An Immune-Fuzzy Neural Network For Dynamic System

Dong Hwa Kim, Jae Hoon Cho
Dept. of Instrumentation and Control Eng., Hanbat National University,
16-1 San Duckmyong-Dong Yuseong-Gu, Daejon City, Korea, 305-719.
E-mail: kimdh@hanbat.ac.kr, Hompage: ial.hanbat.ac.kr
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 making in complex systems. The fuzzy-neural network (FNN) learning represents one of the most effective algorithms to build such linguistic models. This paper proposes learning approach of fuzzy-neural network by immune algorithm. The proposed learning model is presented in an immune based fuzzy-neural network (FNN) form which can handle linguistic knowledge by immune algorithm. The learning algorithm of an immune based FNN is composed of two phases. The first phase used to find the initial membership functions of the fuzzy neural network model. In the second phase, a new immune algorithm based optimization is proposed for tuning of membership functions and structure of the proposed model.**

Index Terms-Fuzzy neural network; Immune algorithm; Multiobjective control; Optimzation.

I. Introduction

Some researchers describe 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. 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 realisation 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. This paper proposes immune based learning approach of fuzzy-neural (FNN) [8-10].

II. Structure of a Fuzzy-Neural Network

The structure of the fuzzy-neural network (FNN) is shown in Fig.1 [3] and the output of this FNN can be represented by the following equation:

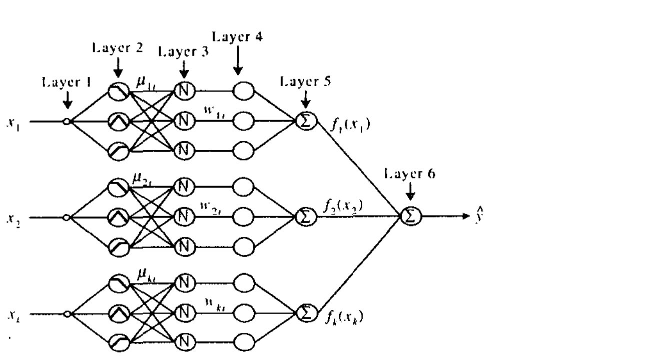


Fig. 1. The structure of fuzzy-neural network.

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

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

$$\begin{aligned} &R^1 : \textit{If } x_i \textit{ is } A_{i1} \textit{ then } C_{yx1} = w_{x1} \\ &\quad\bullet \\ &\quad\bullet \\ &\quad\bullet \\ &R^n : \textit{If } x_i \textit{ is } A_{in} \textit{ then } C_{yxn} = w_{xn} \end{aligned} \tag{2}$$

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.

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

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;

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

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

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

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.

III. Immune Algorithms for FNN

A. Immune Algorithm

When an antibody on the surface of a B cell binds an antigen, that B cell becomes stimulated. The level of stimulation depends not only on how well the B cell's antibody matches the antigen, but also how it matches other B cells in the immune network [4]. The stimulation level of the B cell also depends on its affinity with other B cells in the immune network. This network is formed by B cells possessing an affinity to other B cells in the system. If the stimulation level rises above a given threshold, the B cell becomes enlarged and if the stimulation level falls below a given threshold, the B cell die off. The more neighbors a B cell has an affinity with, the more stimulation it will receive from the network, and vice versa. Against the antigen, the level to which a B cell is stimulated relates partly to how well its antibody binds the antigen. We take into account both the strength of the match between the antibody and the antigen and the B cell object's affinity to the other B cells as well as its enmity. Therefore, generally the concentration of i-*th* antibody, which is denoted by δ_i, is calculated as follows [6-8]:

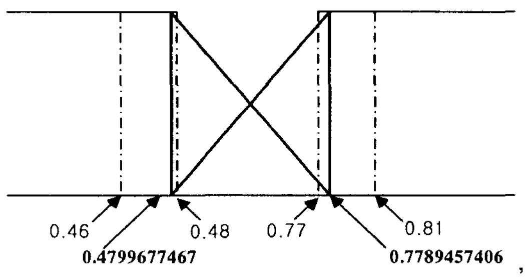
$$\frac{dS_i(t)}{dt} = \left(\begin{array}{l} \sigma \sum_{j=1}^{N} m_{ji} \xi_j(t) \\ -\sigma \sum_{k=1}^{N} m_{ik} \xi_k(t) + \beta m_i - \gamma_i \end{array} \right) \delta_i(t), \quad (6a)$$

$$\frac{d\sigma_i(t)}{dt} = \frac{1}{1+\exp\left(0.5 - \frac{dS_i(t)}{dt}\right)}, \quad (6b)$$

In Eq. (6), N is the number of antibodies, and σ and β are positive constants. m_{ji} denotes affinities between antibody j and antibody i (i.e. the degree of interaction), m_i represents affinities between the detected antigens and antibody i, respectively.

