International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

Neural Network Learning based on Chaos

Truong Quang Dang Khoa and Masahiro Nakagawa

***Abstract*—**Chaos and fractals are novel fields of physics and mathematics showing up a new way of universe viewpoint and creating many ideas to solve several present problems. In this paper, a novel algorithm based on the chaotic sequence generator with the highest ability to adapt and reach the global optima is proposed. The adaptive ability of proposal algorithm is flexible in 2 steps. The first one is a breadth-first search and the second one is a depth-first search. The proposal algorithm is examined by 2 functions, the Camel function and the Schaffer function. Furthermore, the proposal algorithm is applied to optimize training Multilayer Neural Networks.

Keywords—Learning and Evolutionary Computing, Chaos optimization algorithm, Artificial Neural Networks, Nonlinear optimization, Intelligent Computational Technologies.

I. Introduction

CHAOS optimization algorithms as a novel method of global optimization have attracted much attention. At present, there are two groups working on the research of chaotic optimization. The first group aimed at global optimization by chaotic dynamic neural network proposed by Aihara et al. in [2,3] which mainly used the transiently chaotic neural network with chaotic annealing developed to find globally optimal solutions for combinatorial problems, such as traveling salesman problem and maintenance scheduling problem. The second group searched the global optimum by the chaos optimization algorithm, which utilized the nature of chaos sequences such as pseudo-randomness, ergodicity and irregularity in [4-13].

The main ideas of chaos optimization algorithm in the second group generally include two major steps. Firstly, define a chaotic sequences generator based on the Logistic map. Generate a sequence of chaotic points and map it to a sequence of design points in the original design space. Then, calculate the objective functions with respect to the generated design points, and choose the point with the minimum objective function as the current optimum. Secondly, to enhance the effective computation the hybrid methods attracted the attention of some researchers, in which the chaos optimization algorithm was used for global search and the conventional optimization algorithms were employed for local search near the global optimum. The hybrid methods can save much computing time and enhance the computational efficiency of algorithms. In [7] a combination of chaos optimization algorithm (COA) and weighted gradient direction was proposed. In [8,9] the chaos optimization algorithm was combined with genetic algorithm. In [11] the interior point algorithm was a hybrid of COA to solve various global optimizations for power flow of electrical engineering. In [12] authors suggested a hybrid method of simulated annealing and in [13] particle swarm optimization with chaos search was proposed.

On the other hand, Neural Networks is one of the most interesting areas of A.I. of which the well-known characteristic is the learning ability. However the training of networks is being argued to improve it. The back-propagation algorithm is very popular gradient descent method for training feed forward neural networks. In its simplest form, it performs fixed step size, steepest descent on an error surface in parameter space. In [16], these error surfaces are highly nonlinear and quite complicated, forcing the use of very small step size to ensure stable convergence of the search procedure. This makes the algorithm really slow and trapped in suboptimal local minima.

Both global optimization and faster convergence are two keys issues of training algorithms. This work proposes a simple algorithm based on the chaotic sequence generator with the highest ability to adapt and reach the global optima. The adaptive ability of proposal algorithm is flexible in 2 procedures. The first one is called “Breadth-first search” and the second one is “Depth-first search”. The proposal algorithm is examined by 2 functions, the Camel function with six local minima and two global minima and the Schaffer function with infinite local minima and one global minimum. After that, the proposal algorithm is applied to optimizing training multilayer neural networks. A typical problem used to illustrate is XOR problem. By the way, some illustrations are shown to compare with a traditional global search, Genetic Algorithms. The remainder of this paper is organized as follows. In section 2, the mathematic model of chaos algorithm is introduced. In section 3, the backpropagation algorithm for training is described. All the illustration results are showed in section 4.

Manuscript received April 2nd, 2007.

This work was supported in part by The 21st Century COE (Center of Excellence) Program “Global Renaissance by Green Energy Revolution” from Ministry of Education, Culture, Sports, Science and Technology of Japan.

Authors are with the Chaos and Fractals Informatics Laboratory, Nagaoka University of Technology, 1603-1 Kamitomiokamachi, Nagaoka, Niigata 940-2188, JAPAN (e-mail: khoa@ieee.org).

