

Optimal Learning of Neo-Fuzzy Structure Using Bacteria Foraging Optimization

Dong Hwa Kim

Dept. of Instrumentation and Control Eng., Hanbat National University,
16-1 San Duckmyong-Dong Yuseong-Gu, Daejeon City, Korea, 305-719.
E-mail: kimdh@hanbat.ac.kr, Homepage: aialab.net
Tel: +82-42-821-1170, Fax: +82-821-1164

Abstract: Fuzzy logic, neural network, fuzzy-neural network play an important as the key technology of linguistic modeling for intelligent control and decision in complex systems. The fuzzy-neural network (FNN) learning represents one of the most effective algorithms to build such linguistic models. This paper proposes bacteria foraging algorithm based optimal learning fuzzy-neural network (BA-FNN). The proposed learning scheme is the fuzzy-neural network structure which can handle linguistic knowledge as tuning membership function of fuzzy logic by bacteria foraging algorithm. The learning algorithm of the BA-FNN is composed of two phases. The first phase is to find the initial membership functions of the fuzzy neural network model. In the second phase, bacteria foraging algorithm is used for tuning of membership functions of the proposed model.

Keywords: Terms-Fuzzy neural network; Bacteria algorithm; Multiobjective control; Optimization.

1. Introduction

Some researchers suggest a model of fuzzy neuron that linear synaptic connections can be replaced with a nonlinearity characterized by a membership function and a fuzzy neural network model [1], [2]. The nonlinear characteristics of which are represented by fuzzy if-then rules with complementary membership functions. Since neo fuzzy neuron model or fuzzy neural network can have a good ability to describe a nonlinear relationship between multi-inputs and multi-output as well as its short leaning time compared with a conventional neural network, they are expecting as future linguistic tool for intelligence or soft computing. On the other hand, radial basis function networks (RBFNs) and back propagation neural networks (BPNNs) have yielded useful results in many practical areas such as pattern recognition, system identification and control, due primarily to their simple structures for realization and well established training algorithms. Many fuzzy paradigms, meanwhile, have been studied is recent years by viewing a fuzzy logic system (FLS) as a functionally equivalent RBFN or BPNN. As indicated in some papers [3], [4], the most important advantage of such an FLS spanned by fuzzy basic functions is the provision of a natural framework for combining numerical values and linguistic symbols in a uniform way. From a mathematical point of view, the input-output expressions of those mappings are identical in spite of the distinct inference procedure. Capability discrimination between neural and fuzzy system is thus diminished for proofs of universal neural/fuzzy approximators. Using neural networks or fuzzy systems to approximate a given plant or to control a process flow depends on whether rich available data are at hand or whether the 'If-Then' control heuristics could be established by human experts familiar with system dynamics under consideration. A

simple sigmoidal-like neuron is employed as a preassigned algorithm of the law of structural change which is directed by the current value of the error signal. However, in case of almost fuzzy logic, fuzzy-neural network, grade of membership and weighting function must be tuned by an approximation or experience-based tuning method. Some papers are written with a couple of objectives to demonstrate that genetic algorithms (GAs) are an efficient and robust tool for generating fuzzy rules and weighting function. GAs can construct a set of fuzzy rules that optimize multiple criteria [5]. An important observation that the rules searched by GAs are randomly scattered is made and a solution to this problem is provided by including a smoothness cost in the objective function.

