IEICE Proceeding Series

How to Decide Solutions of Quadratic Assignment Problem from Chaotic Neural Network

Takafumi Matsuura, Tohru Ikeguchi

Vol. 2 pp. 358-361
Publication Date: 2014/03/18
Online ISSN: 2188-5079

How to Decide Solutions of Quadratic Assignment Problem from Chaotic Neural Network

Takafumi Matsuura† and Tohru Ikeguchi‡

†Department of Management Science, Faculty of Engineering, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
‡Graduate School of Science and Engineering, Saitama University
225 Shimo-Ohkubo, Sakura-ku, Saitama-city 338-8570, Japan
Email: matsuura@ms.kagu.tus.ac.jp, tohru@mail.saitama-u.ac.jp

Abstract—The quadratic assignment problem (QAP) is one of famous combinatorial optimization problems which belong to a class of $\mathcal{NP}$-hard. To solve the QAP, a chaotic search method which uses chaotic neural network has been proposed. In the method, chaotic dynamics of the chaotic neural network effectively controls to avoid the local minima and to search optimal or near-optimal solutions. However, it is not so easy to generate feasible solutions from the chaotic neural network, because an output of a chaotic neuron takes an analog value. Thus, for obtaining good solutions from the chaotic neural network, it is important to develop a method that always generates a feasible solution of the QAP. To generate a feasible solution of the QAP, we have already proposed a firing decision method. In this paper, to improve performances of the method, we investigate what factors are essential to the firing decision method.

1. Introduction

The quadratic assignment problem (QAP) is one of famous combinatorial optimal problems [1]. The QAP is described as follows: A set of N facilities, a set of N locations, distance matrix $D = (d_{ij})$ between locations, and flow matrix $F = (f_{mn})$ between facilities are given. Here, d_{ij} is the distance between locations i and j, and f_{mn} is the flow between the facilities m and n. In the QAP, every facility is assigned to exactly one location and no location is assigned more than one facility. The goal of the QAP is to find an assignment of N facilities to N locations such that the sum of the product between flows and distances is minimized. The QAP is formulated as follows:

$$\text{minimize} \quad \sum_{i=1}^{N}\sum_{m=1}^{N}\sum_{j=1}^{N}\sum_{n=1}^{N} d_{ij} f_{mn} x_{im} x_{jn} \tag{1}$$

$$\text{subject to} \quad \sum_{m=1}^{N} x_{im} = 1 \quad (i = 1, 2, ..., N) \tag{2}$$

$$\sum_{i=1}^{N} x_{im} = 1 \quad (m = 1, 2, ..., N) \tag{3}$$

$$x_{im} \in \{0, 1\} \quad (i, m = 1, 2, ..., N) \tag{4}$$

where x_{im} is decision variable. If facility i is assigned to location m, $x_{im} = 1$, otherwise, $x_{im} = 0$. The QAP belongs to a class of $\mathcal{NP}$-hard. Thus, it is required to develop effective approximate algorithms for finding near-optimal solutions in a reasonable time frame.

As an approximate algorithm, the Hopfield-Tank neural network (HNN) has already been proposed [2]. In the method, a firing pattern of HNN represents a solution of the QAP. If synaptic weights of HNN are set to appropriate values, good solutions are obtained by the energy minimization principle. However, this method cannot always show good performance because the states of the HNN get stuck at local minima.

To avoid local minima, a method which uses a chaotic neural network (CNN) [3] has already been proposed [4–7]. In the method, chaotic dynamics of the CNN effectively controls to avoid the local minima and to search optimal or near-optimal solutions. For N-size QAP, N^2 neurons arranged on an $N \times N$ grid are required, and then a state of the CNN represents a solution of the QAP. If the *im*th chaotic neuron fires, facility i is assigned to location m. However, it is not so easy to generate feasible solutions from the CNN, because an internal state of a chaotic neuron takes an analog value. Thus, one of the most important issues in this method is how to decide the firing pattern of the CNN, that is, how to decide the feasible solutions from the CNN.