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

II. Mathematical Model

A. Chaos Variables

Chaos variables are usually generated by the well-known logistic map. The logistic map is a one-dimensional quadratic map defined by:

$$\gamma(k+1) = \mu\,\gamma(k)(1-\gamma(k)) \qquad (1)$$

Where $0 \le \mu \le 1$ is a control parameter. Despite the apparent simplicity of the equation, the solution exhibits a rich variety of behavior. For μ = 4, generates chaotic evolutions. Its output is like a stochastic output, no value of γ is repeated and the deterministic equation is sensitive to initial conditions as shown in Fig. 1.

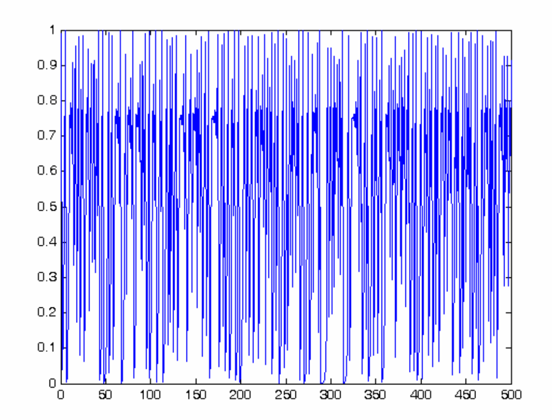

Fig. 1 The Logistic mapping.

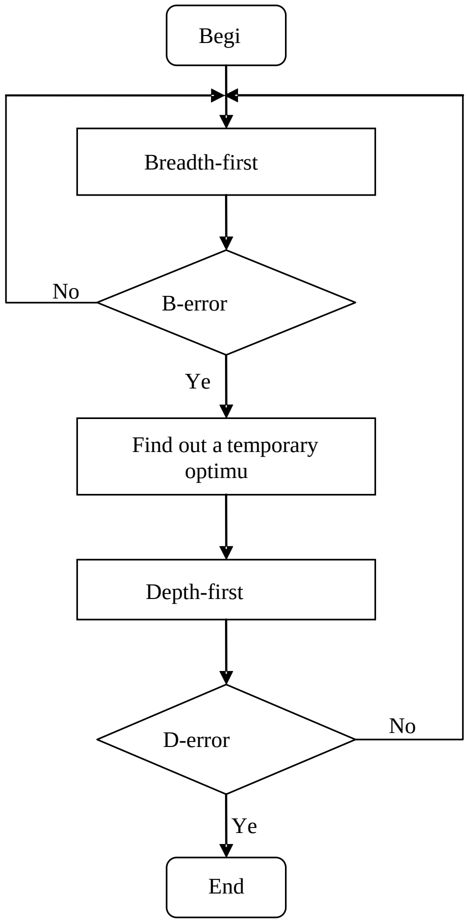

Fig. 2 Global Chaos Algorithms.

B. Optimization Algorithm

All of optimal search algorithms have only one criteria finding out the global optimum in a large space with the shortest computing time. There are many of different algorithms with different ideas such as Genetic Algorithms (GA) based on gene theory, simulated annealing based on solid annealing, and recently particle swarm algorithms based on behavior of birds or ants. This work proposes an algorithm that simpler than the others based on chaos theory. The main idea of the algorithm divides whole search space into a breadth space and a depth space. The mission of breadth search finds out a temporary optimum in a short time. After that, the depth search looks for the small space around the temporary optimum. Both two procedures are iterated until catch the global optimum as shown in Fig. 2. The flexible characteristic of proposal is the breadth level and the depth level can be controlled and adapted with specific problems by tolerated errors of each ones called B-error and D-error correspondently. While B-error is a low level of temporary tolerate, D-error is a final tolerate of the algorithm.

Many functional optimization problems can be formulated as following:

Min f(X), X=[x_1,x_2,…,x_n]

s.t. $x_i \in [a_i\ b_i]$, i=1,2,…,n,

where f is the objective function, and X is decision vector consisting of n variables.

The first stage is a breadth search to tolerate a temporary error shown in Fig. 3.