On the other hand, as natural selection tends to eliminate animals with poor foraging strategies through methods for locating, handling, and ingesting food and favor the propagation of genes of those animals that have successful foraging strategies, they are more likely to apply reproductive success to have an optimal solution. Optimization models are also valid for social foraging where groups of animals communicate to cooperatively forage. Foraging can be modeled as an optimization process where an animal seeks to maximize the energy obtained per unit time spent foraging. Generally, a foraging strategy involves finding a patch of food (e.g., group of bushes with berries), deciding whether to enter it and search for food, and when to leave the patch. There are predators and risks, energy required for travel, and physiological constraints (sensing, memory, cognitive capabilities). Foraging scenarios can be modeled and optimal policies can be found using, for instance, dynamic programming. Search and optimal foraging decision-making of animals can be broken into three basic types: cruise (e.g., tunafish, hawks), saltatory (e.g., birds, fish, lizards, and insects), and ambush (e.g., snakes, lions). While to perform

social foraging an animal needs communication capabilities, it can gain advantages in that it can exploit essentially the sensing capabilities of the group, the group can gang-up on large prey, individuals can obtain protection from predators while in a group, and in a certain sense the group can forage with a type of collective intelligence. Social foragers include birds, bees, fish, ants, wild beasts, and primates. Generally, endowing each forager with more capabilities can help them succeed in foraging, both as an individual and as a group. From an engineering perspective, both ends of such a spectrum are interesting.

This paper proposes bacterial foraging based optimal learning approach of fuzzy-neural. The first phase of the BA-FNN is to find the initial membership functions of the fuzzy neural network model and the second phase is to obtain optimal membership functions of the proposed model by bacteria foraging algorithm.

2. Structure of a Bacteria Foraging Algorithm Based Fuzzy-Neural Network

The structure of BA-FNN is shown in Fig. 1 [3] and the output of the FNN part of BA-FNN can be represented by the following equation (1).

In Fig. 1 and Equation (1), the input space x_i is divided into several fuzzy segments which are characterized by membership functions $\mu_{i1}, \mu_{i2}, \dots, \mu_{in}$ within the range between x_{\min} and x_{\max} . The grade of membership function is also given as numbers assigned to labels of fuzzy membership function. The membership functions are followed by variable weights w_{i1}, \dots, w_{in} . Mapping from x_i to $f_i(x_i)$ is determined by fuzzy inferences and fuzzy rule is defined as Equation (2).

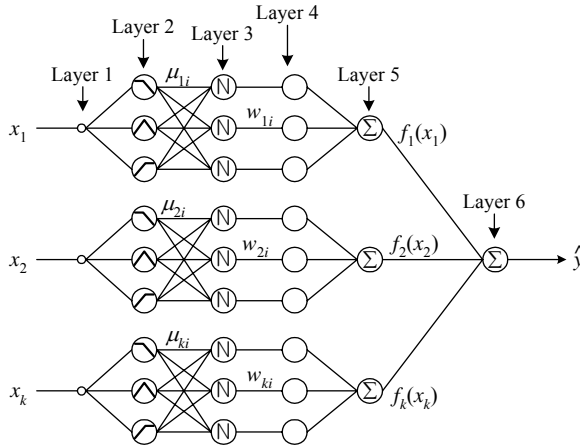


Fig. 1. The structure of bacteria algorithm based optimal learning fuzzy-neural network.

$$\begin{aligned} \bar{y} &= f_1(x_1) + f_2(x_2) + \dots + f_m(x_m) \\ &= \sum_{i=1}^m f_i(x_i) \end{aligned} \quad (1)$$

$$\begin{aligned} R^1 : & \text{ If } x_i \text{ is } A_{i1} \text{ then } C_{yxi} = w_{xi} \\ & \bullet \\ & \bullet \\ & \bullet \\ R^n : & \text{ If } x_i \text{ is } A_{in} \text{ then } C_{yxi} = w_{xi} \end{aligned} \quad (2)$$

As the fuzzy inferences adopted here is that of a singleton consequent, each weight w_{ij} is a deterministic value such as 0.8, 0.9. It should be emphasized that each membership function in antecedent is triangular and assigned to be complementary (so called by the authors) with neighbouring ones. In other words, an input signal x_i activates only two membership functions simultaneously and the sum of grades of these two neighbouring membership functions labelled by k and $k+1$ is always equal to 1, that is $\mu_{i,k}(x_i) + \mu_{i,k+1}(x_i) = 1$. So, the output of the fuzzy neural network can be represented by the following simple Equation (3).

