2016 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2016, Yugawara, Japan, November 27th-30th, 2016

A Chaotic Search with the Effect of Wide Range Search for Solving QAP

Hikaru Ohnishi†, Yutaka Shimada†,‡, Kantaro Fujiwara†,‡and Tohru Ikeguchi†,‡

†Department of Management Science, Graduate School of Engineering
‡Department of Information and Computer Technology, Faculty of Engineering, Tokyo University of Science
6–3–1 Niijuku, Katsushika–ku, Tokyo 125–8585, Japan

Abstract—One of the most important issues in engineering and science is to develop algorithms for finding good approximate solutions of NP-hard combinatorial optimization problems. In this paper, we proposed a new algorithm for solving Quadratic Assignment Problem (QAP) by chaotic neural dynamics. The proposed algorithm introduced a modified assignment of the neuron, which means that while the conventional method assigns neurons to the pair of facility and location, the proposed method assigns neurons to the location. In addition, we changed the effect of an external input: even if an exchange is bad, we applied a strong input to the chaotic neural dynamics. This effect enables a wide range search. As a result, our algorithm can find good solutions even though the number of neurons are reduced.

1. Introduction

In our daily life, many optimization problems exist, for example, scheduling, vehicle routing, facility location problem and so on. It is important to obtain possibly optimal solutions of these problems, because the cost can be reduced. However, it is very hard to obtain optimal solutions of such problems, because these problems are often classified into nondeterministic polynomial time solvable (NP)-hard problems. Therefore, we need to develop approximate algorithms to obtain near optimal solutions of these problems in a reasonable time frame.

On the other hand, several approximate algorithms are proposed for solving these NP-hard problems. One of the well known methods to solve these problems is a heuristic method, for example the 2-exchange method of Quadratic Assignment Problem (QAP). However, the heuristic methods such as the 2-exchange method are generally trapped into local minima. Due to this reason, many methods to escape from the local minima have also been proposed: for example, a tabu search[2, 3], a genetic algorithm[4], chaotic neural dynamics[5, 6] and so on. In this paper, we improve the method of using chaotic neural dynamics that we have already proposed in Ref.[7]. The method that we have proposed controls the 2-exchange method by using the chaotic neural dynamics. Compared with the method in Ref.[6], our proposed method reduced the number of neurons. In addition, to improve the performance of the method, we introduced a new strategy. Namely, we changed the effect of an external input: even if an exchange is bad, we applied a strong input to chaotic neural dynamics. This is the different point from the chaotic search which we have proposed in Ref.[7]. Due to this effect, bad exchanges are frequently executed, which means that the state can escape from undesirable local minima. We show that our proposed method enables a wide range search so that it succeeded to improve the performance.

2. QAP

The QAP is one of the most difficult NP-hard combinatorial optimization problems. The QAP is formulated as follows: when two $n \times n$ matrices, a distance matrix D and a flow matrix R, are given, we are asked to find an assignment $\boldsymbol{p} = \{p(1), p(2), \cdots, p(n)\}$ that minimizes an objective function. The objective function of QAP is then defined by Eq.(1):

$$F(\boldsymbol{p}) = \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} R_{p(i)p(j)}, \tag{1}$$

where $p(i)$ is the element i of the permutation $\boldsymbol{p}$. If $p(i) = j$, the element i is assigned to the location j. In the following, we introduced a 2-exchange method which is a basic algorithm for solving QAPs.

Step1： A random solution (assignment) $\boldsymbol{q}$ is made.

Step2： The objective function $F(\boldsymbol{q})$ is calculated.

Step3： From all the elements, two elements s_1 and s_2 are chosen. Then, locations assigned to s_1 and s_2 are changed. Let us describe a provided solution as $\boldsymbol{q}'$.

Step4： The objective function $F(\boldsymbol{q}')$ is calculated.

