

Performance of Quadratic Assignment Problem by Hopfield NN with Periodic Brake

Hironori Kumeno, Yoko Uwate and Yoshifumi Nishio
Dept. of Electrical and Electronic Eng., Tokushima University,
2-1 Minami-Josanjima, Tokushima, 770-8506 JAPAN
Email: {kumeno, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—Solving combinatorial optimization problems is one of the important applications of neural networks. Many researchers have proposed noise induced hopfield neural networks in which noises are induced state values of neurons. However, the noise inducing method to state values of neurons cause problems.

In this study, we propose hopfield neural networks with periodic brake. In the proposed system, external noises are not induced to state values of neurons. Thus, the proposed system can avoid the problem caused in the noise induced system. We investigate the solving ability of the proposed system for quadratic assignment problems and designing of parameters.

I. INTRODUCTION

The hopfield neural networks (HNN) is a form of recurrent artificial neural network invented by Hopfield and have been applied to solve combinatorial optimization problems [1]. When connection weights between neurons are related to given problems, the network gives a good solution. Because, the energy of the network converges to a minimum value with natural operation determined by the connection weights. However, the solutions are often trapped into local minimums and do not reach the global minimum demanded. In order to avoid this problem and solve the global minimum effectively, several methods inducing some kinds of noises are proposed by researchers [2]-[8]. Especially, methods that noises are induced into state values of neurons are well proposed and studied. In these methods, noises are induced to state values of neurons in the HNN, and then firing neurons are forcibly switched by the noises. These methods are effective to avoid local minimums. However these noise inducing methods sometimes causes a problem. When the noise induced HNNs are used to solve quadratic assignment problems (QAP), the neurons of HNNs are arranged on a plane surface in order to adapt to two-dimensional matrices of the solving problems. The HNN without noise is constructed to fire only one neuron on each line. However, in the noise induced HNNs, two or more neurons located on a line sometimes fire. Figure 1 shows a pattern of the firing of two neurons on a line. In the figure, black colored squares show firing neurons, and two neurons enclosed in the red colored circle fire on the same horizontal line. The firing of two or more neurons on a line is a problem of this noise inducing method and causes high dependence on solving problems.

By the way, in the real world, human cannot concentrate on one thing for a long period of time. Human can keep high-concentration for ten minutes at most in professional

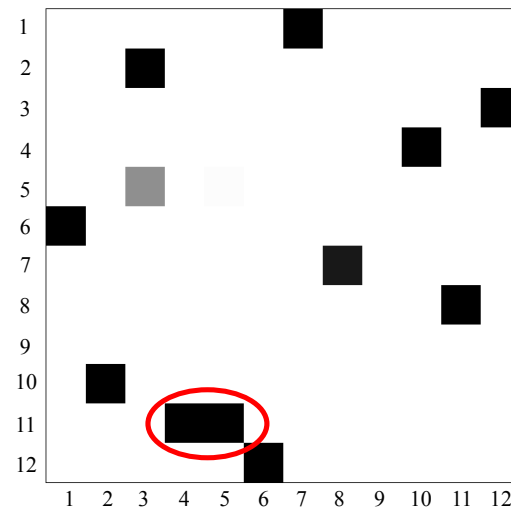


Fig. 1. An example of firing of two neurons on a line.

view. So that, a break is important to refresh. It is necessary to have a break for keep his mind clear and concentrate again. If he continues to do his task without a break, the efficiency of his task is down. Such a break is inferred to make neurons rest and refresh and then yields high-efficiency. Although the HNN is not real physical neural networks, we adapt this idea to the HNN for escape local minimums and propose a method to escape local minimums. We call the proposing system HNN with periodic brake (HNN-PB). In the proposed system, random values are periodically given to coupling weights between neurons in the network. The state values of neurons in HNN-PB converge according to the given random values of the coupling weights. Then, the system escapes from local minimums. The advantage of the proposed method is that firing of two or more neurons on a line is not caused because the system does not include noise terms in state values.

In this study, we investigate the solving ability of the HNN-PB for QAP. We confirm that the method is effective to solve QAP by computer simulations. Then, we investigate designing of optimal parameters for the HNN-PB.