$$\begin{aligned} f_i(x_i) &= \sum_{i=1}^n \mu_{xi} \cdot C_{yxi} \\ &= \frac{\sum_{j=1}^n \mu_{ij}(x_i) w_{ij}}{\sum_{j=1}^n \mu_{ij}(x_i)} = \frac{\mu_{ik}(x_i) w_{ik} + \mu_{i,k+1}(x_i) w_{i,k+1}}{\mu_{ik}(x_i) + \mu_{i,k+1}(x_i)} \\ &= \sum_{i=1}^n \mu_{xi} \cdot w_{xi} \end{aligned} \quad (3)$$

In this Equation, the weight w_{ij} are assigned by learning the rule of which is described by n if-then rules. That is, If input x_i lies in the fuzzy segment μ_{ij} , then the corresponding weight w_{ij} should be increased directly proportional to the output error $(y - \bar{y})$, because the error is caused by the weight. This proposition can be represented by the following equation;

$$f_i(x_i) = \mu_{xi}(x_i) w_{xi} + \mu_{xi+1}(x_i) w_{xi+1} \quad (4)$$

The learning procedure is the incremental change of weights for each input pattern. That is, the incremental change of minimizing the squared error (4) is obtained from

$$\Delta w_{xi}(t+1) = 2\delta(y - \bar{y}) \mu_{xi} + \alpha_i (w_{xi}(t) - w_{xi}(t-1)) \quad (5)$$

In this learning algorithm, all the initial weights are assigned to be zero and the updating of the weights is achieved after calculation of cumulative value in Equation (5).

Where, y is the given data, \bar{y} is the output of model,

δ learning rate, α is momentum constant and δ, α have the range of 0 to 1, respectively. The w_{xi} is the present weighting function and $w_{xi}(t-1)$ is the previous weighting function.

3. Behavior Characteristics and Modeling of Bacteria Foraging

3.1 Over view of Chemotactic Behavior of E. coli.

This paper considers the foraging behavior of E. coli, which is a common type of bacteria as in reference 12. Its behavior to move comes from a set of up to six rigid 100–200 rps spinning flagella, each driven as a biological motor. An E. coli bacterium alternates between running and tumbling. Running speed is 10–20 $\mu\text{m}/\text{sec}$, but they cannot swim straight. When we can summarize the chemotactic actions of bacteria as the following description:

- If in neutral medium, alternate tumbles and runs, its action is having search.

- If swimming up a nutrient gradient (or out of noxious substances), swim longer (climb up nutrient gradient or down noxious gradient)its behavior seeks increasingly favorable environments.

- If swimming down a nutrient gradient (or up noxious substance gradient), then search action is avoiding unfavorable environments.

So, it can climb up nutrient hills and at the same time avoid noxious substances. The sensors it needs for optimal resolution are receptor proteins which are very sensitive and high gain. That is, a small change in the concentration of nutrients can cause a significant change in behavior. This is probably the best-understood sensory and decision-making system in biology. Mutations in E. coli affect the reproductive efficiency at different temperatures, and occur at a rate of about 10^{-7} per gene and per generation. E. coli occasionally engages in a conjugation that affects the characteristics of a population of bacteria. Since there are many types of taxes that are used by bacteria such as, aerotaxis (it are attracted to oxygen), light (phototaxis), temperature (thermotaxis), magnetotaxis (it it can be affected by magnetic lines of flux. Bacteria can form intricate stable spatio-temporal patterns in certain semisolid nutrient substances. They can eat radially their way through a medium if placed together initially at its center. Moreover, under certain conditions, they will secrete cell-to-cell attractant signals so that they will group and protect each other. These bacteria can swarm.