To generate the feasible solutions of the QAP, we have already proposed a firing decision method by using internal states of all the neurons. The method repeatedly sets fire to the neuron whose internal state is the maximum value among all neurons to satisfy the feasibility conditions of the QAP. In this paper, to improve the performances of the firing decision method, we investigate what factors are essential to the firing decision method.

2. Chaotic Neural Network for Solving the QAP

As an approximate algorithm for solving the QAP, a method which uses mutual connection chaotic neural network (CNN) [3] has already been proposed [4–7]. The CNN is constructed by chaotic neurons [3]. This neural network model can qualitatively reproduce a chaotic dynamics observed in real neural membrane.

For solving the N-size QAP, N^2 chaotic neurons are required, and these are arranged on an $N \times N$ grid. An internal state of the *im*th chaotic neuron for the QAP is defined as follows:

$$\begin{aligned} y_{im}(t+1) &= k_r y_{im}(t) + \sum_{j=1}^{n} \sum_{l=1}^{n} w_{im;jn} f(y_{jn}(t)) \\ &\quad -\alpha f(y_{im}(t)) + \theta_{im}(1-k_r), \end{aligned} \tag{5}$$

where k_r is a decay parameter of a refractory effect and α is a strength parameter of a refractory effect. The chaotic neurons are coupled each other with a synaptic connection weight. $w_{im;jn}$ is the synaptic weight between the *im*th neuron and the *jn*th neuron. θ_{im} is a threshold of the *im*th chaotic neuron, and f is an output function of the chaotic neuron. As an output function, a sigmoidal function is used: $f(y) = 1/(1+\exp(-y/\epsilon))$, where ϵ is a gradient parameter of the sigmoidal function.

For solving the QAP by the CNN, the objective function of the QAP is newly defined as follows:

$$\begin{aligned} F(X) &= A \sum_{i=1}^{N} \left(\sum_{m=1}^{N} X_{im} - 1 \right)^2 + B \sum_{m=1}^{N} \left(\sum_{i=1}^{N} X_{im} - 1 \right)^2 \\ &\quad + \sum_{i=1}^{N} \sum_{m=1}^{N} \sum_{j=1}^{N} \sum_{n=1}^{N} d_{ij} f_{mn} X_{im} X_{jn}, \end{aligned} \tag{6}$$

where A and B are positive constraints. $X = (X_{im})$ is $N \times N$ matrix and represents solutions of the QAP generated from the CNN, namely firing patterns of the CNN. Therefore, the matrix X satisfies following conditions: $\sum_{m=1}^{N} X_{im} = 1$, $\sum_{i=1}^{N} X_{im} = 1$, and $X_{im} \in \{0,1\}$ $(i, m = 1, ..., m)$. In Eq. (6), the first and the second terms correspond to the constraints of the QAP (Eqs. (2) and (3)) and the third term is the objective function of the QAP. From Eqs. (5) and (6), the synaptic weight between the *im*th neuron and the *jn*th neuron, and the threshold of the *im*th neuron are defined as follows:

$$w_{im;jn} = -2 \left\{ A(1-\delta_{mn})\delta_{ij} + B\delta_{mn}(1-\delta_{ij}) + \frac{d_{ij} f_{mn}}{q} \right\} \tag{7}$$

$$\theta_{im} = A + B \tag{8}$$

where δ_{ij} is Kronecker's delta, and q is a normalization parameter. In a single iteration, all neurons are asynchronously updated.

3. Firing Decision Method

For the N-size QAP, N^2 chaotic neurons on an $N \times N$ square grid are prepared, and then, a firing patter of the CNN represents a solution of the QAP. If the *im*th chaotic neuron fires, the *i*th facility is assigned at the *m*th location. However, we cannot always obtain feasible solutions from outputs of the neurons because an output of the chaotic neuron takes an analog value. Thus, a firing decision method is necessary to obtain the feasible solutions from the CNN. In the CNN, if the internal states of the *im*th chaotic neuron takes large value, it indicates that, for the facility i, an assignment to location k is good. Therefore, it is expected that if the neurons whose internal state is large value fire, good feasible solutions are constructed.