Set number of iteration k =1 and a temporary error

While

Step 1) Initialize $\gamma_i(k)$

$$\gamma_i(k) = \mu\gamma_i(k-1)(1-\gamma_i(k-1))$$

Step 2) chaos variable γ *is mapped into the variance ranges* $x_i \in [a_i, b_i]$ *of optimal variables by the following equation:*

$$x_i(k) = a_i + \gamma_i(k)(b_i - a_i) \qquad (2)$$

Step 3) Computing objective function f(X)

Until f reaches B-error and obtains X^* *as a temporary optimum.*

The second stage is a depth search to tolerate an final accurate error of problem. During this procedure, if the

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

algorithm is trapped at a local optimum it goes back the first stage, shown in Fig. 4.

While

Step 4) decrease current error slightly into tolerated error.

$$\textit{Current error} = \eta \textit{ B-error} \tag{3}$$

where η *is a decrease factor.*

While

Step 5) Perform chaos search

$$\gamma_i(k) = \mu \gamma_i(k-1)(1-\gamma_i(k-1)) \tag{4}$$

Step 6) Search around $x_i \in [\alpha_i \beta_i]$

$$x(k) = x^* + \alpha_i + \gamma(k)(\beta_i - \alpha_i) \tag{5}$$

where $[\alpha_i \beta_i]$ *is a narrow search interval of variable* x_i

Step 7) Computing objective function f(X)

Step 8) Checking the limit number of iterations for a local optimum violated and going back the first stage if it is violated.

End

Until f reaches D-error, final tolerate error.

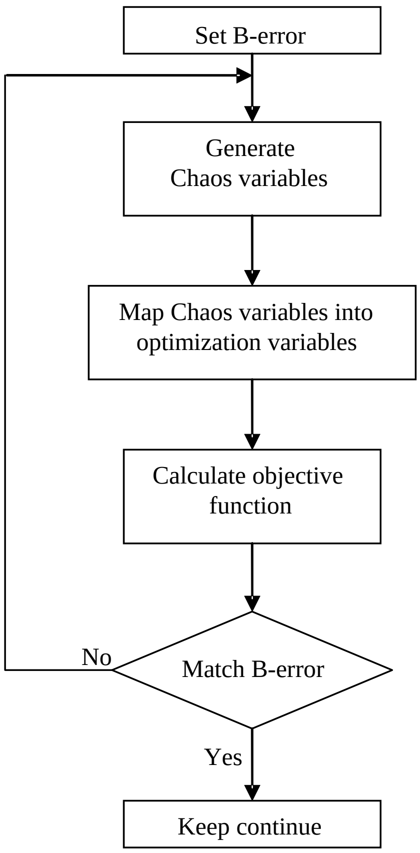

Fig. 3 The Breadth-first search algorithm.

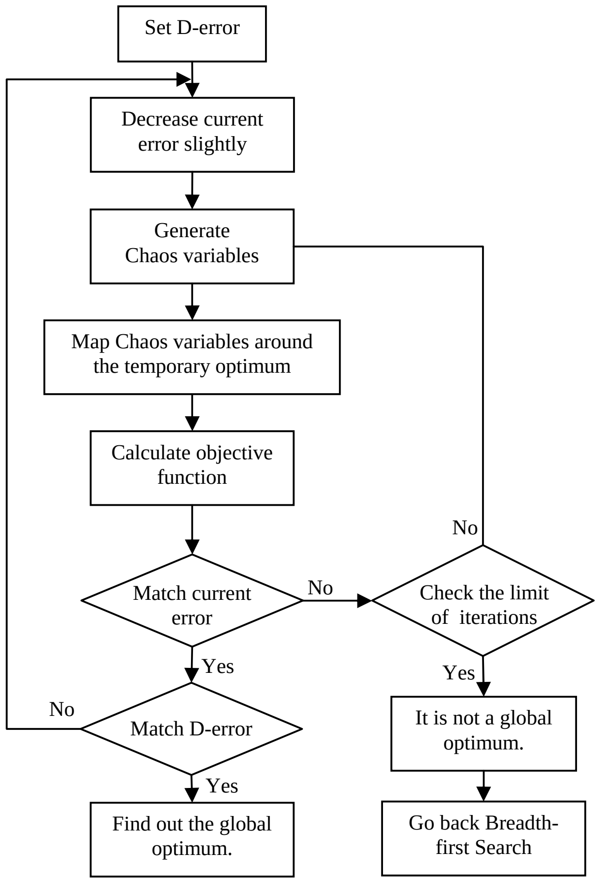

Fig. 4 The Depth-first search algorithm.

