2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

A chaotic search with small memory consumption for solving QAP

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

†Department of Management Science, Graduate School of Engineering, Tokyo University of Science
1–3 Kagurazaka, Shinjuku-ku,Tokyo, Japan

Abstract—Quadratic assignment problem (QAP) is a typical example of NP-hard problems. Therefore, we need to develop algorithms for finding good approximate solutions in a reasonable time frame. In this paper, we proposed a new algorithm that controls execution of the 2-exchange method, which is one of the heuristic algorithms for solving QAPs by chaotic dynamics. In the proposed algorithm, we modified the assignment of neurons and proposed a parameter tuning method for connection weights. As a result, our algorithm can find not only good solutions but also reduce the memory consumption.

1. Introduction

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

On the other hand, several approximate algorithms are proposed for solving Quadratic Assignment Problem (QAP). For example, the 2-exchange method is well known to solve QAP. However, local searches such as the 2-exchange method are generally trapped into local minima. For this reason, many methods to escape from the local minima have also been proposed: for example, tabu search[1, 2], genetic algorithm[3], chaotic dynamics[4, 5] and so on. In Ref.[4], the 2-opt method for solving traveling salesman problems is controlled by the chaotic dynamics, while in Ref.[5], the 2-exchange method for solving QAPs is controlled by the chaotic dynamics. In Refs.[4, 5], the chaotic dynamics is introduced to control the heuristic algorithms. In this paper, we propose an algorithm for solving QAPs based on Ref.[5]. In the proposed algorithm, the assignment of neurons is modified to reduce memory consumption. In addition to control chaotic dynamics, we adaptively decide a parameter depending on the state of the solution. As a result, we succeeded to improve the performance of the proposed algorithm.

2. QAP

The QAP is one of the most difficult NP-hard combinatorial 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 defined by Eq.(1):

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

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 explain the algorithm of the 2-exchange method for solving QAPs.

Step1： A random solution $\boldsymbol{q}$ 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 $F(\boldsymbol{q}) > F(\boldsymbol{q'})$, 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 local minima. Therefore, we used a chaotic dynamics to escape from the local minima.

3. Proposed algorithm

The dynamics of the chaotic neuron i in the chaotic neural network[6] is described as follows:

$$\begin{cases} \xi_i(t+1) = k_e \xi_i(t) + \sum_{j=1}^{m} v_{ij} a_j(t), & (2) \\ \eta_i(t+1) = k_f \eta_i(t) + \sum_{j=1}^{l} w_{ij} h_{ij}(x_j(t)), & (3) \\ \zeta_i(t+1) = k_r \zeta_i(t) - \alpha x_i(t) + \theta_i, & (4) \end{cases}$$

$$x_i(t+1) = f\left\{\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\right\}. \quad (5)$$

To control the 2-exchange method by the chaotic neural network, we transformed Eqs.(2) ~ (5) into Eqs.(6) ~ (9).

$$\begin{cases} \xi_j(t+1) = \beta\Delta_{ij}(t), & (6) \\ \eta_j(t+1) = k_f\eta_j(t) + \displaystyle\sum_{i=1,(i\neq j)}^{n} w_{ji}(t)x_i(t), & (7) \\ \zeta_j(t+1) = k_r\zeta_j(t) - \alpha x_j(t) + (1-k_r)\theta_j, & (8) \end{cases}$$

$$x_j(t+1) = f\left\{\xi_j(t+1) + \eta_j(t+1) + \zeta_j(t+1)\right\}, \quad (9)$$

where $\xi_j(t)$ is an external input to the chaotic neuron j, $\eta_j(t)$ is a feedback input from other neurons in the network to the chaotic neuron j, $\zeta_j(t)$ 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_f and k_r are decay constants, $w_{ji}(t)$ is a connection weight from the chaotic neuron i to the chaotic neuron j at time t, α is a scaling parameter of refractoriness effect, θ_j is a threshold of the chaotic neuron j, and f is a sigmoidal function defined by $f(y) = 1/(1 + e^{-y/\epsilon})$.