3.1. Greedy Method [4, 5]

To construct feasible solutions at every iteration, we have already proposed a firing decision method using the internal states of the chaotic neurons [4, 5]. This method is greedy algorithm to maximize a sum of the internal state of firing neurons. The procedure of the method is described as follows:

1. The *im*th chaotic neuron, that gives the maximum value of the internal state among all of the internal states is selected. Then, we set $X_{im} = 1$, namely the *im*th neuron fires.
2. To satisfy the constraints of the QAP (Eqs. (2) and (3)), for other neurons in the *i*th row and the *m*th column, we set $X_{ik} = 0$ $(k \neq m)$ and $_{ml} = 0$ $(l \neq i)$.
3. The neurons which have already been selected in Steps 1 and 2 are excluded from the candidate at step 1. Steps 1 and 2 are repeated N times to make a feasible solution.

3.2. Max-Sum Model

In the CNN, the neurons whose internal state is large value indicate good assignments of facilities. Thus, it seems that if a sum of the internal state (y_{im}) of firing neurons is maximum, the best solution is obtained from the CNN. A max-sum model maximizes the sum of the internal state of firing neurons. The max-sum model is described as follows:

$$\text{maximize} \quad \sum_{i=1}^{N} \sum_{m=1}^{N} X_{im} y_{im} \tag{9}$$

$$\text{subject to} \quad \sum_{m=1}^{N} X_{im} = 1 \ (i = 1, ..., N) \tag{10}$$

$$\sum_{i=1}^{N} X_{im} = 1 \ (m = 1, ..., N) \tag{11}$$

$$X_{im} \in \{0, 1\} \quad (i, j = 1, ..., N) \tag{12}$$

where y_{im} is the internal state of the *im*th chaotic neuron and X_{im} is a decision variable or a firing pattern of the CNN. If the decision variable X_{im} is 1, the facility i is assigned to the location m, namely the *im*th chaotic neuron fires.

3.3. Max-Min Model

The max-sum model maximizes the sum of internal state (y_{im}) of firing neurons. Therefore, the neuron whose internal state is low may fire. In order not to fire such neuron, a max-min model is used. The max-min model maximizes the minimum value of the internal state of firing neurons. The model is described as follows:

$$\text{maximize} \quad \min_{i=1,...N} \sum_{m=1}^{N} X_{im} y_{im} \tag{13}$$

$$\text{subject to} \quad \sum_{m=1}^{N} X_{im} = 1 \ (i = 1, ..., N) \tag{14}$$

$$\sum_{i=1}^{N} X_{im} = 1 \ (m = 1, ..., N) \tag{15}$$

$$X_{im} \in \{0, 1\} \quad (i, m = 1, ..., N) \tag{16}$$

3.4. Mixed Model

In a mixed model, a firing pattern is decided by considering the minimum value of the internal state of firing neurons and the sum of the internal state of firing neurons. In the mixed model, first, a minimum value L of the internal state of firing neurons is calculated by the max-min model. Next, a firing pattern of the CNN is decided by the max-sum model with a new constraint. The new constraint is that the internal state of firing neurons is not less than the minimum value L. The mixed model is described as follows:

$$\text{maximize} \quad \sum_{i=1}^{N} \sum_{m=1}^{N} X_{im} y_{im} \tag{17}$$

$$\text{subject to} \quad \sum_{m=1}^{N} X_{im} = 1 \quad (i = 1, ..., N) \tag{18}$$

$$\sum_{i=1}^{N} X_{im} = 1 \quad (m = 1, ..., N) \tag{19}$$

$$\sum_{m=1}^{N} X_{im} \geq L \quad (i = 1, ..., N) \tag{20}$$

$$X_{im} \in \{0, 1\} \quad (i, j = 1, ..., N), \tag{21}$$

where L is the minimum value of the internal state of the firing neurons. Thus, by Eq.(20), the neurons whose internal state is less than L cannot fire.