III. Training Multilayer Neural Networks

The standard Mutilayer Neural Network is usually 3 layers following: Input layer: X= $\{x_1, x_2, \ldots, x_{Ni}\}$, hidden layer and output layer Y. All neurons are connected by weights w and their output values are calculated through net value and transfer function, shown in Fig. 5.

Net value: $$net = \sum_{i=1}^{N_i} w_{ij}.x_i + \theta \tag{6}$$

Transfer function is usually used a saturation function such sigmoid function.

The objective function is based on least mean square error:

$$E = \frac{1}{2}\sum_{p=1}^{N}(d_p - y_p)^2 \tag{7}$$

The backpropagation algorithm is a well known learning algorithm for Multilayer Neural Networks:

$$\Delta_p w_{ij} = \eta \delta_{pj}(d_{pi} - y_{pi}) \tag{8}$$

η is the learning rate and δ_{pj} is derivative of the error between output y and desired output d at the j_{th} node for the input pattern p.

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

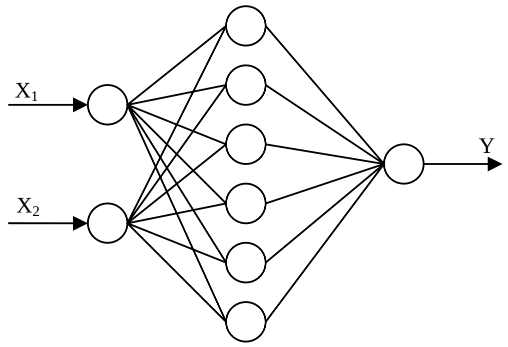

Fig. 5 Conventional Multilayer Neural Network model.

IV. ILLUSTRATIONS

A. Camel function:

$$f(x,y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2 \qquad 10<x,y<10 \qquad (9)$$

The Camel function has six local minima and two global minima at (x,y)=(-0.0898,0.7126),(0.0898,-0.7126) and the optimal objective function f=-1.031628

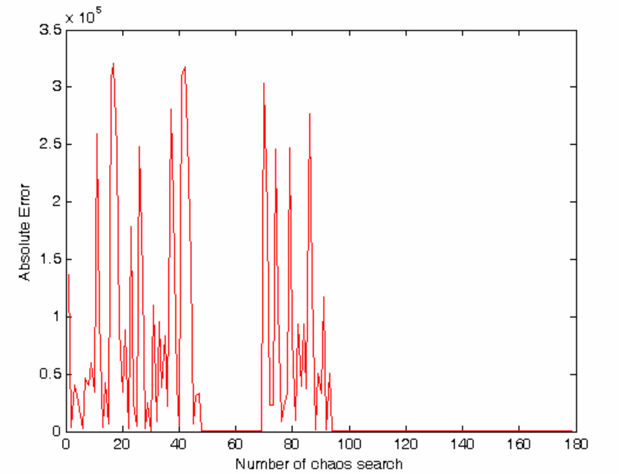

Fig. 6 Camel function process.

Fig. 6 shows a typical run after around 180 chaos generations. The final value of objective function f= -1.0313 with absolute error = 1.7169e-004 and solution (x,y)=(0.0832, -0.7123). The process jumped out one local optimum and reach global optimum at the second try.

Camel function is used to compare GCO with Genetic Algorithms (GA), one of the most well-known optimal algorithms. This illustration shows the success rate of both of algorithms under the same number of searches. With GCO, the number of generated searches is the maximum of chaos searches. With GA, this equals to number of individuals multiplying number of maximum generation in each run. In this case, the maximum of generation is const 100 and the number of individual runs from 10 to 100. This is equivalent to the number of chaos search running from 1000 to 10000 times. The results shown in Fig.7 and Fig. 8 indicate that GA has a low success rate when the number of individual is small. Furthermore, the computing time of GA takes much more than GCO for crossover, selection, mutation, reproduction. The most essential difference is that GA takes long time to calculate full of individuals while GCO only takes short time to search around the best individual.