In the conventional method[5], when the problem size is n, the $n \times n$ chaotic neurons are prepared to represent each (i, j) assignment. If the chaotic neuron (i, j) fires, the element i is assigned to the location j. Although this method[5] shows good performance, this method uses much memory. On the other hand, in the proposed method, we use n chaotic neurons for solving the problem of size n. For this reason, we can reduce the memory consumption. If the chaotic neuron i fires, we perform the 2-exchange method for the element i. We explain the proposed 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.(6) ~ (8).

Step3: The output of the chaotic neuron j is calculated by using Eq.(9).

Step4: If $\max_j\{x_j(t+1)\} > 1/2$, the chaotic neuron j fires and the element $p(i)$ and $p(j)$ are exchanged by the 2-exchange method.

Step5: If $i = n$, this iteration is finished. Otherwise let $i = i + 1$ and return to **Step2**.

4. The parameter tuning method

In this paper, we also introduce a parameter tuning method for deciding connection weights. The connection weight w_{ij} is controlled as follows. First, $w_{\max} = \max_{ij}\{D_{ij}R_{p(i)p(j)}\}$, the largest matrix element of the product of the distance matrix D and the flow matrix R, is calculated. Then, the connection weight from the neuron j to the neuron i is decided by Eq.(10):

$$w_{ij} = \frac{D_{ij}R_{p(i)p(j)}}{w_{\max}}. \quad (10)$$

If the elements $p(a)$ and $p(b)$ are exchanged by the 2-exchange method, the parameters w_{ai}, w_{bi}, w_{ia}, and $w_{ib}(i = 1, 2, \ldots, n)$ are updated by Eq.(10).

5. Results

Table 1 shows the parameter values that we used for solving each problem. In the numerical experiments, $\Delta_{ij}(t)$ is normalized by $d_M r_M$ where $d_M = \max_{ij}\{d_{ij}\}$ and $r_M = \max_{ij}\{r_{ij}\}$. We evaluated the performance of the proposed algorithm using benchmark problems from QAPLIB[7]. To evaluate the performance, we used the gap which is defined by the following Eq.(11).

$$\text{gap}[\%] = \frac{\text{found best solution} - \text{optimal solution}}{\text{optimal solution}} \times 100. \quad (11)$$

We calculated 10 trials for each parameter, and calculated the average gap across trials.

Table 1: The parameters that we used for solving each problem.

Problem	α	β	θ	k	ϵ
Bur26a	5	1400000	0.05	variable	0.002
Bur26b	5	1400000	0.05	variable	0.002
Bur26c	5	1400000	0.05	variable	0.002
Bur26d	5	1400000	0.05	variable	0.002
Ste36a	5	10000	0.05	variable	0.002
Ste36b	1	10000	0.05	variable	0.001
Ste36c	1	10000000	0.05	variable	0.0001
Tai20b	5	variable	0.05	0.9	0.0001
Tai30b	2	variable	0.05	0.9	0.00005
Tai40b	1	variable	0.05	0.9	0.0001
Tai50b	1	variable	0.05	0.9	0.0001
Tai60b	1	variable	0.05	0.9	0.0001
Tai80b	1	variable	0.05	0.9	0.0001
Tai150b	10	variable	0.05	0.9	0.009

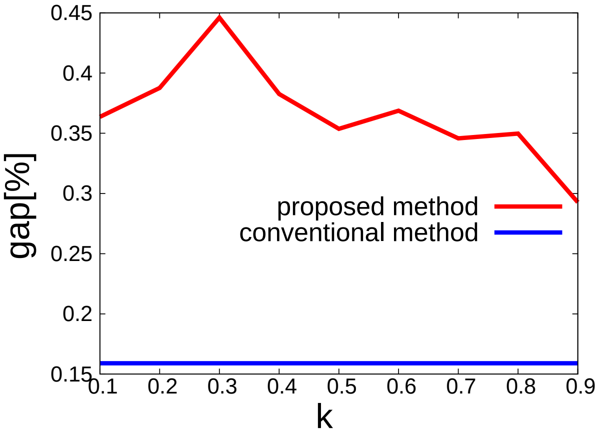
Table 2 shows the best gaps for all parameters. Numerals with bold faced types indicate the best gap. From Table 2, even though our method consumes less memories, we can get almost equivalent performance to the conventional method[5].

