2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Lyapunov Exponents of Chaotic Neural Network with Dynamical Noise for Solving Quadratic Assignment Problem

Takayuki Suzuki†, Shun Motohashi†, Takafumi Matsuura†, and Tohru Ikeguchi‡

Graduate School of Science and Engineering, Saitama University
255 Shimo-Ohkubo, Saitama, 338-8570 Japan
Email: †{suzuki, motohashi, takafumi}@nls.ics.saitama-u.ac.jp, ‡tohru@mail.saitama-u.ac.jp

Abstract—The quadratic assignment problem (QAP) is one of the most difficult NP-hard combinatorial optimization problems. To solve the QAP, various approximate algorithms for finding near optimal solutions have already been proposed. Among them, a method which uses the Hopfield neural network (HNN) can be applied to find solutions. However, this method cannot always offen gets good performance because it stuck at local minima. To avoid local minima, a method which uses chaotic neural network (CNN) has already been proposed. The method with CNN can solve the QAP effectively and shows good performance. On the other hand, to avoid undesirable local minima, it is possible to inject dynamical noise to a solver. Then, we have already proposed a method which uses both chaotic dynamics and dynamical noise for avoiding local minima. The result shows that when a small amount of dynamical noise is added, the performance becomes high. In this paper, we investigate the performance of the method using several types of dynamical–stochastic and deterministic noise. To analyze the proposed method, we investigate the relation between the performance and the Lyapunov exponents of the CNN with such dynamical noise.

1. Introduction

In the real world, various combinatorial optimization problems exist, for example, VLSI design, scheduling problem, routing problem, facility layout problem, and so on. It is important to obtain optimal solutions of these problems because operation costs can be reduced. These kind of combinatorial problems can be formulated as a quadratic assignment problem (QAP)[1]. It is almost impossible to find an optimal solution in reasonable time, because the QAP is classified into an NP-hard problem. Then, it is required to develop approximate algorithms for finding near optimal solutions in reasonable time.

As the approximate algorithm, a method which uses the mutual connection neural networks or the Hopfield neural network (HNN) has already been proposed[2]. In this method, a firing pattern of HNN represents a solution of the QAP. If we decide good synaptic weights of HNN for solving the QAPs, we can obtain a good solution by descent down-hill dynamics of HNN. However, this method gets trapped into local minima. Then, to avoid local minima, a method which uses chaotic neural network (CNN)[3] has already been proposed [4, 5]. Chaotic dynamics of CNN works to avoid the local minima effectively. As another method for avoiding local minima, a method which injects dynamical noise into HNN for solving combinatorial optimization problems has also been proposed [8, 9]. Using fluctuation of dynamical noise, this method can escape from local minima, and often exhibits good performance. We have already proposed a new algorithm for solving QAP by combining these two basic strategies chaotic dynamics and dynamical noise[6, 7]. As a result, the combination of chaotic dynamics and dynamical noise leads to better performance.

On the other hand, in Refs.[8, 9], it is shown that a method, which uses the HNN with intermittency chaotic noise near period three window of the logistic map, shows higher performance. It is pointed out that intermittency chaotic noise near period three window of the logistic map offers better solution than other type of chaotic noise[8, 9]. Then, in this paper, first, we investigate the performance of combination of chaotic dynamics and chaotic noise for solving QAPs. Next, to analyze the proposed method, we calculate Lyapunov exponents of CNN with dynamical noise, and investigate the relation between the Lyapunov exponents and the solving performance.

2. Solving Quadratic Assignment Problem with Hopfield Neural Network

2.1. Quadratic Assignment Problem

The quadratic assignment problem (QAP) is one of the NP-hard combinatorial optimization problems. The goal of the QAP is to find an optimal location of facilities to cities that to minimize the total cost. The QAP is described as follows: given two $N \times N$ matrices, a distance matrix D and a flow matrix C, find a permutation $\boldsymbol{p}$ which minimizes a value of the following objective function $F(\boldsymbol{p})$:

$$F(\boldsymbol{p}) = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} c_{p(i)p(j)}, \qquad (1)$$

where d_{ij} is the (i, j)th element of D, $p(i)$ is the ith element of $\boldsymbol{p}$, $c_{p(i)p(j)}$ is the $(p(i), p(j))$th element of C, and N is a size of the problem. The QAP belongs to the NP-hard problems. Thus, it is required to develop an effective approximate algorithm for finding near optimal solutions in a reasonable time frame.