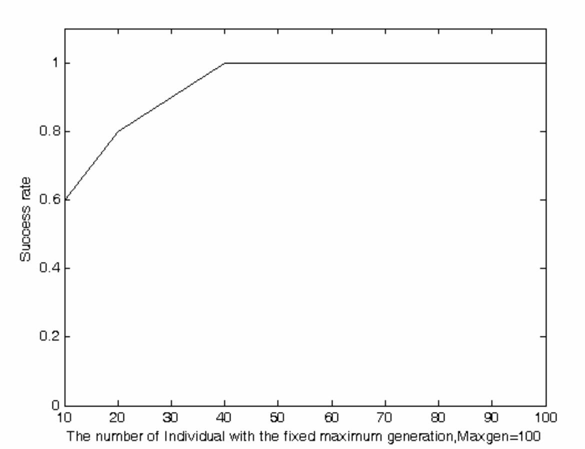

Fig. 7 Success rate of GA with maximum generation = 100, and number of individual from 10 to 100.

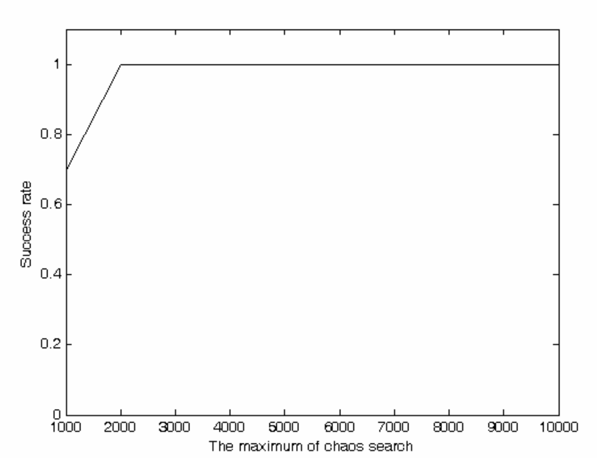

Fig. 8 Success rate of GCO with maximum of chaos search from 1000 to 10000.

B. Schaffer function:

$$f(x,y) = -\left[0.5 - \frac{\sin^2\sqrt{x^2+y^2} - 0.5}{(1+0.001(x^2+y^2))^2}\right] \qquad -4<x,y<4 \quad (10)$$

The Schaffer function has infinite local minima and one global minimum at (x,y)=(0,0) and objective function f=-1.

Fig. 9 shows a typical process carrying 120 runs. The processing jumped over 4 local optimum and stop at the final value of objective function f= -0.9997 with absolute error= 2.7151e-004 and the solution (x,y)=(-0.0084, 0.0142) after three tries.

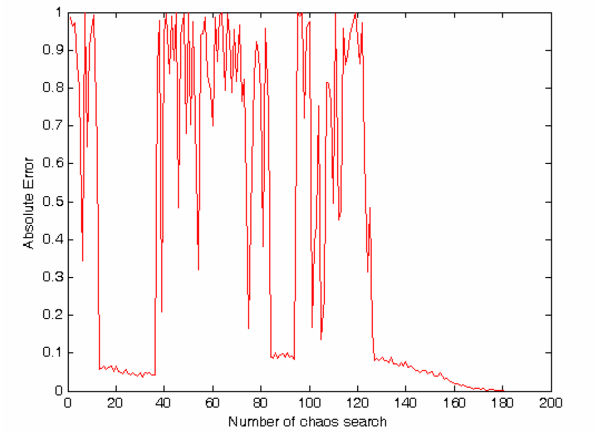

Fig. 9 Schaffer function process.

C. Training Multilayer Neural Networks

Training Multilayer Neural Networks is an optimization problem of multi variable. XOR is a typical problem to study characteristics of training Neural Networks. Its optimization surface is irregular and there are many local optimums. So it is difficult to find a global optimum.

Conventional Neural Network is used with three layers, 6 hidden neurons, logsig transfer function for every neuron shown in Fig. 5. Training the network is as optimal problem of 6x3=18 variables, the objective function is the least mean square error between network outputs and desire outputs.

The conventional model with the gradient back-propagation training algorithms is completely very slow to reach the tolerated error. With 10000 epochs, it can not reach 10^{-3} tolerated error shown in Fig. 10.

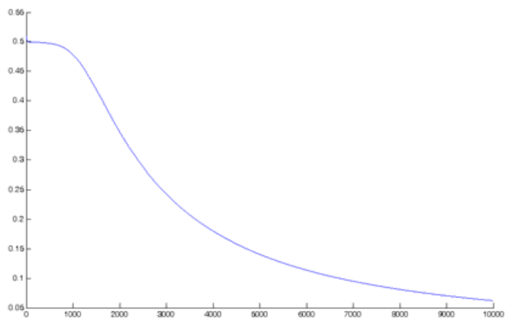

Fig. 10 Least mean square error of conventional model with Backpropagation training process.

