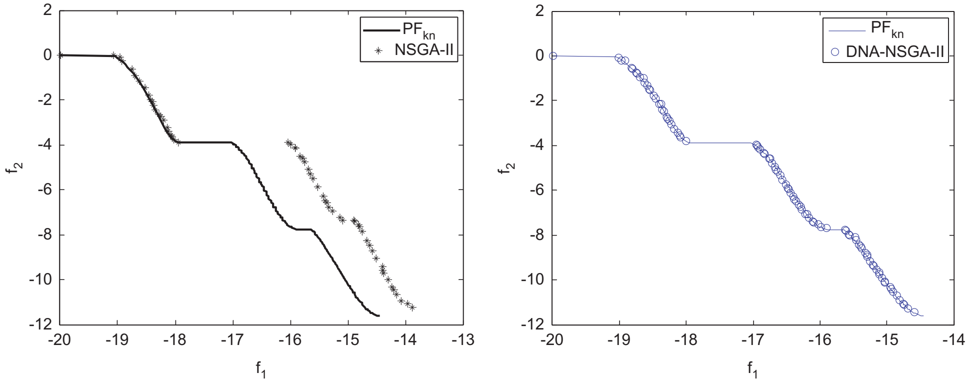

Fig. 11. Pareto-optimal front obtained by NSGA-II and DNA-NSGA-II for KUR.

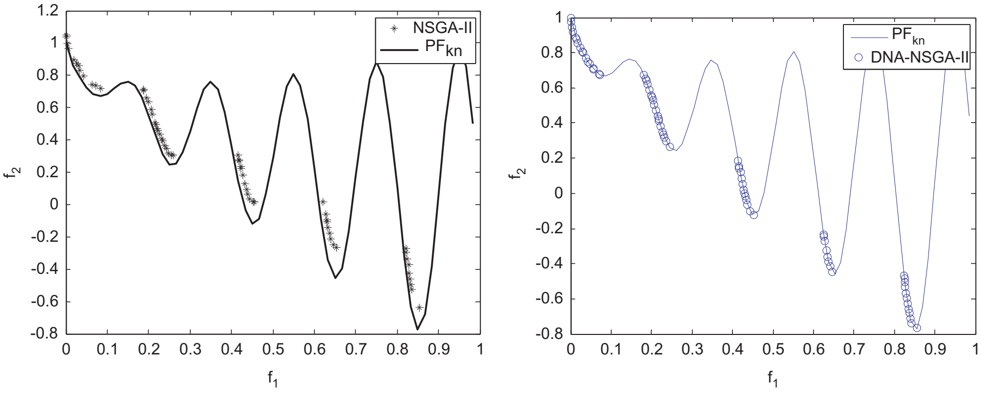

Fig. 12. Pareto-optimal front obtained by NSGA-II and DNA-NSGA-II for ZDT3.

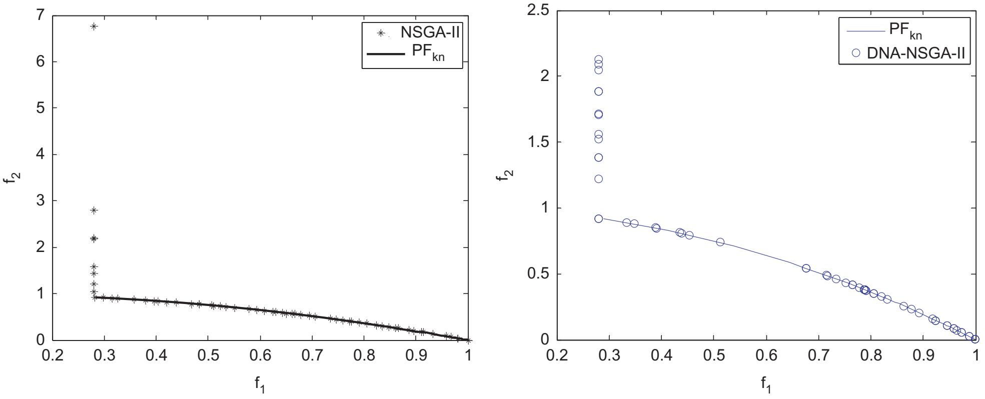

Fig. 13. Pareto-optimal front obtained by NSGA-II and DNA-NSGA-II for ZDT6.