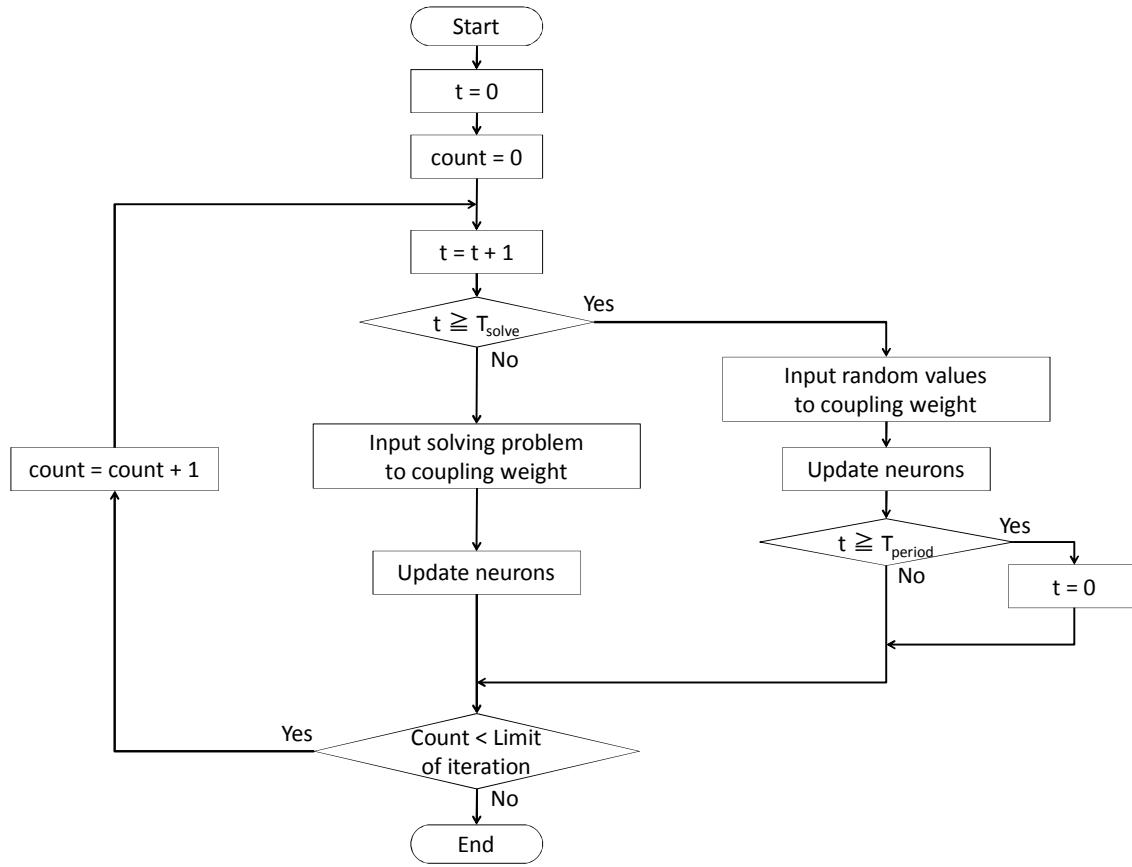


Fig. 2. Flow chart of HNN-PB.

II. SOLVING QAP WITH HNN

Various methods are proposed for solving QAP which is one of the NP-hard combinatorial optimization problems. We explain QAP with a factory arrangement problem. The problem is given by two matrices, distance matrix C denoting the distances between the factories and flow matrix D denoting the flow of the products between the factories, and is to find the permutation P which corresponds to the minimum value of the objective function $f(P)$ in Eq. (1).

$$f(P) = \sum_{i=1}^N \sum_{j=1}^N C_{ij} D_{p(i)p(j)}, \quad (1)$$

where C_{ij} and D_{ij} are the (i,j)-th elements of C and D , respectively, $p(i)$ is the i-th element of vector P , and N is the size of the problem. There are many real applications which are formulated by Eq. (1). Other examples are the placement of logical modules in an IC chip and the distribution of medical services in large hospital.

Because the QAP is very difficult, it is almost impossible to solve the optimum solution in large problems. Further, computation time is very long to obtain the exact optimum solution. Therefore, it is usual to develop heuristic methods which search nearly optimal solutions in reasonable time.