Using chaos optimization to solve this problem, the range of wide search is set [-50+50] and the range of deep search is set [-2+2]. Obtained result is shown in Fig. 11.

XOR problem is used to examine as an n-dimensional optimal problem. The number of variable is three times as many as the number of hidden-layer neurons. With 20 hidden-layer neurons, the optimal problem has 60 variables and the success rates after 10 trials are shown in Fig. 12. This illustration determines the algorithm can work so efficiently in high dimensional optimal problem.

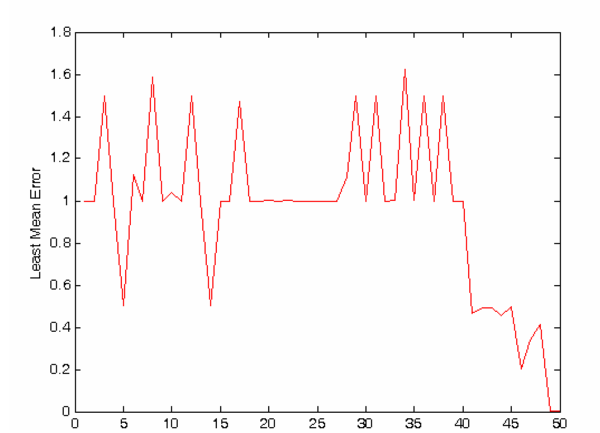

Fig. 11 Chaos optimization process of XOR problem with number of chaos search.

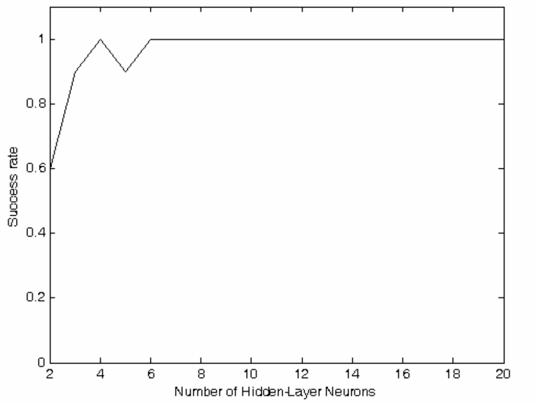

Fig. 12 Examining n-dimension optimization problem of GCO.

One of the most importance pivots of GCO is control of 2 steps, “Breadth-first search” and “Depth-first search”, shown in Fig. 13. The condition to evaluate them is tolerated errors, B-error and D-error. For much more complex nonlinear optimal problems, finding a suitable B-error is not easy. The following illustration shows the B-error from 0.1 to 1. With the B-error form 0.1 to 0.4 the problem gets 100% success rate but the rest one is not like this.

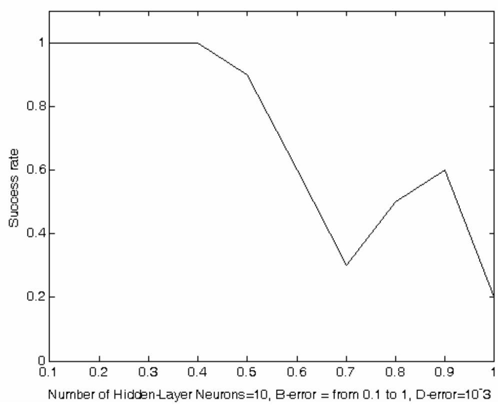

Fig. 13 Success rate when B-error from 0.1 to 1, and D-error is fixed 0.001

V. CONCLUSION

In principle, chaos optimization always reaches to the global minimum and more easily escape from local minima than other stochastic optimization algorithms. To control 2 procedures, Breadth-first search and Depth-first search, suitable parameters need to be defined. The wide level and the deep level are flexible and adaptive depending on specific problems. With two benefits, simple calculating and global power of chaos optimization, the algorithm definitely takes much faster than other evolution programming.

REFERENCES

[1] Masahiro Nakagawa, *Chaos and Fractals in Engineering*, World Science, Singapore, 1999.