2.2. Hopfield Neural Network for Solving QAP

As an approximate algorithm for solving the QAP, a method which uses the descent down-hill dynamics of the Hopfield neural network (HNN) has already been proposed[2]. In this method, to solve an N-size QAP, $N \times N$ neurons are prepared. The (i, m)th neuron corresponds an assignment of the ith facility and the mth city. If the ith facility is assigned to the mth city, an output of the (i, m)th neuron x_{im} is 1, otherwise, 0. The output pattern of the HNN $X = (x_{im})$ represents a solution of QAP. In the HNN, the energy function is defined as follows:

$$\begin{aligned} F(X) &= A\sum_{i=1}^{N}(\sum_{m=1}^{N} x_{im} - 1)^2 \\ &+B\sum_{m=1}^{N}(\sum_{i=1}^{N} x_{im} - 1)^2 \\ &+\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} d_{ij}c_{mn}x_{im}x_{jn}, \end{aligned} \tag{2}$$

where A and B are positive constants. By using Eq.(2), synaptic weights between the (i, m)th neuron and the (j, n)th neuron $w_{im;jn}$ and thresholds of the (i, m)th neuron θ_{im} are defined as follows:

$$\begin{aligned} w_{im;jn} &= -2\{A(1-\delta_{mn})\delta_{ij} + B\delta_{mn}(1-\delta_{ij}) \\ &+\frac{d_{ij}c_{mn}}{q}\}, \end{aligned} \tag{3}$$

$$\theta_{im} = -(A+B), \tag{4}$$

where δ_{ij} is Kronecker's delta and q is a normalization parameter. A state of the (i, m)th neuron is updated asynchronously using the following equation:

$$x_{im}(t+1) = g\left(\sum_{j=1}^{N}\sum_{n=1}^{N} w_{im;jn}x_{jn}(t) - \theta_{im}\right), \tag{5}$$

where g is an output function.

2.3. Hopfield Neural Network for Solving QAP with Dynamical Noise

The method which uses the HNN has a local minimum problem. To solve this problem, a method which adds dynamical noise to the HNN has already been proposed [8, 9]. An output of state the (i, m)th neuron (Eq.(6)) is redefined as follows:

$$x_{im}(t+1) = g\left(\sum_{j=1}^{N}\sum_{n=1}^{N} w_{im;jn}x_{jn}(t) - \theta_{im} + \beta z_{im}(t)\right) \tag{6}$$

where z_{im} is an additive noise and β is a weight of noise.

3. Proposed Method

3.1. Chaotic Neural Network for Solving Quadratic Assignment Problem

As another approach of avoiding the local minima, a method which uses chaotic dynamics produced from a chaotic neural network (CNN)[3] have been proposed [4, 5]. The CNN model is proposed by Aihara, Takabe and Toyoda[3]. This neural network model can reproduce a chaotic dynamics observed in real neural membrane.

An internal state of the (i, m)th neuron in CNN is defined as follows:

$$\begin{aligned} y_{im}(t+1) &= ky_{im}(t) + \sum_{j=1}^{N}\sum_{n=1}^{N} w_{im;jn}f(y_{jn}(t)) \\ &-\alpha f(y_{im}(t)) + \theta_{im}(1-k), \end{aligned} \tag{7}$$

where k is a decay parameter, α is a strength parameter of a refractory effect, and f is an output function. As an output function, a sigmoidal function is often used:

$$f(y) = \frac{1}{1+\exp(-\frac{y}{\epsilon})}, \tag{8}$$

where ϵ is a gradient parameter of the sigmoidal function.

Updating each neuron asynchronously, the CNN generates solutions. However, if we use outputs of the neurons, we cannot always obtain feasible solutions, because an output of the chaotic neuron takes an analog value. Then, we use the firing decision method[4] which can always generate a feasible solution for QAP. The procedure is described as follows:

1. Choose an index (i, m) whose internal state y_{im} takes the maximum value among all the neurons. Then, set the (i, m)th neuron as to firing state, and let $x_{im} = 1$.