For solving N-element QAP by the HNN, $N \times N$ neurons are required and the following energy function is defined to fire (i,j)-th neuron at the optimal position:

$$E = \sum_{i,m=1}^N \sum_{j,n=1}^N \omega_{im;jn} x_{im} x_{jn} + \sum_{i,m=1}^N \theta_{im} x_{im}. \quad (2)$$

The neurons are coupled each other with coupling weight between (i,m)-th neuron and (j,n)-th neuron and the threshold of the (i,m)-th neuron are described by:

$$\omega_{im;jn} = -2 \left\{ A(1 - \delta_{mn})\delta_{ij} + B\delta_{mn}(1 - \delta_{ij}) + \frac{C_{ij}D_{mn}}{q} \right\} \\ \theta = A + B \quad (3)$$

where A and B are positive constant, and δ_{ij} is Kroneker's delta. The state of $N \times N$ neurons are asynchronously updated due to the following difference equation:

$$x_{im}(t+1) = f \left(\sum_{j,n=1}^N \omega_{im;jn} x_{im}(t) x_{jn}(t) - \theta_{im} \right), \quad (4)$$

where f is sigmoidal function defined as follows:

$$f(x) = \frac{1}{1 + \exp(-\frac{x}{\epsilon})}. \quad (5)$$

III. ALGORITHM OF HNN-PB

We explain hopfield neural network with periodic brake (HNN-PB) in detail. Figure 2 shows flow chart of the HNN-PB. The algorithm of the HNN-PB can be separated into two parts which are alternately repeated in iteration. One of the two parts is a solving part in which the distance matrix and the flow matrix of QAP are inputed to C_{ij} and D_{mn} in Eq. 3. Then, the states of the neurons in the HNN-PB are updated with the same updating method of the general HNN. In the solving part, the operation of the HNN-PB is completely the same as the general HNN. The other part is a brake part in which random values are inputed to C_{ij} and D_{mn} in Eq. 3 every iteration. The random value is a value between 0 and N_{max} . The updating operation of the neurons in the brake part is the same as that of the solving part. So, all the equations and the updating operation of neurons in the HNN-PB are the same as the general HNN. The two parts are alternately repeated. The iteration time of the solving part and the brake part are denoted by T_{solve} and T_{brake} , respectively. The length of one cycle of the two parts is termed as T_{period} :

$$T_{period} = T_{solve} + T_{brake}. \quad (6)$$

In the HNN-PB, C_{ij} and D_{mn} in Eq. 3 periodically become random values, although the updating equation of the states of neurons does not include a noise term. Therefore, firing of a neuron which is forcibly caused by a noise does not happen in the HNN-PB. Firing of two or more neurons on a line does not caused.

IV. SIMULATION RESULTS

In this section, the simulation results of HNN-PB for 12-element QAP are shown. The problem used here was chosen from the site QAPLIB named “Nug12” [9]. The global minimum of Nug 12 is known as 578. The parameters of HNN-PB are set as $A = 0.9$, $B = 0.9$, $q = 70$ and $\varepsilon = 0.35$. The value of ε used in this study is larger than other general HNNs which include noise terms in update equations of state values of neurons. Namely, the gradient of the sigmoid function used in this study is glaxis. The glaxis gradient of the sigmoid function makes neurons need a long iteration time to converge to a stable state. This prevent that state values of neurons suddenly change to states which do not relate to the solving problem when random values are inputed to the coupling weights in brake parts. To prevent sudden change of the state values of neurons in brake parts is important to solve good solutions. The other parameters associated with the periodic brake are set as $N_{max} = 5.0$, $T_{period} = 10$ and $T_{brake} = 3$. We carried out 1000 trails of 10000 iterations. In order to verify the solving ability of the HNN-PB, we compare the solving ability of the HNN-PB to the noise induced HNN that intermittency chaotic noises are induced to update equations of state values of neurons, proposed in [5].

Next, we tried another problem, whose name is “Tai12a”. The global minimum of Tai12a is 224416. The parameters of the HNN are fixed as $A = 0.9$, $B = 0.9$, $q = 9000$ and $\varepsilon = 0.20$. The other parameters associated with the periodic

TABLE I
SOLVING ABILITIES FOR NUG12.

