2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Optimization using Higher-Order Chaotic Neural Networks

Taro Kuroda and Mikio Hasegawa
Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science, TUS
Tokyo, Japan
kuroda@haselab.ee.kagu.tus.ac.jp, hasegawa@ee.kagu.tus.ac.jp

Abstract— **Chaotic neural networks have been applied to various combinatorial optimization problems, and effectiveness of chaotic dynamics for solution search has been shown. However, the conventional chaotic neural networks based on the Hopfield-Tank neural network can solve the problems whose objective function is only second or lower-order function of the state of the neurons. In order to apply the chaotic optimization framework to more general problems, we introduce the higher-order neural networks which have higher-order connections and energy function. We verify effectiveness of a chaotic method on such a higher order neural network by comparing its performances with gradient dynamics and stochastic dynamics.**

I. INTRODUCTION

Effectiveness of the chaotic dynamics on optimization problems has been shown by various experimental results. There are several approaches, such as the methods using chaotic dynamics on the Hopfield neural networks [1,2], the chaotic dynamical tabu searches [3,4], and so on. The Hopfield neural network has property that its energy function monotonically decreases, and has been applied to various optimization problems [5]. However, the state of this neural network converges to an undesirable local minimum. In order to improve the performance of the Hopfield neural network, the chaotic neural network has been applied [1,2], whose chaotic behavior moves its state and avoid trapping at a local minimum. The solution generated by the chaotic neural network is better than the stochastic searches such as the Boltzmann machines or simulated annealing. Such an optimization method using the chaotic neural network can be implemented on electric circuits which have super high-speed computational ability. Horio et al. have realized a chaotic neural network circuit which has 400 chaotic neurons and reported its great performance [6].

Although the chaotic neural network has high performance on combinatorial optimization problems, the conventional chaotic methods were based on the Hopfield neural network, whose energy function is only the second order products of the state of the neurons. By extending this framework to a higher order energy function, we will be able to solve more general problems by chaotic dynamics. As an example of the higher order combinatorial optimization problems, autonomous and decentralized radio resource usage optimization to minimize difference of the throughput among the mobile terminals becomes fourth order objective function [7,8]. As another example, an optimal energy function for the Traveling Salesman Problems becomes higher order energy function [9].

In order to apply effective chaotic dynamics to such general problems, we introduce the higher order neural networks [10] which have higher order connections and energy function. We apply the chaotic dynamics to the higher order neural network and evaluate effectiveness of the proposed framework on artificial problems.

II. HIGHER ORDER CHAOTIC NEURAL NETWORKS

A. Higher order energy function

The update equation of the conventional first order Hopfield neural network is given as follows,

$$x_i(t+1) = \begin{cases} 1 & \text{if} \sum_{j=1}^{N} W_{ij} x_j(t) + \theta_i > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where $x_i(t)$ is the output of the i th neuron at time t, W_{ij} is the connection weight between the i th and the j th neurons, θ_i is the threshold of the i th neuron, respectively. The energy function of this neural network which always decreases by each neuronal update can be defined as follows,

$$E(t) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} x_i(t) x_j(t) - \sum_{i=1}^{N} \theta_i x_i(t). \quad (2)$$

Equation (2) includes second or lower order products of the state of the neurons but does not include third or higher order products. Therefore, the Hopfield neural network can solve 0-1 integer programming problem with only second or lower order products. However, for example, the objective function for fair radio resource management in wireless network [7,8], that minimizes the difference of the throughputs assigned to each terminal can be defined as follows,

$$E_{\mathrm{W}}(t)=\sum_{i=1}^{N_m}\sum_{k=1}^{N_m}\sum_{m=1}^{N_m}\sum_{j=1}^{N_{Ap}}\sum_{o=1}^{N_{Ap}}\sum_{l=1}^{N_{Ap}}\frac{1}{C_jC_l}\{x_{mj}(t)x_{il}(t)x_{ol}(t)x_{il}(t)$$
$$-2x_{mj}(t)x_{ij}(t)x_{ol}(t)x_{kl}(t)+x_{mj}(t)x_{kj}(t)x_{ol}(t)x_{kl}(t)\}. \tag{3}$$

This is a fourth order objective function, which cannot be solved by the chaotic method based on the conventional Hopfield neural network.