3.5. Simulations and Results

To investigate the important properties of the firing decision methods, we prepare three benchmark problems from QAPLIB: Had20, Hug20, and Tai20a [8]. For all instances, the distance and the flow matrices are symmetric. d_{ij} and f_{mn} are uniformly generated for all instances.

The values of parameter α in the CNN method (Eq.(5)) are set to between from 0.95 to 1.25 by step size 0.025. The values of parameter k_r are set to between from 0.700 to 0.975 by step size 0.025. The value of parameter ϵ and θ_{im} is set to 0.02 and 1.0, respectively. The parameters A, B, and q are set to various values depending on the instances (Table 1). The CNN method is applied for 2,000 iterations, namely, 2,000 solutions are obtained in one trial. We compared the average gap between obtained solutions with 30 different initial conditions and the optimum solution. The initial conditions of the permutation p

Table 1: Values of parameters A, B, and q

Instance	A	B	q
Had20	34	34	1,100
Nug20	32	32	540
Tai20a	34	34	90,000

are randomly decided. In the simulations, for the max-sum model, the max-min model, and the mixed-model, a firing pattern X is found by the general-purpose MIP solver. In this simulation, we used the Gurobi Optimizer 5.5.0 [9].

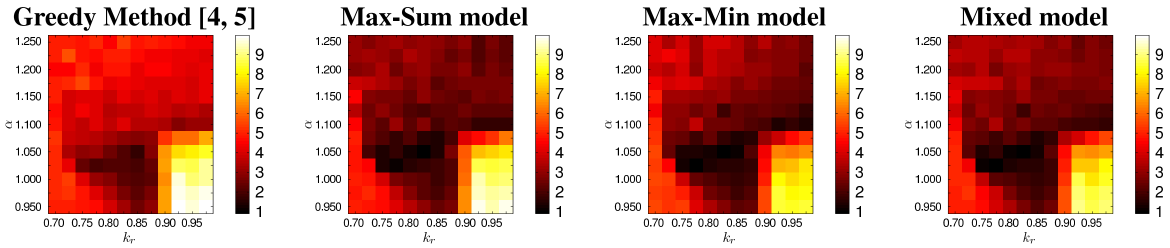
Figure 1 shows the performances of each method. From Fig. 1, we observe that, for all methods, good solutions are obtained from same values of parameters α and k_r. However, the best result obtained by each method is large difference.

Table 2 summarizes the best result of each method. In Table 2, the numbers with bold characters indicate the best and italic characters indicate the second best. From Table 2, for Nug20 and Had20, the max-min model and mixed model show the best performance. Therefore, for obtaining good solutions from the CNN, it is important to maximize the minimum value of the internal state of firing neurons. However, for Tai20a, the performances of the max-min model is the worst. On the other hand, the models that optimize the sum of the internal states of firing neurons good solutions. These results indicate that firing patterns of the CNN (solutions of the QAP) should be decided not only to maximize the sum of the internal state of firing neurons but also to maximize the minimum value of the internal state of firing neurons.