Iteration	HNN with noise	HNN-PB
1000	632.96	602.234
2000	623.30	596.942
3000	619.82	594.188
4000	616.18	592.354
5000	613.56	591.114
6000	612.68	590.234
7000	611.74	589.624
8000	610.48	588.948
9000	610.30	588.434
10000	609.96	587.916

TABLE II
SOLVING ABILITIES FOR Tai12.

Iteration	HNN with noise	HNN-PB
4000	252624.44	245032.000
8000	251337.90	242803.980
12000	250291.38	240742.440
16000	249638.16	239133.920
20000	249520.72	237915.700
24000	249495.62	237111.160
28000	249458.04	236779.760
32000	249335.58	236673.520
36000	249228.40	236375.820
40000	249225.84	236044.340

brake are set as $N_{max} = 110$, $T_{period} = 10$ and $T_{brake} = 4$. We carried out 1000 trails of 40000 iterations.

Tables 1 and 2 show the mean values of the best solutions obtained during each iteration numbers for Nug12 and Tai12a, respectively. From these results, the HNN-PB gains better performance than the noise induced HNN.

The convergence speed of state values of neurons in the HNN-PB is slow because of the glaxis gradient of the sigmoid function and the system cannot solve the global minimum in brake parts. However, comparing with the noise induced HNN, the HNN-PB solve better solutions in the same limit of iteration. In the noise induced HNN, chaotic noises are generated every iterations and induced to the state values of all neurons,

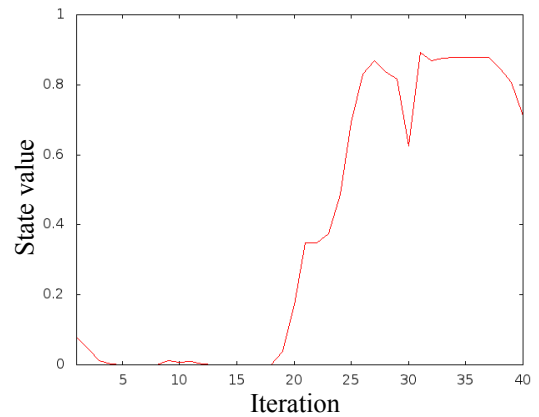


Fig. 3. Passage of the state value of a neuron which shift from a non-firing neuron to a firing neuron.