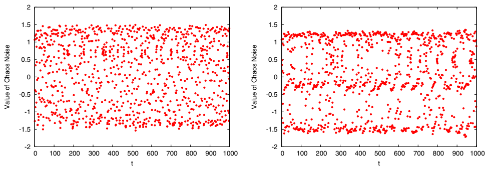
2. Set other neurons in the ith row and the mth column to a resting state, and let $x_{ik} = 0 (k \neq m)$ and $x_{ml} = 0 (l \neq i)$. Then, exclude neurons which have already been selected in Steps 1 and 2.

3. Repeat Steps 1 and 2 until all states of neurons are decided.

3.2. Chaotic Neural Network for Solving QAP with Dynamical Noise

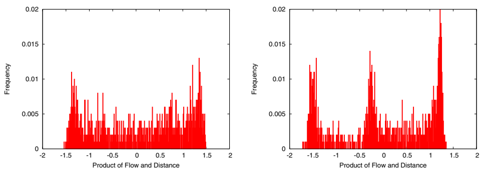
We have already been proposed the method which uses CNN and dynamical noise for solving QAP[6, 7]. An internal state of the (i, m)th neuron with dynamical noise is defined as follows:

$$\begin{aligned} y_{im}(t+1) &= ky_{im}(t) + \sum_{j=1}^{N}\sum_{n=1}^{N} w_{im;jn}f(y_{jn}(t)) - \alpha f(y_{im}(t)) \\ &+\theta_{im}(1-k) + \beta z_{im}(t), \end{aligned} \tag{9}$$



(a) $a = 4.0$ (b) $a = 3.8274$

Figure 1: Time series of chaotic noise.



(a) $a = 4.0$ (b) $a = 3.8274$

Figure 2: Frequency distributions of chaotic noise.

where β is a weight of dynamical noise and $z_{im}(t)$ is a sequence of dynamical noise added to the internal state of the (i, m)th neuron at time t. Then, we define a single iteration as updating all neurons asyncronusly.

4. Experimental results

4.1. Performance with Respect to Weight of Noise and Parameter of Chaotic Noise

To evaluate the performance of the proposed method which uses chaotic neural network (CNN) with dynamical noise, we use the benchmark problem from QAPLIB[11]. Parameters of the proposed method (Eq.(9)) are decided as $A = 0.34$, $B = 0.34$, $k = 0.87$, $\alpha = 1.01$, $\epsilon = 0.02$, $\theta = 0.68$ and $q = 1100$. Then, we use two kinds of dynamical noise: white Gaussian noise whose average is zero and variance is unity and chaotic noise. The chaotic noise is generated by a logistic map and normalized as follows:

$$\hat{z}_{im}(t+1) = a\hat{z}_{im}(t)(1 - \hat{z}_{im}(t)), \tag{10}$$

$$z_{im}(t) = \frac{\hat{z}_{im} - \overline{z}}{\sigma_z}, \tag{11}$$

where a is a parameter, $\overline{z}$ is the average of $\hat{z}_{im}(t)$ and σ_z is the standard deviation of $\hat{z}_{im}(t)$. When $a = 4.0$, the chaotic noise is fully developed chaos, and when $a = 3.8274$, the chaotic noise is intermittency chaos near the three period window. In the proposed method, we use $a = 4.0$ and $a = 3.8247$. Figure 1 shows examples of the chaotic noise with $a = 4.0$ and $a = 3.8274$. Figure 2 shows distribution of chaotic noise with $a = 4.0$ and $a = 3.8274$. The proposed method is applied for 3,000 iterations. Figure 3 shows results of the proposed method for Had12 when we change β from 0 to 0.008 with intervals of 0.0005. In Fig.3, the results are expressed by percentages of average gaps between obtained solutions and the optimal solutions for 50 trials. From Fig.3, adding small amount of dynamical noise, the proposed method shows higher performance. The performance of the proposed method also depends on types of dynamicsl noise. If the chaotic noise of $a = 3.8274$ is injected, better solutions are obtained than other types of dynamical noise, such as fully-developed chaos or Gaussian random numbers.