B. Higher order neural networks

The higher order neural network has second or higher order mutual connections between neurons [10]. Its update equation can be defined as follows,

$$x_i(t+1)=\begin{cases}1 & \text{if } D_i(t)>0\\ 0 & \text{otherwise}\end{cases}, \tag{4}$$

$$D_i(t)=\sum_{j=1}^{N}\sum_{k=1}^{N}\cdots\sum_{l=1}^{N}w_{ijk\cdots l}x_j(t)x_k(t)\cdots x_l(t)$$
$$+\cdots$$
$$+\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N}w_{ijkl}x_j(t)x_k(t)\cdots x_l(t)$$
$$+\sum_{j=1}^{N}\sum_{k=1}^{N}w_{ijk}x_j(t)x_k(t)+\sum_{j=1}^{N}w_{ij}x_i(t)+\theta_i \,, \tag{5}$$

where $w_{ij\cdots k}$ are the higher order connection weights between i th, j th, … , and k th neurons. The energy function of the higher order neural network, which decreases autonomously by neuronal updates using (4) and (5), becomes as follows,

$$E(t)=-\frac{1}{m+1}\sum_{i=1}^{N}\sum_{j=1}^{N}\cdots\sum_{k=1}^{N}w_{ij\cdots k}x_i(t)x_j(t)\cdots x_k(t)$$
$$-\cdots$$
$$-\frac{1}{3}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}w_{ijk}x_i(t)x_j(t)\cdots x_k(t)$$
$$-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}x_i(t)x_j(t)-\sum_{i=1}^{N}\theta_i x_i(t) \,, \tag{6}$$

where *m* is the order of the neural network. Although we can solve various problems by the higher order neural network, it stops search at an undesirable local minimum similar to the conventional Hopfield neural networks.

C. Higher order chaotic neural networks

The chaotic neural network can avoid trapping at undesirable local minimum by its chaotic behaviors. Compared to stochastic searches, such as Boltzmann machines or simulated annealing, effectiveness of the chaotic search has been shown by many experimental results [1,2]. The chaotic neurons in the chaotic neural network have analog output function and refractoriness, which make it hard for neurons to fire after its previous firings. The update equations of the chaotic neural network are given by the following equations,

$$\eta_i(t+1)=k_m\eta_i(t)+\sum_{j=1}^{N}W_{ij}x_j(t)+\theta_i, \tag{7}$$

$$\zeta_i(t+1)=k_r\zeta(t)-\alpha(x_i(t)-r), \tag{8}$$

$$x_i(t+1)=f[\eta_i(t+1)+\zeta_i(t+1)], \tag{9}$$

$$f(z)=\frac{1}{1+\exp(-z/\varepsilon)}, \tag{10}$$

where $\eta_i(t)$ and $\zeta_i(t)$ are the feedback inputs and the internal states of the *i* th neuron at time *t*, k_m and k_r are the decay parameters of the feedback inputs and the internal states, α is the scaling parameter of the refractory effects, *r* is the positive bias, ε is a parameter of the sigmoid function, respectively.

When the value of each parameter was set appropriately, the chaotic neural network continues search of the global minimum with avoiding trapping at local minima by chaotic dynamics. However, the chaotic search based on the conventional Hopfield neural network cannot deal with the energy function of the third or higher order products of the neurons.

To introduce higher order connections and energy functions to chaotic neural network, we define the following equations to update each neuron,

$$\eta_i(t+1) = k_m \eta_i(t) + D_i(t)\,, \tag{11}$$

$$\zeta_i(t+1) = k_r \zeta(t) - \alpha(x_i(t) - r)\,, \tag{12}$$

$$x_i(t+1) = f[\eta_i(t+1) + \zeta_i(t+1)]\,, \tag{13}$$

$$f(z) = \frac{1}{1+\exp(-z/\varepsilon)}\,. \tag{14}$$

We call this a higher order chaotic neural network.