Table 2: Results of gaps[%] for (i) the conventional method with the chaotic search(CS), and (ii) the proposed method. The best parameters for each problem are shown in parentheses. The values of the parameter k are shown in BurXXX and SteXXX, and the values of the parameter β are shown in TaiXXXb.

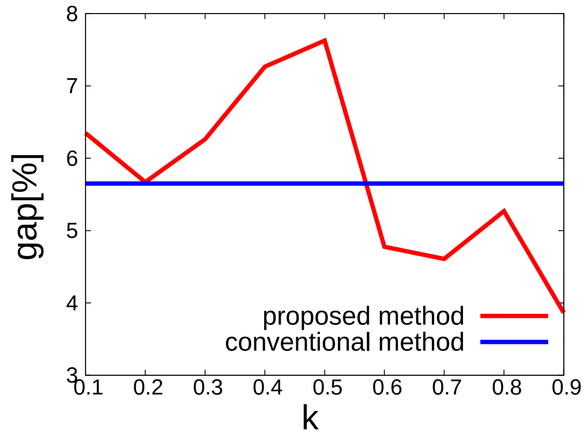
Problem	(i) Conventional method	(ii) Proposed method
Bur26a	**0.159**	0.293 (0.9)
Bur26b	**0.0814**	0.111 (0.3)
Bur26c	**0.0496**	0.130 (0.1)
Bur26d	**0.0234**	0.080 (0.7)
Ste36a	5.65	**3.86 (0.9)**
Ste36b	12.7	**8.29 (0.9)**
Ste36c	4.40	**3.68 (0.1)**
Tai20b	**1.80**	3.14 (15000000)
Tai30b	2.33	**1.91 (15000000)**
Tai40b	**3.70**	4.58 (10000000)
Tai50b	**2.21**	3.96 (10000000)
Tai60b	2.52	**2.48 (13000000)**
Tai80b	2.88	**2.08 (11000000)**
Tai150b	2.44	**1.46 (100000)**

Figure 1 shows the change of the gap in case of changing k. In Fig.1, red line indicates the result of the proposed algorithm, blue line indicates the result of the conventional algorithm[5] with chaotic search (CS). The conventional method has higher performance for any value of k in Fig.1(a)(Bur26a). However, as shown in Fig.1(b)(Ste36a), the proposed method finds better solutions than the conventional method with chaotic search depending on k. In Fig.1(c)(Ste36b), we find that the proposed method has higher performance than the conventional method with chaotic search for almost any value of k.

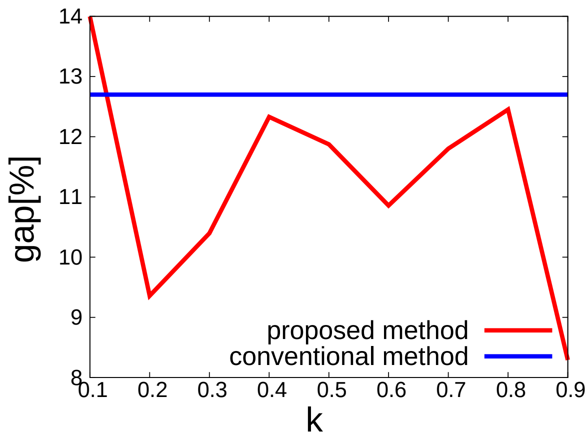
Figure 2 shows the temporal change of the internal state of the chaotic neuron 1. From Fig.2, temporal behavior of the chaotic neuron looks like chaotic.