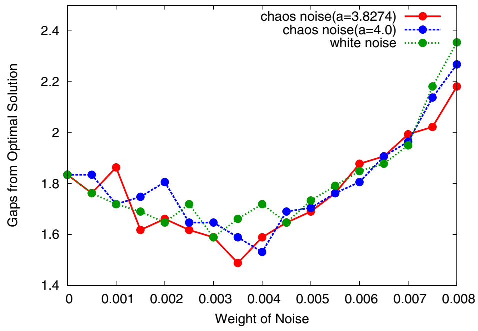


Figure 3: Percentages of average gaps between obtained solutions and the optimal solutions for 50 trials when noise weight β is changed for Had20.

4.2. Lyapunov exponent

4.2.1. Lyapunov exponents for the Synchronous Update

Next, we investigate the reason why the performance of the proposed method depends on the types of dynamical noise and a weight of dynamical noise. We calculate the Lyapunov exponents of the CNN with dynamical noise, with which the chaotic dynamics is quantified. To calculate Lyapunov exponents, we have to calculate the Jacobian matrix[10]. The Jacobian matrix of N^2-dimensional nonlinear map is described as follows:

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{y}_1} & \frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{y}_2} & \cdots & \frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{y}_{N^2}} \\ \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{y}_1} & \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{y}_2} & \cdots & \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{y}_{N^2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{F}_{N^2}}{\partial \boldsymbol{y}_1} & \frac{\partial \boldsymbol{F}_{N^2}}{\partial \boldsymbol{y}_2} & \cdots & \frac{\partial \boldsymbol{F}_{N^2}}{\partial \boldsymbol{y}_{N^2}} \end{pmatrix}, \tag{12}$$

where $\boldsymbol{F}_k$ is a nonlinear map of the kth neuron and $\boldsymbol{y}_l$ is an internal state of the lth neuron. Then, the Jacobian matrix of the CNN is calculated as follows:

$$\frac{\partial y_{ij}(t+1)}{\partial y_{mn}(t)} = \begin{cases} \frac{w_{ij;mn}}{\epsilon} f(y_{mn}(t))(1 - f(y_{mn}(t))), \\ \qquad ((ij) \neq (mn)), \\ k + \frac{(w_{mn;mn} - \alpha)}{\epsilon} f(y_{mn}(t))(1 - f(y_{mn}(t))), \\ \qquad ((ij) = (mn)). \end{cases} \tag{13}$$

If equations of the dynamical system are explicitly given, we can calculate the Jacobian matrix and calculate Lyapunov exponents with the QR decomposition[10].

4.2.2. Lyapunov exponents for the asynchronous update

When we calculate Lyapunov exponents of the asynchronously updating CNN, we cannot use the same Jacobian matrix, because updated neurons and un-updated neurons coexist in the same iteration. Thus, we must define an asynchronous-update version of the Jacobian matrix for the CNN. First, we consider $\boldsymbol{F}'_h$ as a nonlinear map for the hth update of neurons. The Jacobian matrix when the (r, s)th neuron is updated at the hth update is calculated as follows:

$$\boldsymbol{J}'_h(t) = \begin{cases} \dfrac{w_{im;jn}}{\epsilon} f(y_{jn}(t))(1 - f(y_{jn}(t))), & \\ \qquad ((im) \neq (jn) \cap (im) = (rs)), & \\ k + \dfrac{w_{im;jn} - \alpha}{\epsilon} f(y_{jn}(t))(1 - f(y_{jn}(t))), & \\ \qquad ((im) = (jn) \cap (im) = (rs)), & \\ 0, \qquad ((im) \neq (jn) \cap (im) \neq (rs)), & \\ 1, \qquad ((im) = (jn) \cap (im) \neq (rs)). & \end{cases} \tag{14}$$

The update function of internal states for a single iteration is represented as follows:

$$\begin{aligned} y(t+1) &= \boldsymbol{F}(y(t)) \qquad (15) \\ &= \boldsymbol{F}'_1 \boldsymbol{F}'_2 \ldots \boldsymbol{F}'_k \ldots \boldsymbol{F}'_{N^2}(y(t)). \end{aligned}$$