Table 2
Comparison of indices between NSGA-II and DNA based NSGA-II.

	NSGA-II			DNA based NSGA-II		
	DE	ES	MS	DE	ES	MS
DEB	0.0661	1.4533	0.9981	0.0371	1.2279	0.9981
KUR	0.6495	0.7389	0.9833	0.0352	0.7149	0.9794
ZDT3	0.0520	0.6142	0.8967	0.0313	0.7984	0.9360
ZDT6	0.8351	3.6819	1	0.3819	1.9203	1

After the graphical comparison, we would continue the comparison in statistics of the indices through Table 2. First, in all test functions, the DS values of the proposed algorithm are smaller than that of NSGA-II. Although in some problems the ES and MS values of the proposed algorithm are not as good as NSGA-II, the solutions of NSGA-II are still inferior to that of DNA based NSGA-II since the former solution are dominated by the latter. For example, in the comparison Pareto-optimal front of ZDT3, the front of NSGA-II obviously does not converge to the known Pareto-optimal front while that of DNA based NSGA-II does as shown in Fig. 12. As mentioned in the last paragraph, the precision that the obtained solutions converging to the known Pareto-optimal front is more important. Hence, considering all the indices, we can say that the performance of DNA based NSGA-II is improved. From all the above comparison, we can find that the solutions obtained by DNA based NSGA-II outperform that of NSGA-II.

4. PH neutralization process

4.1. Description of pH neutralization process

In a pH neutralization process, reagent (acid or alkalescent solution) is neutralized by influent (alkalescent or acid solution) and pH of the solution in the reactor is achieved by measuring the effluent. Because the species contained in the influent and the reagent and the selection of the mixing equipment could be different, a practical pH neutralization process may be very complicated. In this paper, a simplified pH neutralization system is used whose schematic diagram is shown in Fig. 14.

In this process, pH value of effluent is controlled through manipulating the base flow rate. The parameters in the process are described as follows: q_1, q_2, q_3 and q_4 is the flow rate of acid stream, buffer stream, base stream and effluent stream separately, W_{a1},W_{a2},W_{a3} and W_{a4} is the charge balance coefficient of acid stream, buffer stream, base stream and effluent stream separately, W_{b1}, W_{b2}, W_{b3} and W_{b4} is the mass balance coefficient of acid stream, buffer stream, base stream and effluent stream separately, and pH_4 is the measurement of the effluent pH defined as $pH_4 = pH(t-\tau)$ due to time delay τ is introduced in the pH measurement. Refs. [27,28] give the dynamic model of this pH neutralization process as below.

$$\dot{h} = \frac{1}{A}(q_1 + q_2 + q_3 - C_v h^{0.5}) \tag{11}$$

$$\dot{W}_{a4} = \frac{1}{Ah}\left[(W_{a1} - W_{a4})q_1 + (W_{a2} - W_{a4})q_2 + (W_{a3} - W_{a4})q_3\right] \tag{12}$$

$$\dot{W}_{b4} = \frac{1}{Ah}\left[(W_{b1} - W_{b4})q_1 + (W_{b2} - W_{b4})q_2 + (W_{b3} - W_{b4})q_3\right] \tag{13}$$

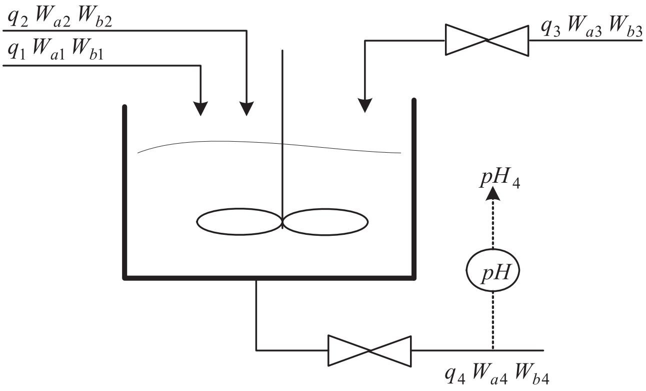

Fig. 14. A pH neutralization process.

Table 3
Operation parameters of the pH neutralization process.