Step5： If a solution is improved, or $F(\boldsymbol{q}) > F(\boldsymbol{q}')$, then let $\boldsymbol{q} = \boldsymbol{q}'$. Return to **Step3**. When a solution was not improved, even if any two elements s_1 and s_2 were chosen, we stop a solution search.

Generally, the 2-exchange method is trapped into undesirable local minima. In this paper, we used a chaotic neural dynamics to escape from the local minima.

3. Proposed method

To control the 2-exchange method by the chaotic neural dynamics[8], we used Eqs.(2) ~ (5).

$$\begin{cases} \xi_j(t+1) = \beta\Delta_{ij}(t), & (2) \\ \eta_j(t+1) = w\sum_{l=1,(l\neq j)}^{n} x_l(t) + w, & (3) \\ \zeta_j(t+1) = k\zeta_j(t) - \alpha x_j(t) + (1-k)\theta, & (4) \end{cases}$$

$$x_j(t+1) = f\left\{\xi_j(t+1) + \eta_j(t+1) + \zeta_j(t+1)\right\}, \quad (5)$$

where $\xi_j(t)$ is an external input to the chaotic neuron j, $\eta_j(t)$ is a feedback input to the chaotic neuron j from other neurons in the network, $\zeta_j(t)$ is a refractoriness term of the chaotic neuron j, and $\Delta_{ij}(t)$ is a gain of the objective function when we change $p(j)$ to $p(i)$ by the 2-exchange method, k is a decay constant, w is a connection weight of chaotic neurons, α is a scaling parameter of refractoriness effect, θ is a threshold of the chaotic neuron, $x_j(t)$ is the output of the chaotic neuron j at time t and f is a sigmoidal function defined by $f(y) = 1/(1 + \mathrm{e}^{-y/\epsilon})$.

In the method in Ref.[6], when the problem size is n, the $n \times n$ chaotic neurons are prepared to represent an assignment of i and j. Namely, if the chaotic neuron (i, j) fires, the element i is assigned to the location j. Although this method[6] shows good performance, this method uses many neurons. On the other hand, in this study, we use n chaotic neurons for solving the problem of size n. For this reason, we can reduce the number of neurons. If the chaotic neuron i fires, we perform the 2-exchange method for the element i. We explain our algorithm as follows.

Step1：Let $i = 1$.

Step2：Internal state values of all chaotic neurons except the chaotic neuron i are updated asynchronously by Eqs.(2) ~ (4).

Step3：The output of all the chaotic neurons except for the chaotic neuron i are calculated by Eq.(5).

Step4：If $\Delta_{ij}(t) < 0$, the elements $p(i)$ and $p(j)$ are exchanged by the 2-exchange method and go to **Step6**. Otherwise go to **Step5**.

Step5：If $\max_j\{x_j(t+1)\} > 1/2$, the chaotic neuron j fires and the element $p(i)$ and $p(j)$ are really exchanged by the 2-exchange method.

Step6：If $i = n$, this iteration is finished. Otherwise let $i = i + 1$ and return to **Step2**.

There are two different points from Ref.[7]. First we changed the input term. If we have a bad exchange, we applied a strong input to the dynamics. Namely, in Ref.[7], $\Delta_{ij}(t)$ is defined as $\Delta_{ij}(t) = D_0(t) - D_{ij}(t)$, where $D_0(t)$ is the present value of the objective function at time t, and $D_{ij}(t)$ is the value of the objective function that is made by the exchange of $p(i)$ and $p(j)$. On the other hand, in this paper, we defined $\Delta_{ij}(t)$ as $\Delta_{ij}(t) = D_{ij}(t) - D_0(t)$. Second, we added **Step4** to introduce the steepest descent dynamics. Without **Step4**, only bad exchange is adopted. Then, we cannot get good solutions such as local minima. Therefore, we added **Step4**.