[2] Luonan Chen, K. Aihara, "Global searching ability of chaos neural networks", *IEEE Trans. Circuits Syst.*, 46.8, 1999.

[3] Isao Tokuda, Kazuyuki Aihara, and Tomomasa Nagashima, *Adaptive annealing for chaotic optimization, Phys. Rev. E*, 58.4, 1998.

[4] Mohammad Saleh Tavazoei and Mohammad Haeri, "An optimization algorithm based on chaotic behavior and fractal nature", *J. Comput. Appl. Math.*, 2006.

[5] Dixiong Yang, Gang Li and Gengdong Cheng, "On the efficiency of chaos optimization algorithms for global optimization", *J. Chaos, Solitons & Fractals*, 2006.

[6] Guo Zilong, Wang Sun'an and Zhuang Jian, "A novel immune evolutionary algorithm incorporating chaos optimization", *Pattern Recognition Lett.*, 27.1, 2006,pp. 2-8.

[7] Shengsong Liu; Zhijian Hou, "Weighted gradient direction based chaos optimization algorithm for nonlinear programming problem", *Proc. Intelligent Control and Automation*, 3, 2002, pp. 1779 – 1783.

[8] You Yong; Sheng Wanxing; Wang Sunan; "Study of chaos genetic algorithms and its application in neural networks", *Proc. Computers, Communication, Control and Power Eng.TENCON*,, 1 2002. pp. 232 – 235.

[9] Rongbin Qi; Feng Qian; Shaojun Li; Zhenlei Wang; "Chaos-Genetic Algorithm for Multiobjective Optimization", *Proc. Intelligent Control and Automation WCICA*, 1, 2006, pp 1563 – 1566.

[10] LU Hui-juan, ZHANG Huo-ming, MA Long-hua, A new optimization algorithm based on chaos, *J. Zhejiang University sci. A*, 7.4, 2006, pp. 539-542.

[11] Liu Shengsong; Hou Zhijian; Wang Min, "A hybrid algorithm for optimal power flow using the chaos optimization and the linear interior point algorithm", *Proc. PowerCon*, 2, 2002, pp. 793 - 797

[12] Ji MJ, Tang HW, "Application of chaos in simulated annealing", *J. Chaos, Solitons & Fractals*, 21, 2004, pp. 933-41.

[13] Liu B, Wang L, Yin HY, Tang F, Huang DX, "Improved particle swarm optimization combined with chaos", *J. Chaos, Solitons & Fractals* 25, 2005, pp. 1261-71.

[14] Behera, L., Kumar, S., Patnaik, A., "On Adaptive Learning Rate That Guarantees Convergence in Feedforward Networks", *IEEE Trans. Neural Networks*, 17.5, 2006, pp. 1116–1125.

[15] Yu, X., Onder Efe, M., Kaynak, O., "A backpropagation learning framework for feedforward neural networks", *IEEE Int. Syposium Circuits Sys. ISCAS*, 3, 2001, pp.700–702.

[16] Terrence L. Fine, *Feedforward neural network methodology*, Springer, New York, 1999, pp. 130-131.

[17] K. Bertels1, L. Neuberg1, S. Vassiliadis and D.G. Pechanek, "On Chaos and Neural Networks: The Backpropagation Paradigm", *Artificial Intelligence Rev.* 15, 2001, pp.165–187.

Truong Quang Dang Khoa was born in the South of Vietnam in 1979. He graduated in 2002, received the Master Degree and became a lecturer of the Hochiminh University of Technology in 2004. Now he is working as a research assistant in Nagaoka University of Technology under the support of COE program. His research interests are Artificial Intelligent Computation, and Power System Analysis. He is a member of IEEE.

Masahiro Nakagawa was born in Kyoto, Japan in 1958. He received the B.Sc. and M. Sc. degrees from Nagaoka University of Technology, in 1980 and 1982, respectively. He also received Ph.D degree Nagoya University in 1988. He has been a Professor at the Nagaoka University of Technology since 2001. His recent activity is concerned with the fractals and chaos in science and engineering, e.g. the chaos in EEG and vocal sounds, the chaos neural networks with a periodic activation function, the iterated function system for image coding etc. Also he is interested in the physics of liquid crystals involving the molecular theory and the viscoelastic continuum theory. He is a member of the Japanese Neural Network Society, the Physical Society of Japan, and Japanese Society of Applied Physics.