$A=207$ cm^2	$W_{b2}=3\times10^{-2}$ mol/l
$C_v=8.75$ ml/cm/s	$W_{b3}=5\times10^{-5}$ mol/l
$pK_1=6.35$	$\tau=0.5$ min
$pK_2=10.25$	$q_1=16.6$ ml/s
$W_{a1}=3\times10^{-3}$ mol/l	$q_2=0.55$ ml/s
$W_{a2}=-3\times10^{-2}$ mol/l	$q_3=15.6$ ml/s
$W_{a3}=-3.05\times10^{-3}$ mol/l	$h=14.0$ cm
$W_{b1}=0$	$pH_4=7.0$

$$W_{a4}+10^{pH_4-14}+W_{b4}\frac{1+2\times10^{pH_4-pK_2}}{1+10^{pK_1-pH_4}+10^{pH_4-pK_2}}-10^{-pH_4}=0 \tag{14}$$

where h is the tank level, A is the cross-sectional area of tank, and $0\le q_3\le40$ ml/s. The sampling period is 0.25 s. Nominal model parameters and operating conditions are given in Table 3.

4.2. Result and analysis

As described in Section 2, DNA based NSGA-II is used to optimize the premise part parameters. Thus, one individual in the population represents one premise part of FRNN and then the corresponding consequent part can be obtained through RLS. That means one individual can represent one complete FRNN. In this section, we use DNA based NSGA-II and RLS to get the optimal parameters of FRNN. First, we choose q_3 as the input of model u, and the corresponding effluent pH as the output of model y. In the experiment, q_3 is generated by 500 random samples of between [0, 40] are generated as, and then 500 effluent pH corresponding to each q_3 are computed through Eqs. (11)–(14). Among the 500 groups of samples, 300 groups are used as the training samples, and another 200 groups are test samples. The parameters of the DNA based NSGA-II are set as follows: the maximum evolution generation is 200, population size is 30, encoding length for each parameter is 10, $p_{c1}=0.8$, $p_{c2}=0.5$, $p_{IA}=0.5$, $p_{MM}=0.5$ and $p_m=0.001$.

Through optimization, we select the final FRNN fuzzy rules as follows:

If $pH_4(k)$ is A_1

$$\begin{aligned} y(k) = {} & 0.6388-0.0274u(k)-0.0002u(k-1)-0.0001u(k-2) \\ & +0.0005y(k-1)+0.0004y(k-2)-0.0001y(k-3) \end{aligned}$$

If $pH_4(k)$ is A_2

$$\begin{aligned} y(k) = {} & 0.3766-0.0868u(k)-0.0007u(k-1)-0.0059u(k-2) \\ & +0.004y(k-1)+0.0013y(k-2)-0.0022y(k-3) \end{aligned}$$

If $pH_4(k)$ is A_3

$$y(k) = 7.9375+0.0045u(k)+0.0001u(k-2)+0.0002y(k-2)+0.0001y(k-3)$$

If $pH_4(k)$ is A_4

$$y(k) = 13.4995-0.0033u(k)+0.0001u(k-2)$$

If $pH_4(k)$ is A_5

$$\begin{aligned} y(k) = {} & 1.1518+0.0667u(k)+0.0007u(k-1)-0.0035u(k-2) \\ & -0.0003y(k-1)-0.002y(k-2)-0.0009y(k-3) \end{aligned}$$

If $pH_4(k)$ is A_6

$$\begin{aligned} y(k) = {} & 0.4446+0.017u(k)-0.0002u(k-1)+0.0099u(k-2) \\ & -0.0034y(k-1)+0.0014y(k-2)+0.0033y(k-3) \end{aligned}$$

where A_i, $i=1,2,\cdots,6$ is the fuzzy set.

The optimized membership function is shown in Fig. 15.

In order to verify the validity of the obtained model, it is used to predict the output of the 200 groups test data. The modeling error of the FRNN model is shown in Fig. 16.

Moreover, we give the comparison between the model established in this paper and the other T–S FRNN models optimized by GA and the k-means clustering method [29]. The modeling errors are shown in Figs. 17 and 18.