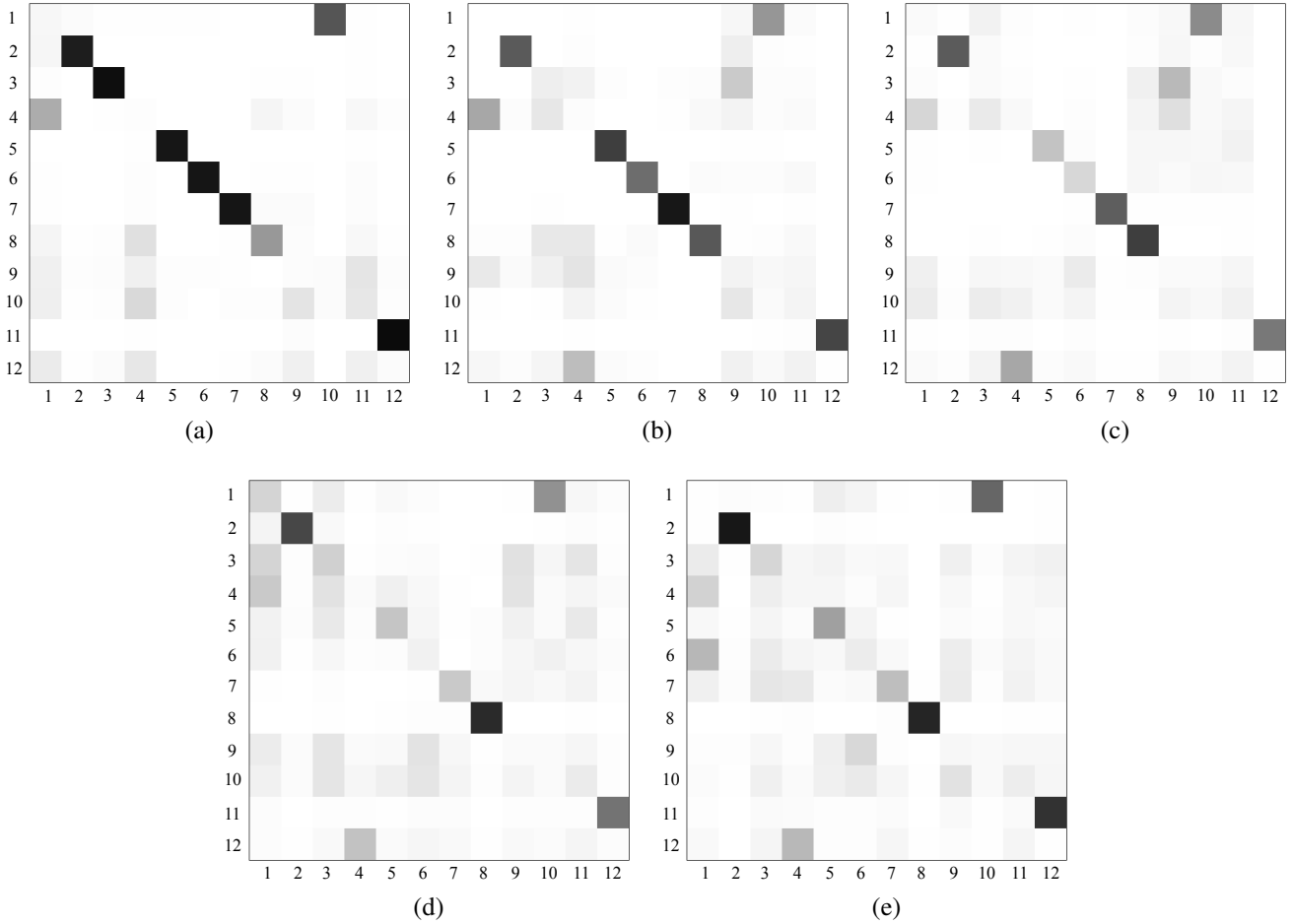


Fig. 4. State values of neurons around a brake part. (a) State values of neurons before a brake part. (b), (c) and (d) shows state values of neurons every iteration in the brake part. (e) State values of neurons after the brake part.

whereas random values are inputted to coupling weights only in brake parts in the HNN-PB. Thus, the computing speed of the HNN-PB is faster than the noise induced HNN.

V. PASSAGE OF STATE VALUES OF NEURONS

We explain the passage of state values of neurons in HNN-PB when $T_{period} = 10$ and $T_{brake} = 3$ which are set as an example. Figure 4 shows the passage of the state value of a neuron which shift 0 to 1 on iteration. In the figure, horizontal axis is iteration time. The iteration is started with a solving part. So, from 1 to 7 iteration, the state value of the neuron converge a local minimum in the solving part. After that, a brake part is started in which random values are inputted to the coupling weights. Then, the brake part is finished at 10 iteration. In the second brake part from 17 to 20 iteration, the state value of the neuron changes significantly. After that, in the next solving part, the state value of the neuron tend toward 1 which correspond to firing of the neuron. In the process, the HNN-PB escape from the local minimum.

Figure 4 shows the states of all neurons around a brake part. Figure 4(a) shows the states of the neurons before the

brake part. Almost neurons converge to a stable state. Figures 4(b), (c) and (d) show the states of the neurons every iteration in the brake part. By increasing iteration time, black colored neurons which converged in Fig. 4(a) fade, and the number of gray colored neurons increase. However, some neurons stay about the same. In this way, initial values of the next solving part become non-uniform state values which are a little similar to the state values before the brake part. Figure 4(e) shows the states of the neurons after the brake part. The neurons tend toward another local minimum or the global minimum.