Thus, the Jacobian matrix for the asynchronously updating CNN in a single iteration is defined as follows:

$$\boldsymbol{J} = \boldsymbol{J}'_{N^2} \boldsymbol{J}'_{N^2-1} \ldots \boldsymbol{J}'_k \ldots \boldsymbol{J}'_2 \boldsymbol{J}'_1. \tag{16}$$

4.2.3. Result

Figure 4 shows results of the maximum Lyapunov exponents and the sum of positive Lyapunov exponents of CNN with dynamical noise for solving QAPs. In Fig.4, we change β from 0 to 0.008 with intervals of 0.001. From Fig.4, the maximum Lyapunov exponent does not change even though an amount of dynamical noise is changed. The sum of positive Lyapunov exponents decreases slightly as an amount of noise increases.

5. Conclusions

In this paper, we analyze the effect of dynamical noise added to the CNN when it solves QAP. We investigated the performance of the proposed method. As a result, in the case of an intermittency chaotic noise and a small amount of noise, the proposed method shows higher performance. To examine the reason why the performance of the proposed method depends on a type of dynamical noise, we calculate the Lyapunov exponents of the CNN. However, the Lyapunov exponents show almost the same tendency even though the type of dynamical noise is changed. Thus, it is an important future work to investigate the reason of good performance with other measures than the Lyapunov exponents.

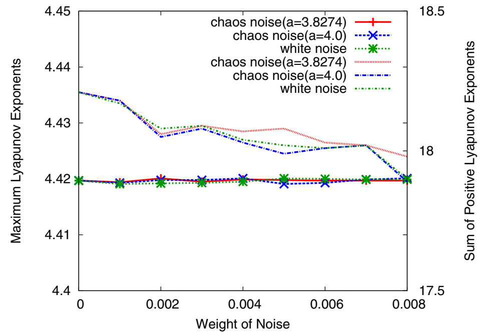


Figure 4: Maximum Lyapunov exponents (lines with symbols) and sum of positive Lyapunov exponents (lines without symbols) when the noise weight β is changed for Had20.

Acknowledgments

The research of T.I. is partially supported by Grant-in-Aid for Scientific Research (B) (No.20300085) from the JSPS. The research of T.M. is partially supported by Grant-in-Aid from the JSPS Fellows (No.20-6863).

References

[1] T. C. Koopmans and M. J. Beckmann, *Econometrica*, **25**, pp.53–76, 1957.

[2] J. J. Hopfield and D. W. Tank, *Biol. Cybern*, **52**, pp.141–152, 1985.

[3] K. Aihara, T. Takabe and M. Toyoda, *Phys. Lett. A*, **144**, pp.333–340, 1990.

[4] T. Ikeguchi, K. Sato, M. Hasegawa and K. Aihara, *Proc. of ISCAS 2002*, **3**, pp.469–472, 2002.

[5] Y. Horio, T. Ikeguchi and K. Aihara, *NN*, **18**, pp.505–513, 2005.

[6] T. Suzuki, S. Motohashi, T. Matsuura and T. Ikeguchi, *IEICE 2009 Gen. Conf. Proc.*, **A-2-27**, 2009.

[7] T. Suzuki, S. Motohashi, T. Matsuura and T. Ikeguchi, *IEICE Tech. Rep.*, **109**, pp. 51–56 2009.

[8] Y. Hayakawa, A. Marumoto and Y. Sawada, *Phys. Rev. E*, **51**, pp.R2693–R2696, 1995.

[9] Y. Uwate, Y. Nishio, T. Ueta, T. Kawabe and T. Ikeguchi, *IEICE Trans. Fund.*, **E87-A**, pp.937–943, 2004.

[10] H. F. von Bremen, F. E. Udwadia, W.Proskurowski, *Physica D*, **101**, pp.1–16, 1997.

[11] R. E. Berkard, S. E. Karisch and F. Rendl, "QAPLIB—A quadratic assignment Problem library," http://www.opt.math.tugaraz.ac.at/qaplib