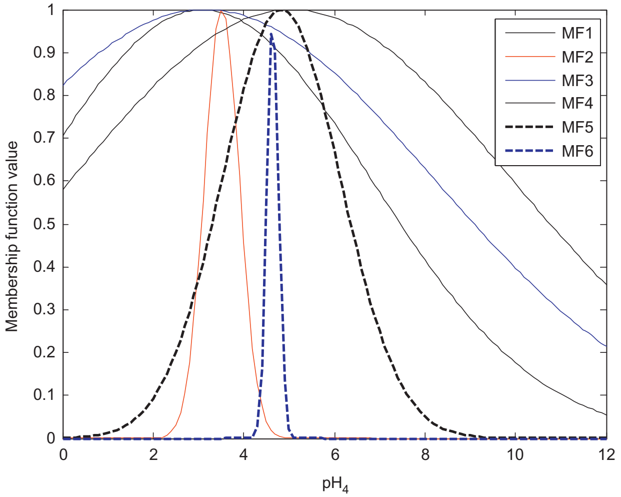

Fig. 15. Optimized membership function.

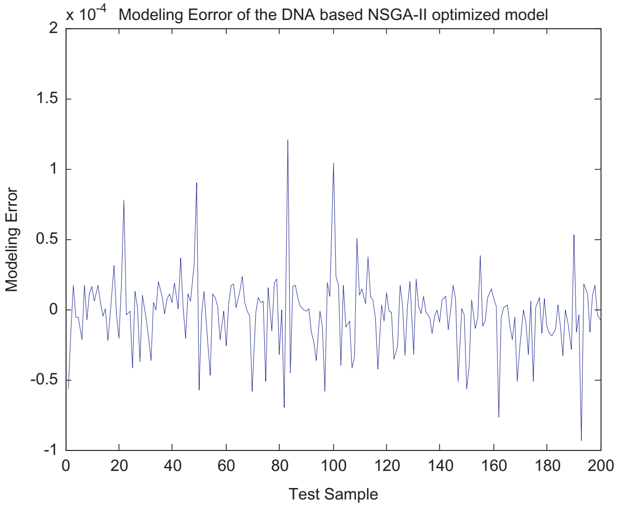

Fig. 16. Modeling error of the DNA based NSGA-II optimized T–S FRNN model.

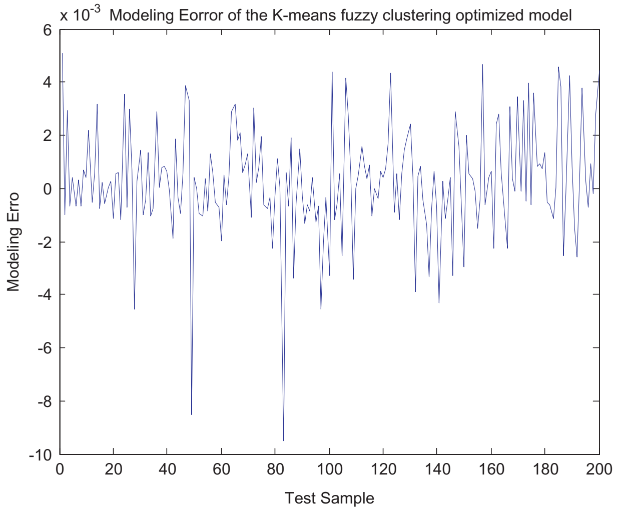

Fig. 17. Modeling error of the k-means clustering optimized T–S FRNN model.

From Figs. 16–18, we can find that the T–S FRNN model optimized by DNA based NSGA-II is more accurate than the other two T–S FRNN models that the maximum and average modeling error of our model is 1.2×10^{-4} and 1.8×10^{-5} compared with 0.0015 and 0.0095 of the k-means clustering optimized model or $5.7 \times 10^{-4}$and $8.1 \times 10^{-5}$of the GA optimized model.