3.2 Optimization Function of Bacterial Swarm Foraging

The main goal based on bacterial foraging is to apply in order to find the minimum of $P(\phi)$, $\phi \in R^n$, not in the gradient $\nabla P(\phi)$. Here, when ϕ is the position of a bacterium, and $J(\phi)$ is an attractant-repellant profile. That is, it means where nutrients and noxious substances are located, so $P < 0$, $P = 0$, $P > 0$ represent the presence of nutrients. A neutral medium, and the presence of noxious substances, respectively can showed by

$$H(j, k, l) = \{\phi^i(j, k, l) | i = 1, 2, \dots, N\}. \quad (6)$$

Equation represents the positions of each member in the population of the N bacteria at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event. Let $P(i, j, k, l)$ denote the cost at the location of the ith bacterium $\phi^i(j, k, l) \in R^n$. Reference [20, 21] let

$$\phi^i(j+1, k, l) = \phi^i(j, k, l) + C(i)\varphi(j), \quad (7)$$

so that $C(i) > 0$ is the size of the step taken in the random direction specified by the tumble. If at $\phi^i(j+1, k, l)$ the cost $J(i, j+1, k, l)$ is better (lower) than at $\phi^i(j, k, l)$, then another chemotactic step of size $C(i)$ in this same direction will be taken and repeated up to a maximum number of steps N_s . N_s is the length of the lifetime of the bacteria measured by the number of chemotactic steps. Functions $P_c^i(\phi)$, $i = 1, 2, \dots, S$, to model the cell-to-cell signaling via an attractant and a repellent is represented by [14-16]

$$P_c(\phi) = \sum_{i=1}^N P_{cc}^i = \sum_{i=1}^N \left[-L_{attract} \exp\left(-\delta_{attract} \sum_{j=1}^n (\phi_j - \phi_j^i)^2\right) \right] + \sum_{i=1}^N \left[-K_{repellant} \exp\left(-\delta_{attract} \sum_{j=1}^n (\phi_j - \phi_j^i)^2\right) \right], \quad (8)$$

When we where $\phi = [\phi_1, \dots, \phi_p]^T$ is a point on the optimization domain, $L_{attract}$ is the depth of the attractant released by the cell and $\delta_{attract}$ is a measure of the width of the attractant signal. $K_{repellant} = L_{attract}$ is the height of the repellent effect magnitude), and $\delta_{attract}$ is a measure of the width of the repellent. The expression of $P_c(\phi)$ means that its value does not depend on the nutrient concentration at position ϕ . That is, a bacterium with high nutrient concentration secrets stronger attractant than one with low nutrient concentration. Model use the function $P_{ar}(\phi)$ to represent the environment-dependent cell-to-cell signaling as

$$P_{ar}(\phi) = \exp(T - P(\phi))P_c(\phi) \quad (9)$$

where T is a tunable parameter. Model consider minimization of $P(i, j, k, l) + P_{ar}(\phi^i(j, k, l))$, so that the cells will try to find nutrients, avoid noxious substances, and at the same time try to move toward other cells, but not too close to them. The function $P_{ar}(\phi^i(j, k, l))$ implies that, with M being constant, the smaller $P(\phi)$, the larger $Par(\phi)$ and thus the stronger attraction, which is intuitively reasonable. In tuning the parameter M, it is normally found that, when M is very large, $Par(\phi)$ is much larger than $J(\phi)$, and thus the profile of the search space is dominated by the chemical attractant secreted by E. coli. On the other hand, if T is very small, then $Par(\phi)$ is much smaller than $P(\phi)$, and it is the effect of the nutrients that dominates. In $Par(\phi)$, the scaling

factor of $P_c(\phi)$ is given as in exponential form.

3.3 Bacteria Foraging Based Membership Function Tuning

In this paper, when the initial value of the membership function type of triangular as Fig. 2 is given by $X1_min=[0.46, 0.48]$, $X1_max=[0.77, 0.81]$, $X2_min=[45.0, 47.0]$, $X2_max=[61.0, 63.0]$, and learning rate boundary $\delta=[0.001, 0.01]$, momentum constant boundary $\alpha=[0.00001, 0.0004]$, respectively, the final membership function obtained by immune algorithm is dashed line as shown in Fig. 2.