III. Evaluation of the Performance of Higher Order Chaotic Neural Networks

A. A higher order assignment problem

In order to verify the effectiveness of the chaotic dynamics on higher order problems, we evaluate the performance of the proposed higher order chaotic neural networks on an artificial higher order combinatorial optimization problem. Here, we define a higher order assignment problems that searches an optimum permutation **P** which minimizes an objective function $E_{\mathrm{HAP}}(\mathbf{P})$,

$$E_{\mathrm{HAP}}(\mathbf{P}) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_d=1}^{n} a_{i_1 i_2 \cdots i_d} b_{p(i_1) p(i_2) \cdots p(i_d)}\,, \tag{15}$$

$$\mathbf{P} = [p(1), p(2), \cdots, p(d)]\,, \tag{16}$$

where *d* is the order of the problem, $a_{ij\cdots k}$ is the $(i, j, \cdots, k)$ th value in a high dimensional matrix **A** , $b_{ij\cdots k}$ is the $(i, j, \cdots, k)$ th value in a high dimensional matrix **B**. Here, we assume that **A** and **B** are symmetric matrices, and their diagonal elements are zero.

To solve this problem by the higher order chaotic neural networks, first we map the state of the permutation **P** on the state of the neural network as shown in Fig.1.

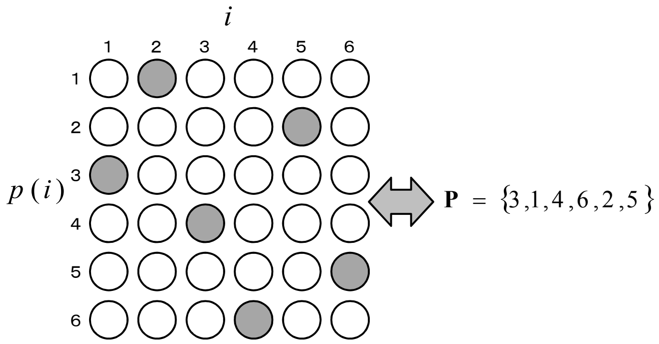

Figure 1. Mapping the higher order assignment problem on the higher order neural network.

When a given problem was the third order assignment problem, it can be solved by minimizing the following energy function,

$$E_{\mathrm{OBJ}}^{\mathrm{HAP}}(t) = \lambda \left\{ \sum_{i=1}^{N} \left(\sum_{j=1}^{N} x_{ij}(t) - 1 \right)^2 + \sum_{j=1}^{N} \left(\sum_{i=1}^{N} x_{ij}(t) - 1 \right)^2 \right\} + \mu \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} a_{ikm} b_{jln} x_{ij}(t) x_{kl}(t) x_{mn}(t)\,, \tag{17}$$

where λ and μ are the weights of the constraint term and the objective function term, respectively. From (6) and (17), the connection weights and thresholds can be obtained as follows,

$$w_{(ij)(kl)(mn)} = -3\mu a_{ikm} b_{jln}\,, \tag{18}$$

$$w_{(ij)(kl)} = -2\lambda \{ \delta_{ij}(1-\delta_{kl}) + \delta_{kl}(1-\delta_{ij}) \}, \tag{19}$$

$$\theta_{(ij)} = 2\lambda\,. \tag{20}$$

By applying these connection parameters to (5) and (11)—(14), we can solve the third order assignment problems by simple updates of the higher order chaotic neural networks.

B. Comparison with other techniques

Solvable performances of the proposed higher order chaotic neural network were compared with the conventional higher order neural network with gradient dynamics and that with the Boltzmann machines as a stochastic search. To update the state of Boltzmann machines for the higher order neural network, $\Pr_i(t)$, which is the probability for the *i* th neuron to fire($x_i(t) = 1$) at time *t*, was defined as follows,

$$\Pr_i(t+1) = \frac{1}{1+\exp(-D_i(t)/T(t))}, \quad (21)$$

$$T(t) = T_0\gamma^t, \quad (22)$$

where $T(t)$ is the temperatures at time *t*, T_0 is an initial value of temperature and γ is the cooling schedule, respectively.

We introduce problems whose sizes are 5 to 10. We used the parameter values shown in Table 1.