B. Immune Based Membership Function Tuning

In this paper, when the initial value of the membership function type of triangular as Fig. 2 is given by X_1_min=[0.46, 0.48], X_1_max=[0.77, 0.81], X_2_min=[45.0, 47.0] X_2_max=[61.0, 63.0], and learning rate boundary δ =[0.001, 0.01], momentum constant boundary α=[0.00001, 0.0004], respectively, the final membership function is obtained by immune algorithm of Fig. 3.



(13)

Fig. 2. Membership function shape.

C. 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 FNN. We use 100 generations, 60 populations, 10 bits per string, crossover rate equal to 0.6, and mutation probability equal to 0.1.

[Step 1] Initialization and recognition of antigen: The immune system recognizes the invasion of an antigen.

[Step 2] Product of antibody from memory cell: The immune system produces the antibodies that were effective to kill the antigen in the past. This is implemented by recalling a past successful solution from memory cell. For each individual of the network population, calculate the fitness function using memory cell to membership function, learning rate and momentum constant.

[Step 3] Antibody with the best fitness value obtained by calculation for searching an optimal solution is stored in memory cell.

[Step 4] Differentiation of lymphocyte: The B-lymphocyte cell, the antibody that matched the antigen, is dispersed to the memory cells in order to respond to the next invasion quickly. That is, select individuals using tournament selection and apply genetic operators (crossover and mutation) to the individuals of network.

[Step 5] Stimulation and suppression of antibody: The expected value η_k of the stimulation of the antibody is given by

$$\eta_k = \frac{m_{\varphi k}}{\sigma_k} \quad (7)$$

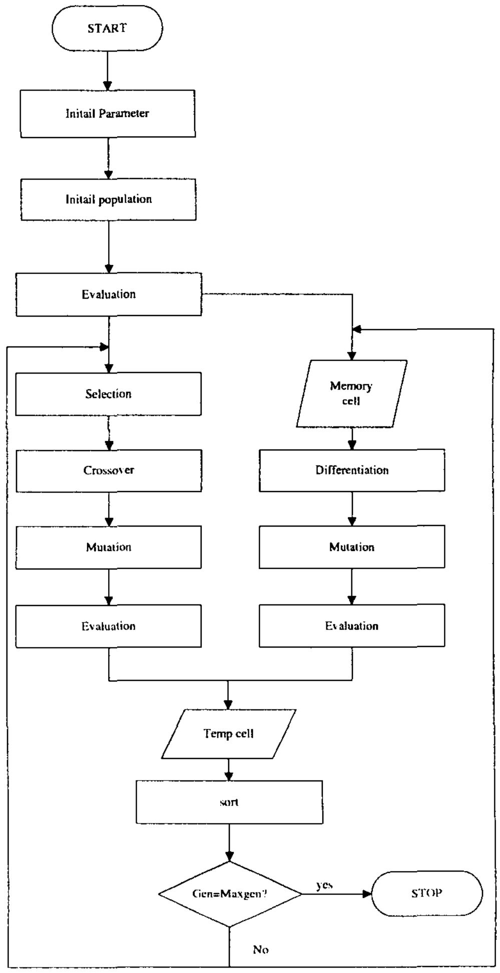


Fig. 3. Computation procedure by genetic and clonal selection of immune algorithm.

where σ_k is the concentration of the antibodies.

An immune system can control the concentration and the variety of antibodies in the lymphocyte population. If antibody obtains a higher affinity against an antigen, the antibody stimulates. However, an excessive higher concentration of an antibody is suppressed. Through this function, an immune system can maintain the diversity of searching directions and a local minimum.

[Step 6] Calculate fitness value between antibody and antigen. This procedure can generate a diversity of antibodies by a genetic reproduction operator such as *mutation or crossover. These genetic operators are expected* to be more efficient than the generation of antibodies.

[Step 7] If the maximum number of generations of memory cell is reached, stop and return the fitness of the best individual fitness value to network; otherwise, go to step 3.