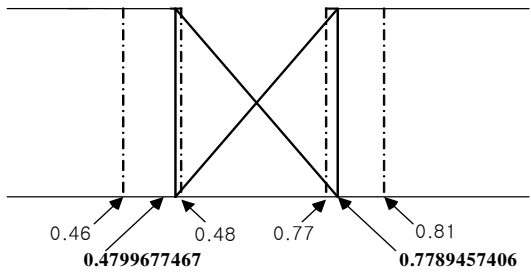


Fig. 2. Membership function shape of x_1 .

3.4 Bacteria Foraging Algorithm Based Computational Procedure for Optimal Selection of Parameter

In this algorithm, we use the immune algorithm based calculation procedure shown in Fig. 3 to optimize the learning rate, momentum term and fuzzy membership function of the above BA-FNN. We use 10 generation and 100 generation, 60 populations, 10 bits per string, crossover rate equal to 0.6, and mutation probability equal to 0.1, respectively.

This paper describes the method in the form of an algorithm to search optimal value of FNN parameter.

[step 1] Initialize parameters $n, N, NC, NS, Nre, Ned, Ped, C(i)$ ($i=1,2,\dots,N$), ϕ^i , and random values of PID parameter. Where,

n : Dimension of the search space (Each Parameter of FNN parameters),

N : The number of bacteria in the population,

NC : chemotactic steps,

Nre : The number of reproduction steps,

Ned : the number of elimination-dispersal events,

Ped : elimination-dispersal with probability,

$C(i)$: the size of the step taken in the random direction specified by the tumble. The controller parameter is searched in the range of parameter $Kp=[0\ 30]$, $Ti=[0\ 30]$, and $Td=[0\ 30]$.

[step 2] Elimination-dispersal loop: $l=l+1$

[step 3] Reproduction loop: $k=k+1$

[step 4] Chemotaxis loop: $j=j+1$

[substep a] For $i=1,2,\dots,N$, take a chemotactic step for bacterium i as follows.

[substep b] Compute $ITSE(i, j, k, l)$.

[substep c] Let $ITSElast=ITSE(i, j, k, l)$ to save this value since we may find a better cost via a run.

[substep d] Tumble: generate a random vector $\Delta(i) \in R^n$ with each element $\Delta_m(i), m=1,2,\dots,P$, a random number on $[-1, 1]$.

[substep e] Move: Let

$$\phi^i(j+1, k, l) = \phi^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i .

[substep f] Compute $ITSE(i, j+1, k, l)$.

[substep g] Swim

i) Let $m=0$ (counter for swim length).

ii) While $m < N_s$ (if have not climbed down too long).

• Let $m=m+1$.

• If $ITSE(i, j+1, k, l) < ITSElast$ (if doing better), let $ITSElast=ITSE(i, j+1, k, l)$ and let

$$\phi^i(j+1, k, l) = \phi^i(j+1, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

and use this $\phi^i(j+1, k, l)$ to compute the new $ITSE(i, j+1, k, l)$ as we did in [substep f]

• Else, let $m=N_s$. This is the end of the while statement.

[substep h] Go to next bacterium ($i, 1$) if $i \neq N$ (i.e., go to [substep b] to process the next bacterium).

[step 5] If $j < NC$, go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.

[step 6] Reproduction:

[substep a] For the given k and l , and for each $i=1,2,\dots,N$, let

$$ITSE_{health}^i = \sum_{j=1}^{N_c+1} ITSE(i, j, k, l)$$

be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost $ITSE_{health}$ (higher cost means lower health).

[substep b] The S_r bacteria with the highest $ITSE_{health}$ values die and the other S_r bacteria with the best values split (and the copies that are made are placed at the same location as their parent).