VI. DESIGNING OF OPTIMAL PARAMETERS

We investigate designing of optimal parameters. For solving Nug12, the optimal parameters are $N_{max} = 5.0$, $T_{period} = 10$ and $T_{brake} = 3$. T_{period} is designed associated with ε . If ε is big, neurons converge a local minimum fast, whereas if ε is small, the convergence speed is slow. When $\varepsilon = 0.35$, neurons converge a local minimum for about 10 iterations. Thus, T_{period} is set at 10. If T_{period} increase, the efficiency to search solutions is down because states of neurons do not change after convergence, iterations after the convergence are

waste. If T_{period} decrease, the effect of the random values induced in brake parts become strong. As the result, it becomes difficult that the system close in the global minimum. Figure 5 shows the mean values of the best solutions when Nug12 is solved, the parameter T_{period} is fixed at 10 and T_{brake} and N_{max} vary. Figure 6 shows the mean values of the best solutions when Nug12 is solved, the parameter T_{period} is fixed at 5 and T_{brake} and N_{max} vary. Comparing the two figures, the solving abilities in Fig. 6 are worse than that of Fig. 5. Small value of T_{period} make the efficiency to solve the global minimum worse.

T_{brake} bears a trade-off relationship to N_{max} . When N_{max} is large, small values of T_{brake} gain good solving abilities. Whereas, when N_{max} is small, relatively small values of T_{brake} gain good solving ability. This relationship can be seen in the figures. As we explained in the last section, the state values of neurons which trapped in a local minimum are changed and the system can escape the local minimum by inputting random values to coupling weights in brake parts. Large random values inputted in the brake parts make sudden change of the state values of neurons. When the small random values are inputted in the brake parts, it need long iteration time to change the state values of the neurons. Therefore, the relationship between T_{brake} and N_{max} explained in the above is obtained.

For solving Tai12a, the optimal parameters are $N_{max} = 110$, $T_{period} = 10$ and $T_{brake} = 4$. Figure 7 shows the mean values of the best solutions when Tai12a is solved, the parameter T_{period} is fixed at 10 and T_{brake} and N_{max} vary. A rough value of N_{max} is decided in association with values of the flow matrices and the distance matrices of the solving problems. The maximum values of the flow matrix and the distance matrix in Nug12 are 10 and 5, respectively. The maximum values of the flow matrix and the distance matrix in Tai12 are 99. In brake parts, random values are inputted to C_{ij} and D_{mn} in Eq. 3. The maximum value of the random values inputted to C_{ij} and The maximum value of the random values inputted to D_{mn} are the same value which is N_{max} . N_{max} is designed as become relatively similar to the maximum values of the solving problems. Because, the parameter q of HNN-PB is decided in association with values of the matrices of the solving problem, and q is constant in solving parts and in break part. Therefore, N_{max} is necessary to suit to the solving problem.

From the figures, we can see combinations of small N_{max} and some few iteration of T_{brake} make better result than combinations of large N_{max} and one iteration of T_{brake} . If N_{max} is large, the states of the neurons are suddenly attracted to the states which are not related to the solving problem in brake parts. Then, the system cannot reach to the global minimum in solving parts, sufficiently. On the other hand, if N_{max} is small, certain informations of the local minimums which obtained for the solving problem before brake parts remain after the brake parts. Then, the system can search around the global minimum, effectively.

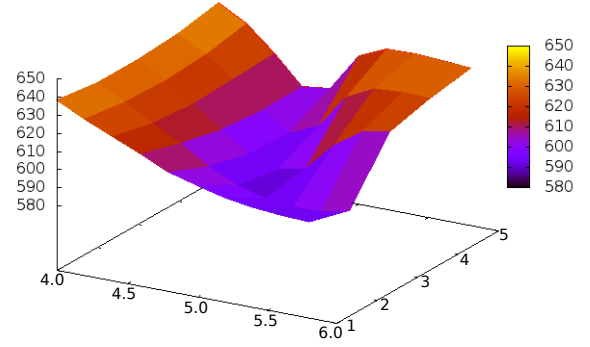


Fig. 5. Mean values of the best solutions for solving Nug12 $T_{period} = 10$. x-axis: N_{max} . y-axis: T_{brake} . z-axis: Mean values of the best solutions.

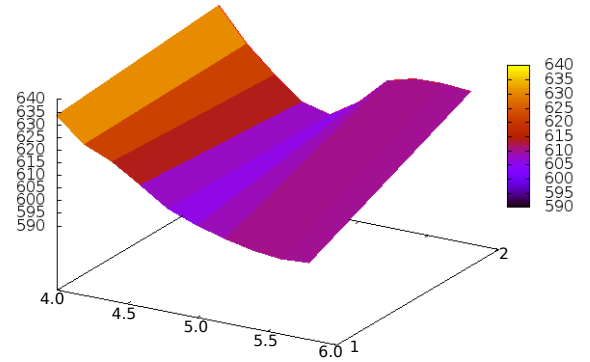


Fig. 6. Mean values of the best solutions for solving Nug12 $T_{period} = 5$. x-axis: N_{max} . y-axis: T_{brake} . z-axis: Mean values of the best solutions.