Table 1: Results of gaps[%] for (i) the chaotic search(CS) with n^2 neurons in Ref.[6], (ii) the chaotic search (CS) with n neurons in Ref.[7] and (iii) the proposed method. The best parameter β for each problem is shown in parenthesis.

Problem	(i) CS with n^2*neurons*[6]	(ii) CS with n neurons[7]	(iii) Proposed method	
	gap	gap	gap	variance
Bur26a	*0.159*	0.293	**0.122 (0.12)**	0.000983
Bur26b	**0.0814**	*0.111*	0.145 (0.40)	0.00469
Bur26c	*0.0496*	0.130	**0.0106 (0.13)**	0.000118
Bur26d	*0.0234*	0.080	**0.00662 (0.13)**	0.000028
Ste36a	5.65	**3.86**	*4.02 (0.06)*	1.04
Ste36b	12.7	*8.29*	**4.14 (0.08)**	6.80
Ste36c	4.40	*3.68*	**2.21 (0.06)**	2.02
Tai20b	*1.80*	3.14	**0.88 (0.04)**	0.21
Tai30b	2.33	*1.91*	**0.79 (0.06)**	0.16
Tai40b	*3.70*	4.58	**0.99 (0.02)**	0.99
Tai50b	*2.21*	3.96	**0.74 (0.02)**	0.089
Tai60b	2.52	*2.48*	**0.58 (0.02)**	0.47
Tai80b	2.88	*2.08*	**1.03 (0.02)**	0.18

4. Result

We evaluated the performance of the proposed algorithm using benchmark problems from QAPLIB[1]. To evaluate the performance, we used the gap which is defined by the following Eq.(6).

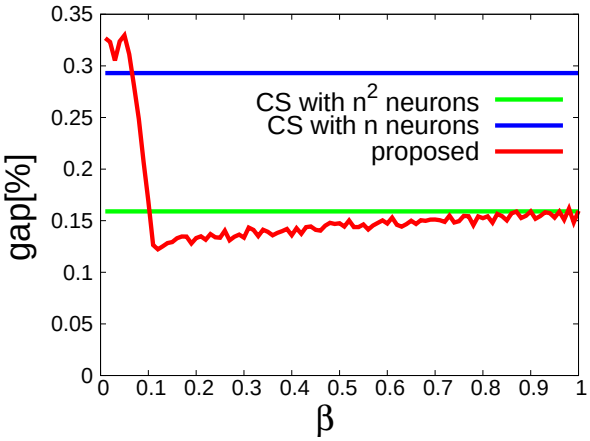
$$\text{gap}[\%] = \frac{\text{found best solution} - \text{optimal solution}}{\text{optimal solution}} \times 100. \quad (6)$$

In this study, we used the following parameter values: $w = 0.0005, k = 0.5^{\frac{1}{n}}, \alpha = 1, \theta = 0.05$ and $\epsilon = 0.002$. We calculated 100 trials for each parameter β, and calculated the average gaps across trials. In the numerical experiments, $\Delta_{ij}(t)$ is normalized by $d_M r_M$ where $d_M = \max_{ij}\{d_{ij}\}$ and $r_M = \max_{ij}\{r_{ij}\}$. d_{ij} and r_{ij} are elements of distance matrix D and flow matrix F.

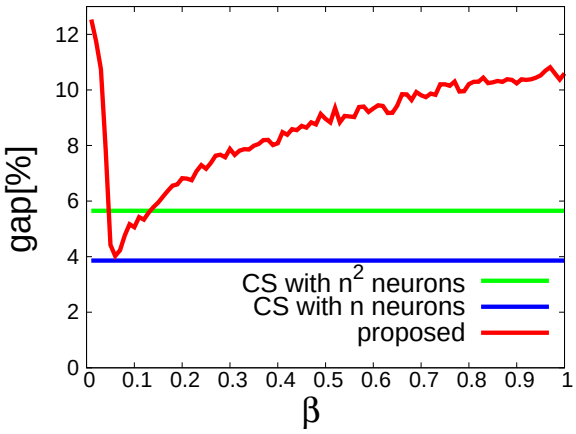
Table 1 shows the obtained best gaps for various parameter values of β. Numerals with bold faced types indicate the best gap, and italic faced types indicate the second best gap. The best parameter β for each problem is shown in parentheses. The third row of Table 1 shows the results with the chaotic search(CS) with n neurons in Ref.[7]. The fifth row of Table 1 shows variances of 100 trials for the best β. From Table 1, even though our method controls the 2-exchange method with small number of neurons, we can get better performance compared to the method with CS with n^2 neurons in Ref.[6] and the method with CS with n neurons in Ref.[7].