[step 7] If $k < N_{re}$, go to [step 3]. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

[step 8] Elimination-dispersal: For $i=1,2,\dots,N$, with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant).

To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to [step 2]; otherwise end.

4. Simulation and Discussions

In order to prove the learning effect of the proposed Bacteria based FNN (BA-FNN), we use the second-order highly nonlinear difference equation given as [4]

$$y_k = \frac{y_{k-1}y_{k-2}(y_{k-1} + 2.5)}{1 + y_{k-1}^2 + y_{k-2}^2} + u_k \quad (10)$$

In the gas furnace, $u(t-3)$ and $y(t-1)$ as input, $y(t)$ as output is used. Fig. 3 shows performance index by foraging differentiation rate of bacteria algorithm for the given model (8) and Figs. 4 and 5 are best fitness and object function depending on and differentiation rate of foraging selection $pCS=0.2$ and foraging selection $pCS=0.5$, respectively, when the number of membership function is 3. Fig. 4 and 5 shows comparison of fitness value depending differentiation rate of foraging (pCS), when the number of member ship function is 2 ($mem=[2, 2]$). Fig. 7 and 9 represents best value of fitness function and object function, when the number of membership function is 3 ($mem=[3, 3]$) and differentiation rate of foraging $pCS=0.2$, respectively. Fig. 10 is comparison of fitness value by pCS to the number of member ship function, Fig. 11 and 12 are showing the best value of fitness when learning parameter of bacteria algorithm is 100 generation, 0.2 pCS (differentiation rate of foraging selection), and the number of membership is 3.

Table 1. Parameter obtained by simulation.

pCS	2:2		3:3	
	PI	E_PI	PI	E_PI
0.2	0.0354	0.2857	0.0354	0.2857
0.3	0.0408	0.2729	0.0356	0.2855
0.4	0.0409	0.2726	0.0359	0.2852
0.5	0.0394	0.2742	0.0361	0.2847

Figs. 16 and 17 is performance index error (PI) and test index error (E_PI), when the number of member ship function is 3. Table 1 is the value of PI and E_PI by pCS and Table 2 is membership function shape depending on generation of immune algorithm. Table 3 depicts comparison of the learning results obtained by GA based FNN model, HCM and GA based FNN, and the immune based FNN model proposed in this paper. Table 4 is the results depending on 10 generation and 100 generation in immune algorithm, respectively.

5. Conclusions

Since Fuzzy sets and fuzzy logic can capture the approximate, qualitative aspects of human reasoning and decision-making processes, they have been considered as effective tools to deal with uncertainties in terms of vagueness, ignorance, and imprecision

On the other hand, neural networks (NN) appeared as promising tools (or designing high performance control systems), because they have the potential for dealing with favorable scenarios owing to nonlinear dynamics, drift in plant parameters, and shifts in operating points. Since then, the fuzzy-neural network (FNN) learning represents one of the most effective algorithms to build such linguistic models for

control system or making decision.

However, in many case, tuning of membership and weighting function remain difficulties. Some papers studied that genetic algorithms (GAs) are an efficient and robust tool for generating fuzzy rules and weighting function.

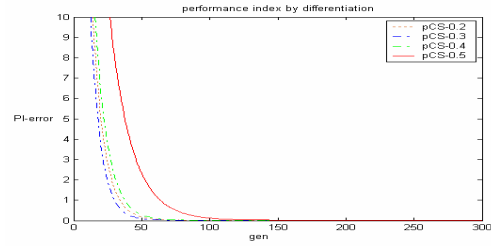


Fig. 3. Performance index by differentiation rate of immune algorithm.