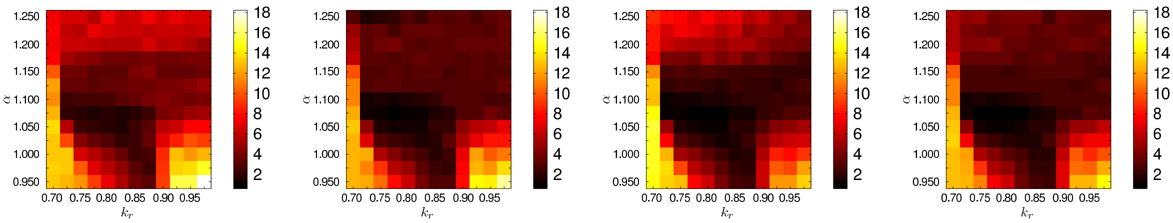
4. Conclusions

In this paper, to improve performances of a firing decision method [4, 5], we investigated what factors are essential to the firing decision method by using the three models: max-sum model, max-min model, and mixed-model. The max-sum model makes firing patterns of the CNN (solutions of the CNN) so that a sum of the internal state firing neurons is maximized. On the other hand, the objective of the max-min model is to maximize the minimum value of the internal state of firing neurons. The firing patters of the mixed model are generated by considering the minimum value of the internal state and the sum of internal state. From the results, we clarified that firing patterns of the CNN (solutions of the QAP) should be decided not only to maximize the sum of the internal state of firing neurons but also to maximize the minimum value of the internal state of firing neurons. In future works, it is important to develop a firing decision method not only for maximizing the minimum value of the internal state of firing neurons and but also for maximizing the sum of the internal state of firing neurons.

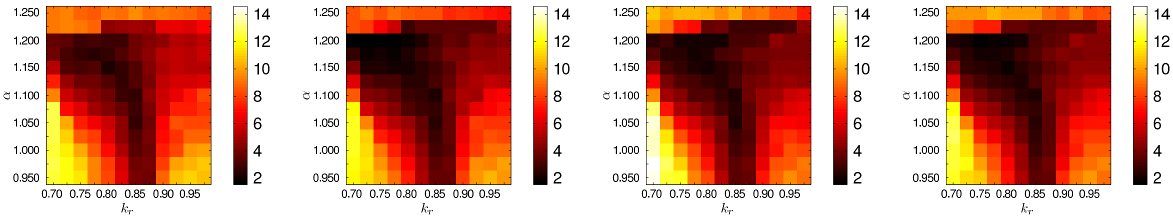
The research of T.M. is partially supported by Grant-in-Aid for Young Scientists (B) (No.25870770) from the JSPS.



(a) Had20



(b) Nug20



(c) Tai20a

Figure 1: Results of each firing decision method. Percentages of gaps between average obtained solutions and the optimal solutions are shown by color-shaded bars.

Table 2: The best results of the CNN with each firing decision method. The values are percentages of gaps between obtained solutions and the optimal solutions for each problem. Bold characters indicate the best and italic characters indicate the second best.

Instance	Greedy method[4, 5]	Max-Sum model	Max-Min model	Mixed model
Had20	1.6835 (1.050/0.850)	1.0165 (1.050/0.825)	**0.9141** (1.050/0.825)	*0.9883* (1.050/0.825)
Nug20	1.5720 (1.075/0.825)	0.7004 (1.075/0.775)	*0.6615* (1.075/0.775)	**0.6070** (1.075/0.775)
Tai20a	2.3186 (1.150/0.800)	**1.5407** (1.200/0.700)	1.9742 (1.200/0.750)	*1.6113* (1.200/0.750)

References

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

[2] J. J. Hopfield and D. W. Tank (1985), *Biological Cybermetics*, **52**, 141–152.

[3] K. Aihara, et al. (1990), *Physics Letter. A*, **144**, 333–340.

[4] T. Ikeguchi, et al. (2002), *Proc. of ISCAS 2002*, **3**, 469–472.

[5] K. Sato, et al. (2001), *Technical Report of IEICE*, **101**(465), 13–20 (in Japanese).

[6] Y. Horio et al. (2005), *Neural Networks* **18**, 505–513.

[7] Y. Horio and K. Aihara (2008), *Physica D*, **237**, 1215–1225.

[8] QAPLIB: http://www.opt.math.tu-graz.ac.at/qaplib/

[9] Gurobi Optimization: http://www.gurobi.com