To compare the performance of the method, we changed the value of β and evaluated the performance. Figure 1 shows the result of the average gap for each β. In Fig.1, the ordinates show the gap and the abscissas show the values of the parameter β. The results of the method in Ref.[6] is shown by green lines, the method in Ref.[7] is shown by blue lines and the proposed method is shown by red lines. These figures show that the proposed method can get better performance than the methods in Refs.[6, 7]. In particular, in Bur26a and Tai60b, our proposed method can get better performance than that of the methods in Refs.[6, 7] for a wide range of β. In Fig.1 shows the average performance. Next, we investigate the variance of the performance.

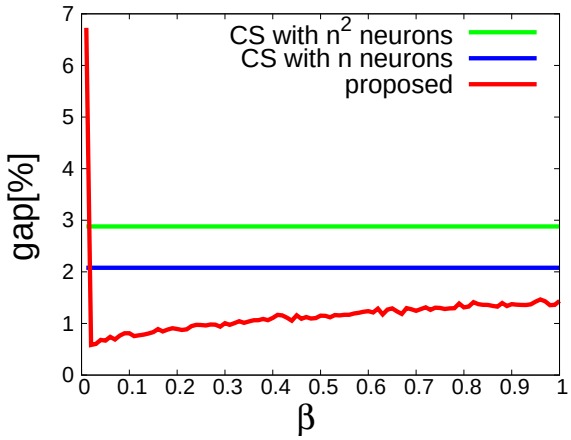
To investigate the variance of the obtained gaps, we show the best gap of one trial in Fig.2. In Fig.2, ordinates show the gap and abscissas show the number of execution times of the 2-exchange method. The result of the method in Ref.[6] is shown by green lines, the method in Ref.[7] is shown by blue lines and the proposed method is shown by red points. These figures show that almost all the red points exist below the green and blue lines. This means that if we select a suitable parameter β, we can get better performance than that of the method in Refs.[6, 7] with high probability.