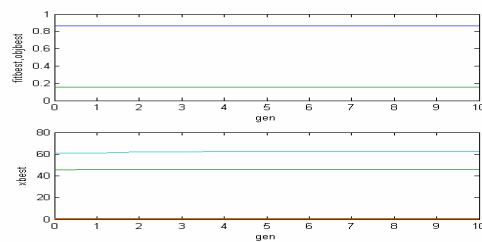


Fig. 4. Best value of fitness function and object function. ($mem=[2, 2]$, $pCS=0.2$)

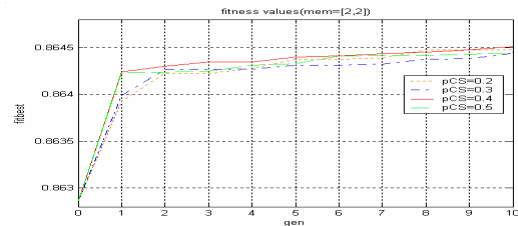


Fig. 5. Comparison of fitness value depending pCS . ($mem=[2, 2]$)

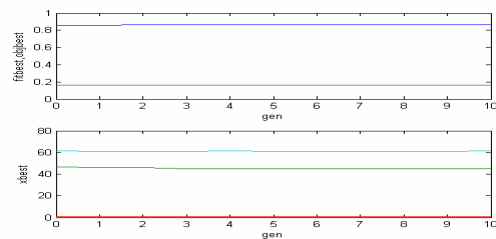


Fig. 6. Best value of fitness function and object function. ($mem=[3, 3]$, $pCS=0.5$)

This paper proposes optimal learning method of fuzzy-neural network by bacteria foraging algorithm. The proposed learning model is the fuzzy-neural network form which can to obtain optimized linguistic knowledge by bacteria foraging algorithm. The learning algorithm of a bacteria foraging based FNN (BA-FNN) is composed of finding the initialization of membership functions and tuning of membership functions. The results are compared with the results by GA (genetic algorithm based neural network) and fuzzy-neural network, respectively. The results by the proposed learning method are showing more satisfactory than the other learning schemes.

Table 2. Membership shape depending on generation of immune algorithm, the number of membership function, pcs.

Item	X_1_{min}	X_1_{max}	X_2_{min}	X_2_{max}	δ	α	PI	E_PI
A _i	[0.46, 0.48]	[0.77, 0.81]	[45.0, 47.0]	[61.0, 63.0]	[0.001, 0.001]	[0.00001, 0.0004]		
A ₀	0.4799677467	0.7789457406	46.247500655	62.2563049853	0.00100149345	0.0000354499	0.040311	0.27306
A ₁	0.4799938583	46.2618954295	0.7814936556	62.4140905514	0.00100555325	0.0000465945	0.040598	0.27277
A ₂	0.479995994	46.2495033736	0.7825169491	62.5021071454	0.00100012016	0.0000197420	0.040491	0.27288
A ₃	0.479996948	46.2499020098	0.7813410294	62.4214891638	0.00100003433	0.0000420978	0.040452	0.27292
A ₄	0.4600006103	45.0051097918	0.7700020980	61.0629330281	0.00268751019	0.0000279584	0.03265	0.28551
A ₅	0.4600009918	45.0002574923	0.7700003814	61.2041646997	0.00262390100	0.0002727694	0.035923	0.28475
A ₆	0.4600003242	45.0000438690	0.7700003814	61.2088834847	0.00261991846	0.00039791196	0.035983	0.28469

A₀ gen=[100], mem=[2, 2], pcs=0.2, A₁ gen=[100], mem=[2, 2], pcs=0.3, A₂ gen=[100], mem=[2, 2], pcs=0.4
A₃ gen=[100], mem=[2, 2], pcs=0.5, A₄ gen=[100], mem=[3, 3], pcs=0.2, A₅ gen=[100], mem=[3, 3], pcs=0.4
A₆ gen=[100], mem=[3,3], pcs=0.5

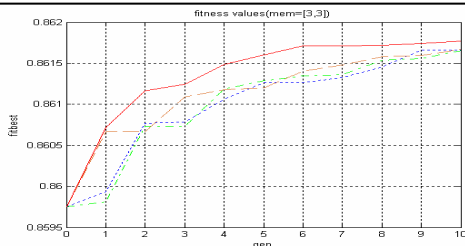


Fig. 7. Comparison of fitness value depending on pcs. (mem=[3, 3])