IV. Simulations and Discussions

In order to prove the learning effect of the proposed immune based FNN, we use the second-order highly nonlinear difference equation given as [8]

$$y_k = \frac{y_{k-1}y_{k-2}(y_{k-1}+2.5)}{1+y_{k-1}^2+y_{k-2}^2} + u_k . \quad (9)$$

In the gas furnace, u(t-3) and y(t-1) as input, y(t) as output is used. Fig. 4 shows the variation of parameter to objection variation obtained using clonal selection of immune algorithm and Figs. 5 and 6 are best fitness and object function depending on the number of membership function 2 and differentiation rate of clonal selection (pCS). Fig. 7 and Fig. 8 represent fitness value by pCS when the number of membership function is 2, 3, respectively.

Fig. 9 is showing the best value of fitness when learning parameter of immune algorithm is 100 generation, 0.2 pCS (differentiation rate of clonal selection) and the number of membership is 3. Figs. 10 and 11 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 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 result depending on generation in immune algorithm.

V. Conclusions

For complex and/or ill-defined systems that are not easily subjected to conventional automatic control methods, FLCs provide a feasible alternative since they can capture the approximate, qualitative aspects of human reasoning and decision-making processes. On the other hand, research of neural control has evolved quickly and a number of neural controller design methods have been proposed in the literature. Since then, the fuzzy-neural network (FNN) learning represents one of the most effective algorithms to build such linguistic models for control system. 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.

This paper proposes learning approach of fuzzy-neural network by immune algorithm. The proposed learning model is presented in an immune based fuzzy-neural network (FNN) form which can handle linguistic knowledge by immune algorithm. The learning algorithm of an immune based FNN is composed of two phases. The first phase used to find the initial membership functions of the fuzzy neural network model. In the second phase, a new immune algorithm based optimization is proposed for tuning of membership functions and structure of the proposed model.

Table 1. Parameter obtained by simulation.

pCS	2:2		3:2		3:3	
	PI	E_PI	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

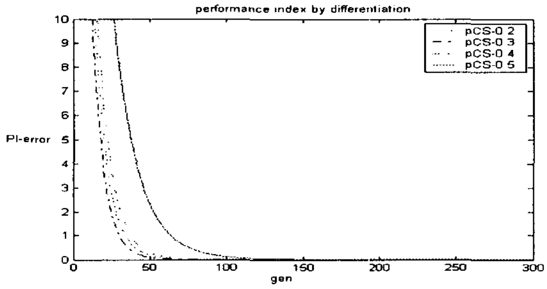


Fig. 4. Performance index by differentiation rate of immune algorithm.

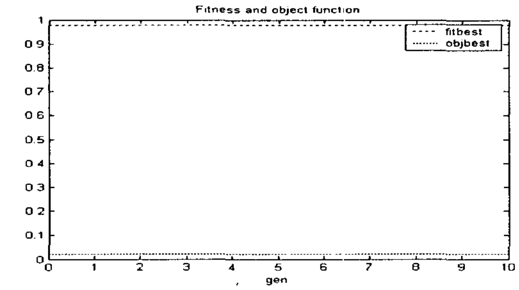


Fig. 5 Best fitness function and object function

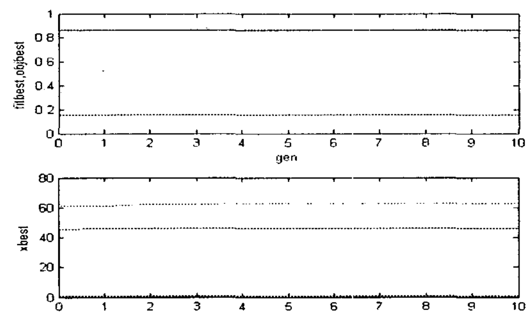


Fig. 6. Best value of fitness function and object function.

(mem=[2, 2], pCS=0.2)

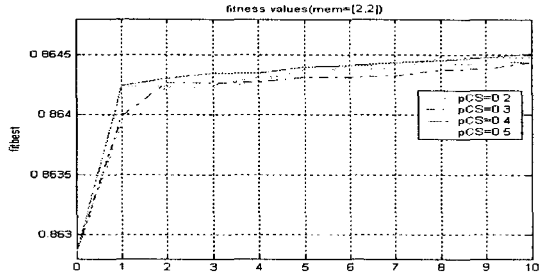


Fig. 7. Comparison of fitness value depending pCS. (mem=[2, 2])

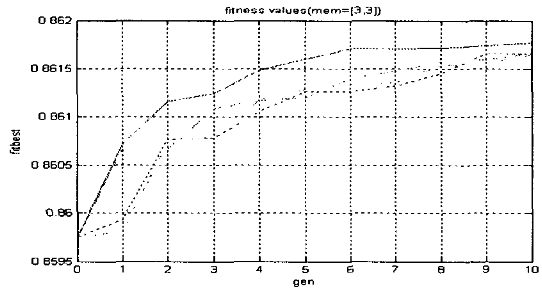


Fig. 8. Comparison of fitness value by pCS (mem=[3, 3])