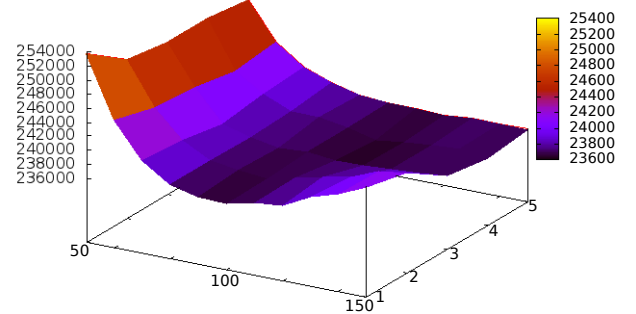


Fig. 7. Mean values of the best solutions for solving Tai12 $T_{period} = 10$. x-axis: N_{max} . y-axis: T_{brake} . z-axis: Mean values of the best solutions.

VII. CONCLUSIONS

In this study, we proposed hopfield neural network with periodic brake in which random values are periodically inputed to coupling weights between neurons instead of inputing flow and distance matrices of solving problems. One main feature of the proposed system is that external noises are not induced to state values of neurons. We investigated the solving ability of the proposed system for quadratic assignment problems. By computer simulation, we have confirmed that the proposed system can find better solutions comparing with the noise induced HNN in which noises are induced to state values of neurons. Moreover, the computing speed of the HNN-PB is faster than the noise induced HNN. Then, we investigated designing of optimal parameters.

ACKNOWLEDGMENT

This work was partly supported by JSPS Grant-in-Aid for Young Scientists 23700269.

REFERENCES

- [1] J. J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons," *Proc. Natl. Acad. Sci. USA*, vol. 81, pp. 3088-3092, 1984.
- [2] Y. Hayakawa and Y. Sawada, "Effects of the Chaotic Noise on the Performance of a Neural Network Model for Optimization Problems," *Physical Review E*, vol.51, no.4, pp.2693-2696, Apr. 1995.
- [3] M. Hasegawa, T. Ikeguchi, T. Matozaki and K. Aihara, "An analysis on additive effects of nonlinear dynamics for combinatorial optimization," *IEICE Trans. Fundamentals*, vol.E80-A, no.1, pp.206213, Jan. 1997.
- [4] K. Sato, T. Ikeguchi, M. Hasegawa and K. Aihara, "An Optimization Method for Quadratic Assignment Problems by Chaotic Dynamics and its Characterization by Lyapunov Dimensions," *IEICE Tech. Rep.*, vol. NLP64-13, pp. 13-20, 2001 (in Japanese).
- [5] Yoko UWATE, Yoshifumi NISHIO and Akio USHIDA, "Markov Chain Modeling of Intermittency Chaos and its Application to Hopfield NN for QAP," *Proc. of ECCTD'03*, vol. 2, pp. 57-60, Sep. 2003.
- [6] Y. Uwate and Y. Nishio, "Search of Many Good Solutions of QAP by Connected Hopfield NNs with Chaos Noise," *Proc. of NCSP 04*, pp. 145-148, Mar. 2004.
- [7] Y. Uwate, Y. Nishio, T. Ueta, T. Kawabe and T. Ikeguchi, "Performance of Chaos and Burst Noises Injected to the Hopfield NN for Quadratic Assignment Problems," *IEICE Trans. on Fundamentals*, vol.E87-A, no.4, pp.937943, Apr. 2004.
- [8] Y. Tada, Y. Uwate and Y. Nishio, "Solving Ability of Hopfield Neural Network with Scale-Rule Noise for QAP," *Proc. of ISCAS08*, pp. 105-108, May 2008.
- [9] R. E. Burkard, S. E. Karisch and F. Rendl, "QAPLIB-A Quadratic Assignment Problem Library," <http://www.opt.math.tu-graz.ac.at/qaplib/>