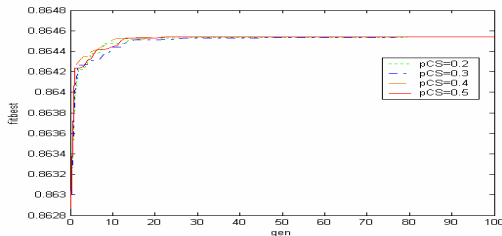


Fig.8. Best value of fitness by pcs. (mem=[2, 2])

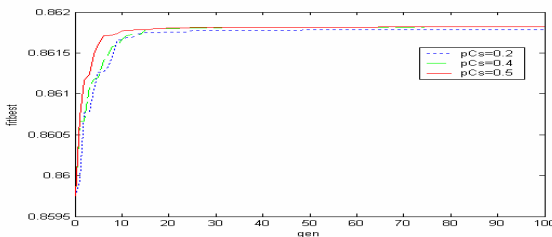


Fig. 10. Best value of fitness by pcs. (mem=[3, 3])

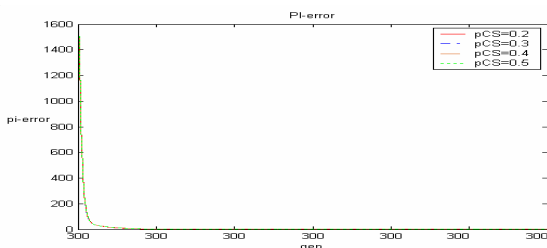


Fig. 11 PI- error by pcs. (mem=[2, 2])

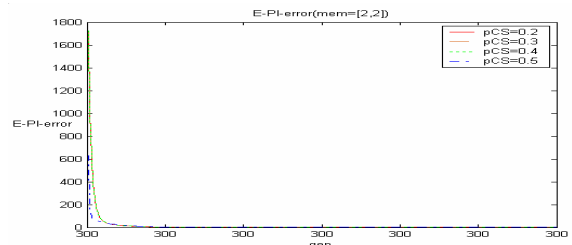


Fig. 12. E_PI- error by pcs. (mem=[2, 2])

ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge the financial support of KESRI (Korea Electrical Engineering & Science Research Institute) under project. R-2003-B-286.

REFERENCES

- [1] Park. S. and Sandberg. I.W, "Approximation and radial-basis-function networks," Neural Computer, pp. 105-110.
- [2] C.C. Lee, "Fuzzy logic in control system: Fuzzy logic controller, part I and II," IEEE Trans. Syst. Man Cybern, Vol.20, No.2, pp 404-435, 1990.
- [3] WANG, H., BROWN, M., and HARRIS, C.J., "Neural network modeling of unknown nonlinear systems subject to immeasurable disturbances," IEE Proc., Control Theory Appl., Vol. 141, No. 4, pp. 216-222, 1994.
- [4] HORIKAWA. S. FURUHASHI. T. and UCHIKAWA. Y, "On fuzzy modeling using fuzzy neural networks with back propagation algorithm," IEEE Trans. Neural network, pp. 801-806, 1992.
- [5] Wael A. Farag. Victor H. Quintana, and Germano Lambert-Torred, "A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems," IEEE Trans. on neural networks, Vol. 9, No. 5, Sept. 1998.
- [6] J. D. Farmer, N. H. Packard and A. S. Perelson, "The

immune system, adaptation, and machine learning,”
Physica, Vol. D, No. 22, pp. 187 - 204, 1986.

- [7] Kazuyuki Mori and Makoto Tsukiyama, “Immune algorithm with searching diversity and its application to resource allocation problem,” Trans. JIEE, Vol. 113 - C, No. 10, 1993.