(a) Bur26a

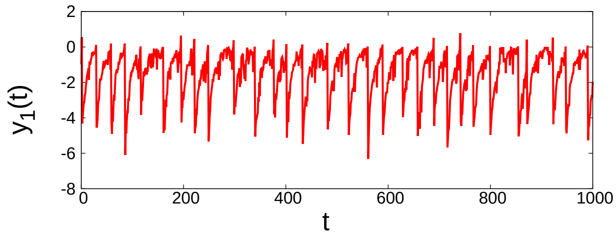


(b) Ste36a

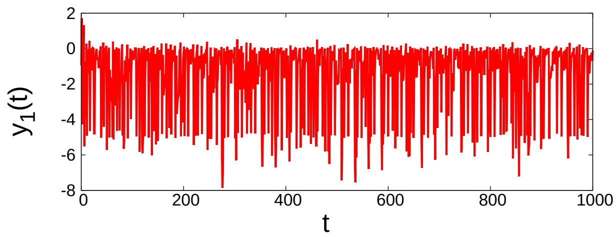


(c) Ste36b

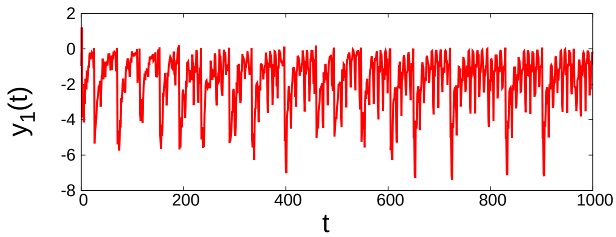
Figure 1: The result of the average gap for each k. Ordinates show the gap and abscissas show values of the parameter k. The result of the proposed method is shown in red lines and CS is shown in blue lines.



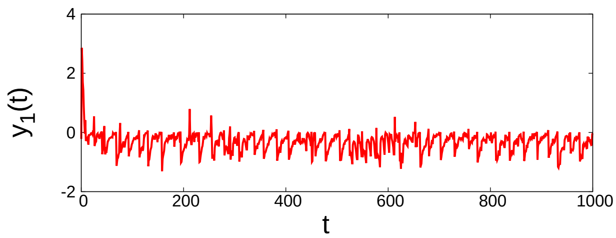
(a) Bur26a



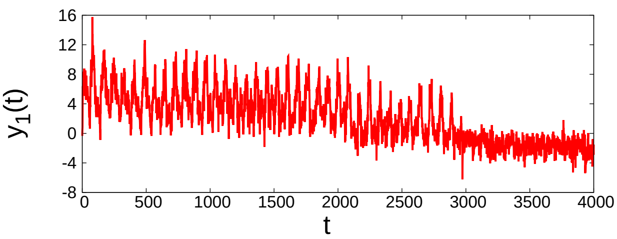
(b) Bur26b



(c) Ste36a



(d) Ste36b



(e) Tai80b

Figure 2: The value of the internal state of the chaotic neuron 1 for each times of update. Ordinates show the internal state of the chaotic neuron 1 and abscissas show the times of update of the chaotic neuron 1.

6. Conclusion

We proposed a new algorithm to find good approximate solutions for QAP. We also introduced a parameter tuning method to get good solutions for any kind of problems. In comparison with the conventional algorithm[5], our algorithm can get equivalent performance. However our algorithm has an advantage, because the number of neuron is reduced, which means that, we can reduce the memory consumption. It is an important future work to introduce chaotic simulated annealing into the proposed method and compare its performance with the conventional chaotic simulated annealing method[5].

Acknowledgment

The authors would like to thank Mr. S. Ogawa and AGS Corp. for their kind encouragements on this research. The research of Y.S. is supported by Grant-in-Aid for Research Activity Start-up (No. 26880020) from JSPS. The research of K.F. was partially supported by Grant-in-Aid for Challenging Exploratory Research (No 15K12137) from MEXT of Japan. The research of T.I. was partially supported by Grant-in-Aid for Exploratory Research (No. 24650116) from JSPS and by Grant-in-Aid for Scientific Research (C) (Generative Research Fields) (No.15KT0112) from JSPS.

References

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

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

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

[4] 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.

[5] 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.

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

[7] "QAPLIB - A quadratic assignment problem library," URL:http://www.opt.math.tu-graz.ac.at/qaplib/