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
Ai	[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
A2	0.4799995994	46.2495033736	0.7825169491	62.5021071454	0.00100012016	0.0000197420	0.040491	0.27288
A_3	0.4799996948	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

A_0 gen=[100], mem=[2, 2], pcs=0.2
A_1 gen=[100], mem=[2, 2], pcs=0.3
A_2 gen=[100], mem=[2, 2], pcs=0.4
A_3 gen=[100], mem=[2, 2], pcs=0.5
A_4 gen=[100], mem=[3, 3], pcs=0.2
A_5 gen=[100], mem=[3, 3], pcs=0.4

Table 3. Comparison of learning methods.

	PI	E_PI	MF
FNN model (GA)	0.027	0.298	4
	0.026	0.304	6
FNN model (HCM+GA)	0.027	0.294	4
	0.032	0.276	6
FNN model (CS-GA)	0.0394	0.274	4
	0.0361	0.284	6

REFERENCES

[1] PARK. S. and SANDBERG. 1.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 subiect 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, '93.

[8] Dong Hwa Kim, "Auto-tuning of reference model based PID controller using immune algorithm," IEEE international conference on evolutionary computation, Hawaii, May 12 - 17, 2002.

[9] Dong Hwa Kim, "Intelligent tuning of a PID controller using an immune algorithm," Trans. KIEE, Vol. 51-D, No.1, pp. 2002.

[10] Dong Hwa Kim, "PID Controller Tuning of a Boiler Control System Using Immune Algorithm Typed Neural Network," ICCS2004, Poland, June 2004.

[11] byoung Jun Park, "Fuzzy polynomial neural networks: Hybrid architectures of fuzzy modeling," IEEE Trans. on Fuzzy systems, Vol. 10, No. 5, pp. 607-621, Oct. 2002.

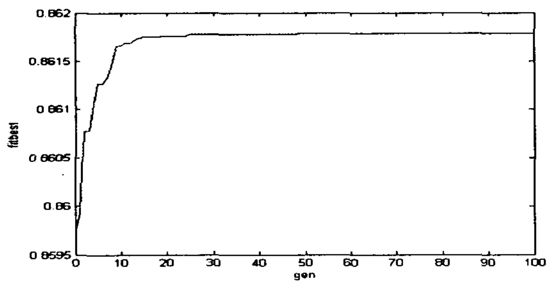


Fig. 9. Best value of fitness(gen=100,pCS=0.2,mem=[3,3])

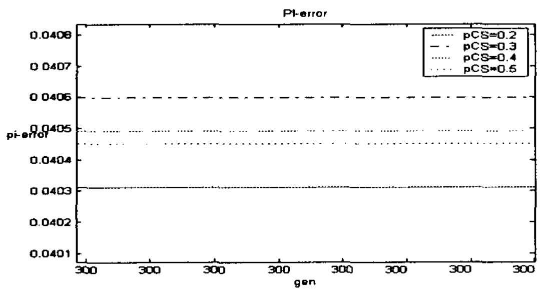


Fig. 10. PI- error by pCS(mem=[3, 3])

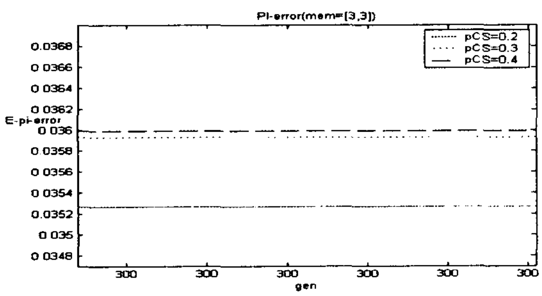


Fig.11. E_PI- error by pCS(mem=[3, 3])

Table 4. Comparison of learning methods in 10 generation and 100 generation.

	Gen=10				Gen=100			
pCS	2:2		3:3		2:2		3:3	
	PI	E_PI	PI	E_PI	PI	E_PI	PI	E_PI
0.2	0.0354	0.2857	0.0354	0.2857	0.040311	0.27306	0.035265	0.28551
0.3	0.0408	0.2729	0.0356	0.2855	0.040598	0.27277		
0.4	0.0409	0.2726	0.0359	0.2852	0.040491	0.27288	0.035923	0.28475
0.5	0.0394	0.2742	0.0361	0.2847	0.040452	0.27292	0.035983	0.28469