5. Conclusion

In this study, the prediction model of a pH neutralization process is constructed by using T–S FRNN due to its ability of adequately approximating to the complicated systems. Since the

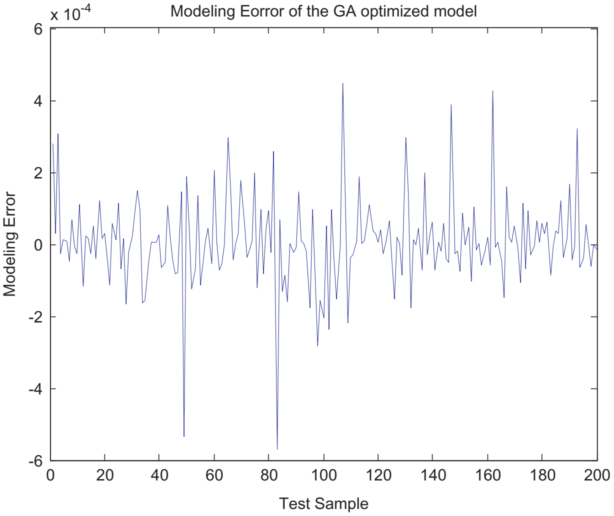

Fig. 18. Modeling error of the GA optimized T–S FRNN model.

accuracy and complexity of the network are two contradictory criteria, the modeling of the pH neutralization process can be represented as a multi-objective optimization problem whose objective functions are the minimum of the prediction error and the minimum of the fuzzy rules. To obtain the desired trade-off between those two objectives, DNA based NSGA-II is applied to optimize the unknown parameters of the T–S FRNN. For each individual, the DNA based NSGA-II uses the DNA coding method to represent the premise part of one possible T–S FRNN, and its corresponding consequent part are obtained by the RLS algorithm. Then the fitness and the crowding distance of the individuals are calculated. Meanwhile, modified DNA based crossover and mutation operators are employed to improve the global searching ability of the algorithm. The crowding distance tournament selection is also used in the algorithm to guarantee the elitisms are preserved in the population. To testify the effectiveness of the DNA based NSGA-II, four multi-objective functions are used to analyze the performance of the algorithm compared with NSGA-II. The comparison results show that the proposed algorithm outperforms NSGA-II in the convergence speed and the quality of the final Pareto-optimum. In view of the good performance of the DNA based NSGA-II, it is used to optimize the T–S FRNN model of the pH neutralization process. Compared with T–S FRNN models optimized by NSGA-II and the k-clustering method, the T–S FRNN model optimized by this paper is more accurate and the complexity of the model is also acceptable. Nevertheless, as a stochastic searching algorithm, the time that the proposed algorithm may be considerable. However, the genetic approach is an attractive solution for the design of efficient artificial neural networks.

Acknowledgment

This paper was supported by the National Natural Science Foundation of China under Grants no. 61104049, scientific research program of Zhejiang province education bureau no. Y201121483, and Foundation of Hangzhou Dianzi University under Grants no. KYF065610037.

References

[1] J. Ge, J. Qu, P. Lei, H. Liu, New bipolar electrocoagulation–electroflotation process for the treatment of laundry wastewater, Separation and Purification Technology 36 (2004) 33–39.

[2] M.A Henson, D.E. Seborg, Adaptive nonlinear control of a pH neutralization process, IEEE Transactions on Control Systems Technology 2 (1994) 169–182.

[3] J.M. Bölinga, D.E. Seborg, J.P. Hespanhac, Multi-model adaptive control of a simulated pH neutralization process, Control Engineering Practice 15 (2007) 663–672.

[4] B.M. Åkesson, H.T. Toivonen, J.B. Waller, R.H. Nyström, Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process, Computers and Chemical Engineering 29 (2005) 323–335.

[5] R. Zhang, A. Xue, S. Wang, Dynamic modeling and nonlinear predictive control based on partitioned model and nonlinear optimization, Industrial and Engineering Chemistry Research 50 (2011) 8110–8121.

[6] B.M. Al-Hadithi, A. Jiménez, F. Matía, A new approach to fuzzy estimation of Takagi–Sugeno model and its applications to optimal control for nonlinear systems, Applied Soft Computing 12 (2012) 280–290.

[7] J.A. Abdalla, R. Hawileh, Modeling and simulation of low-cycle fatigue life of steel reinforcing bars using artificial neural network, Journal of the Franklin Institute 348 (2011) 1393–1403.

[8] D. Hanbay, I. Turkoglu, Y. Demir, Modeling switched circuits based on wavelet decomposition and neural networks, Journal of the Franklin Institute 347 (2010) 607–617.