(a) Bur26a

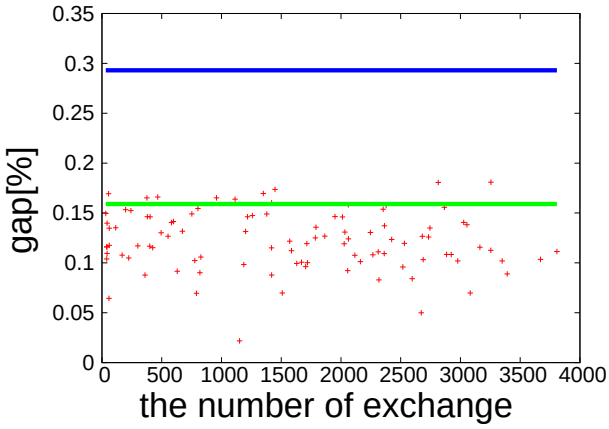


(b) Ste36a

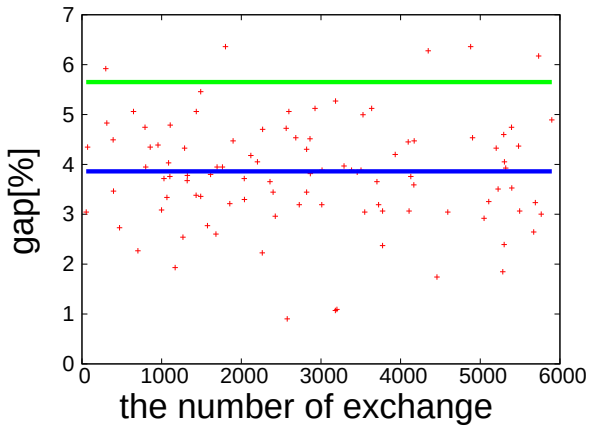


(c) Tai60b

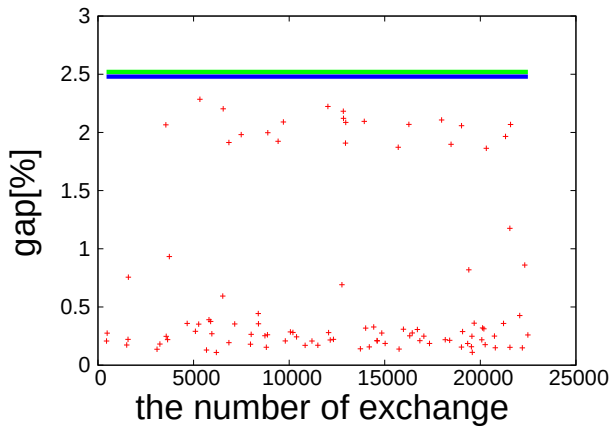
Figure 1: Results of the average gaps for each β.



(a) Bur26a



(b) Ste36a



(c) Tai60b

Figure 2: Results of each gap for the best β. The chaotic search(CS) with n^2 neurons in Ref.[6] is shown in green lines, the chaotic search (CS) with n neurons in Ref.[7] is shown in blue lines and the proposed method is shown in red points.

5. Conclusion

In this paper, we proposed a new algorithm for solving Quadratic Assignment Problem (QAP) by chaotic neural dynamics. Concretely, we changed the effect of external input: the worse the exchange is, the stronger the input is applied. This effect enables a wide range search. By comparing with the performance of the method of chaotic search with n^2 neurons in Ref.[6] and the chaotic search with n neurons in Ref.[7], our method can get better performance. However, the performance of the proposed method largely depends on value of the parameter β. Therefore it is an important future work to construct a parameter tuning method.

Acknowledgment

The authors would like to thank Mr. S. Ishii and AGS Corporation for their kind encouragements on this research. The research is supported by Grant-in-Aid (No.26880020, No.15K12137, No.24650116, No.15KT0112) from JSPS.

References

[1] "QAPLIB - A Quadratic Assignment Problem Library," URL:http://www.opt.math.tu-graz.ac.at/qaplib/

[2] F. Glover, "Tabu search-part I," ORSA Journal on computing, Vol.1, pp.190–206, 1989.

[3] F. Glover, "Tabu search-part II," ORSA Journal on computing, Vol.2, pp.4–32, 1990.

[4] D.M. Tate and A.E. Smith, "A genetic approach to the quadratic assignment problem," Computers & Operations Research, Vol.22, pp.73–83, 1995.

[5] M. Hasegawa, T. Ikeguchi, and K. Aihara, "Solving large scale traveling salesman problems by chaotic neurodynamics," Neural Networks, Vol.15 (2), pp.271–283, 2002.

[6] M. Hasegawa, T. Ikeguchi, K. Aihara and K. Itoh, "A novel chaotic search for quadratic assignment problems," European Journal of Operational Research, Vol.139, pp.543–556, 2002.

[7] H. Ohnishi, Y. Shimada, K. Fujiwara and T. Ikeguchi, "A Chaotic Search with Small Memory Consumption for Solving QAP," Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA2015), pp.610–613, 2015.

[8] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," Physics Letters A, Vol.144, pp.333–340, 1990.