TABLE I. TABLE TYPE STYLES

	N=5	N=6	N=7	N=8	N=9	N=10
λ	1.60	2.70	4.00	5.50	7.20	9.90
μ	0.01	0.01	0.01	0.01	0.01	0.01
T_0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
γ	0.991	0.991	0.991	0.991	0.991	0.991
k_m	0.7	0.7	0.7	0.7	0.7	0.7
k_r	0.9	0.9	0.9	0.9	0.9	0.9
α	1.0	1.0	1.0	1.0	1.0	1.0
r	0.15	0.15	0.15	0.15	0.15	0.15
ε	0.02	0.02	0.02	0.02	0.02	0.02

Fig.2 shows the average solutions of the conventional higher order neural network, the higher order neural network with Boltzmann machines and the higher order chaotic neural networks. The cutoff time of each run is 1000 iterations. From the results, the higher order chaotic neural networks have better solutions than the gradient and the stochastic dynamics in each problem size. Although the stochastic dynamics is better than the gradient dynamics, the higher order chaotic neural networks perform even better than the stochastic search. We confirm that the chaotic dynamics is effective even for the higher order combinatorial problems.

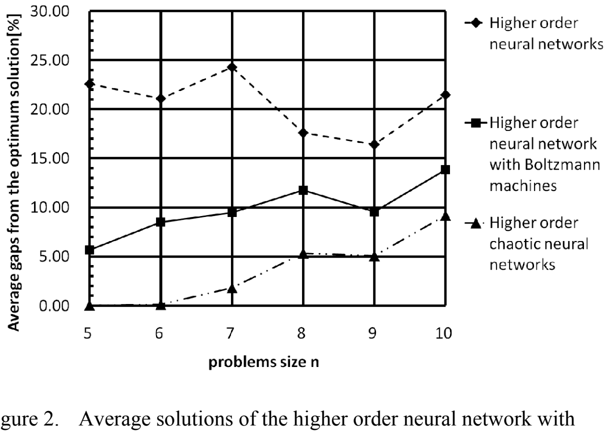

Figure 2. Average solutions of the higher order neural network with gradient dynamics, Boltzmann machines and chaotic dynamics.

IV. CONCLUSION

In this paper, we proposed a higher order chaotic neural network and showed its performance better than the conventional or the stochastic higher order neural networks. This framework is more generally applicable to various combinatorial optimization problems, which could not be solved by the conventional Hopfield neural network approach. For example, fair radio resources management in wireless networks [7,8] is a higher order combinatorial optimization problem, which can be solved by this new proposed approach. We have also applied the dynamics of the higher order chaotic neural network to factorization into prime number, whose objective function becomes third order function. The proposed approach using higher order chaotic neural network enables application of effective chaotic search based on simple updating to various complicated optimization problems.

REFERENCES

[1] H. Nozawa, “A neural network model as a globally coupled map and applications based on chaos,” Chaos, vol. 2, pp. 377–386, 1992.

[2] L. Chen, K. Aihara, “Chaotic simulated annealing by a neural network model with transient chaos,” Neural Networks, vol. 8, pp. 915–930, 1995.

[3] M. Hasegawa et al., “A Chaotic Neuro dynamical Approach to Combinatorial Optimization Problems,” Physica Review Letters, vol. 79, no. 12, pp. 2344–2347, 1997.

[4] M. Hasegawa et al. , “A novel chaotic search for quadratic assignment problems,” European Journal of Operational Research, vol. 139, pp. 543–556, 2002.

[5] J. J. Hopfield and D.W. Tank, “Neural computation of decisions in optimization problems,” Boil. Cybern., vol. 52, no. 3, pp. 141–152, 1985.

[6] Y. Horio, T. Ikeguchi and K. Aihara, “A Mixed Analog/Digital Chaotic Neuro-Computer System for Quadratic Assignment Problems,” Neural Networks, vol. 18, pp. 505–513, 2005.

[7] M. Hasegawa et al., “Autonomous and Decentralized Optimization for Fair Radio Resource Management for Cognitive Wireless Clouds,” IEICE Technical Report SR2007-108, pp. 159–166, 2008.

[8] M. Hasegawa et al., “Autonomous and Decentralized Optimization of Large-Scale Heterogeneous Wireless Networks by Neural Network Dynamics,” IEICE Trans. of Commun., vol. E91-B, no. 1, pp. 101–108, 2008.

[9] S. Matsuda, “"Optimal" Hopfeild network for combinatorial optimization with linear cost function”, IEEE Trans. on Neural Networks, vol. 9, pp. 1319–1330, 1998.

[10] B.S.Cooper, “Higher Order Neural Networks for Combinatorial Optimization - Improving the Scaling Properties of the Hopfield Network,” Proc. of IEEE International Conference on Neural Networks, vol. 4, pp. 1855–1860, 1995.