[9] H. Li, H. Liu, H. Gao, P. Shi, Reliable fuzzy control for active suspension systems with actuator delay and fault, IEEE Transactions on Fuzzy Systems 20 (2012) 342–357.

[10] A. Banakara, M.F. Azeem, Parameter identification of TSK neuro-fuzzy models, Fuzzy Sets and Systems 179 (2011) 62–82.

[11] A. Tewari, M. Macdonald, Knowledge-based parameter identification of TSK fuzzy models, Applied Soft Computing 10 (2010) 481–489.

[12] R. Babuška, H. Verbruggen, Neuro-fuzzy methods for nonlinear system identification, Annual Reviews in Control 27 (2003) 73–85.

[13] K. Salahshoor, M. Hamzehnejad, S. Zakeri, Online affine model identification of nonlinear processes using a new adaptive neuro-fuzzy approach, Applied Mathematical Modelling 36 (2012) 5534–5554.

[14] M. Sadeghian, A. Fatehi, Identification, prediction and detection of the process fault in a cement rotary kiln by locally linear neuro-fuzzy technique, Journal of Process Control 21 (2011) 302–308.

[15] R. Scherer, Neuro-fuzzy relational systems for nonlinear approximation and prediction, Nonlinear Analysis: Theory, Methods and Applications 71 (2009) 1420–1425.

[16] H. Li, B. Chen, C. Lin, Q. Zhou, Mean square exponential stability of stochastic fuzzy Hopfield neural networks with discrete and distributed time-varying delays, Neurocomputing 72 (2009) 2017–2023.

[17] H. Li, B. Chen, Q. Zhou, W. Qian, Robust stability for uncertain delayed fuzzy Hopfield neural networks with Markovian jumping parameters, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (2009) 94–102.

[18] I.S. Baruch, R.B. Lopez, J.O. Guzman, J.M. Flores, A fuzzy-neural multi-model for nonlinear system, Fuzzy Sets and Systems 159 (2008) 2650–2667.

[19] J. Nie, A.P. Loh, C.C. Hang, Modeling pH neutralization processes using fuzzy-neural approaches, Fuzzy Sets and Systems 78 (1996) 5–22.

[20] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II, In Parallel Problem Solving from Nature VI (2000) 849–858.

[21] G.A. Belmehdi, B. Dahhou, Multi-objective optimization of TSK fuzzy models, Expert Systems with Applications 36 (2009) 7416–7423.

[22] D. Mandal, S.K. Pal, P. Saha, Modeling of electrical discharge machining process using back propagation neural network and multi-objective optimization using non-dominating sorting genetic algorithm-II, Journal of Materials Processing Technology 186 (2007) 154–162.

[23] J. Tao, N. Wang, DNA computing based RNA genetic algorithm with applications in parameter estimation of chemical engineering processes, Computers and Chemical Engineering 31 (2007) 1602–1618.

[24] X. Chen, N. Wang, A DNA based genetic algorithm for parameter estimation in the hydrogenation reaction, Chemical Engineering Journal 150 (2009) 527–535.

[25] J.A. Rose, M. Hagiya, R.J. Deaton, A. Suyama, DNA-based in vitro genetic program, Journal of Biological Physics 28 (2002) 493–498.

[26] K. Wang, N. Wang, A protein inspired RNA genetic algorithm for parameter estimation in hydrocracking of heavy oil, Chemical Engineering Journal 167 (2011) 228–239.

[27] E.P. Nahas, M.A. Henson, D.E. Seborg, Nonlinear internal model control strategy for neural network models, Computers and Chemical Engineering 16 (1992) 1039–1057.

[28] R.C. Hall, D.E. Seborg, Modelling and self-tuning control of a multivariable pH process, Part I: Modelling and multiloop control, in: Proceedings of the 1989 American Control Conference, Pittsburgh, pp. 1822–1827.

[29] J. Tao, N. Wang, X. Chen, Multi-objective optimization based FRNN and its application to pH control process, CIESC Journal 60 (2009) 2820–2826.

Please cite this article as: X. chen, et al., Modeling of pH neutralization process using fuzzy recurrent neural network and DNA based NSGA-II, Journal of the Franklin Institute. (2013), http://dx.doi.org/10.1016/j.jfranklin.2013.03.014