2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

Slide-and-Insert Assignment Method with Chaotic Dynamics for Quadratic Assignment Problems

Yusuke Sakamoto† and Yoshihiko Horio†

†Graduate School of Engineering, Tokyo Denki University
2–2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo, 101–8457 Japan
Email: 10kme34@ms.dendai.ac.jp, horio@eee.dendai.ac.jp

Abstract—A quadratic assignment problem (QAP) is an NP-hard combinatorial optimization problem. Some heuristic algorithms for the QAPs is proposed in order to find the sub-optimum solutions efficiently. In this paper, we propose a novel heuristic algorithm for the QAPs based on the Or-opt algorithm used for traveling salesman problems. The proposed method introduces a slide-and-insert operation for the assignments of the elements in the QAP. Furthermore, we improve the proposed algorithm by combining it with the 2-opt algorithm. We also use chaotic dynamics instead of the random numbers. Numerical simulation results, which compare the solving performance of the proposed algorithms with the random numbers and the chaotic dynamics, are shown.

1. Introduction

A quadratic assignment problem (QAP)[1] is one of the combinatorial optimization problems, and belongs to the class of NP-hard. For large-size QAPs, it would take an impractical time to obtain the optimum solution. Therefore, heuristic algorithms have been proposed in order to find the sub-optimum solutions in reasonable time.

In this paper, we propose a slide-and-insert assignment method, which is based on the Or-opt algorithm for traveling salesman problems (TSPs)[2], which is another NP-hard optimization problem. We also introduce chaotic dynamics into the proposed method. We compare the performance of the proposed method with the chaotic dynamics and that with the random numbers through numerical simulations.

1.1. Traveling Salesman Problem (TSP)

The TSP seeks the shortest route that visits each city only once, and returns to the starting point. The length of a tour can be expressed as eq. (1).

$$\text{length} = \sum_{i=1}^{n-1} C(p(i), p(i+1)) + C(p(n), p(1)) \qquad (1)$$

where $p(i)$ $(1 \leq i \leq n)$ is the element of the permutation $\boldsymbol{p}$, which gives a feasible solution, $C(a, b)$ is the distance between cities a and b, and n is the number of cities. There are $(n-1)!/2$ possible routes for the size-n TSP, so that it is impossible to obtain the exact solution for reasonable time if the number of cities increases. Therefore, many heuristic algorithms were proposed to obtain the sub-optimum solutions in reasonable time. The Or-opt algorithm is one of these heuristic algorithms [3].

1.2. Or-opt Algorithm

In the Or-opt method [3], we select successive 1 to 3 cities (block). We then insert them into another path.

Fig. 1 shows the schematic diagram of the Or-opt algorithm. In the example shown in the figure, the block starts from the city 5 with the block size of 3. The block is inserted between the city 4 and the city 1. First, we take out the city 5 to the city 7 from the tour. Next, we connect the path between the city 2 and the city 3. In addition, the path from the city 4 to the city 1 is removed in order to insert the block. Finally, the block is inserted to the empty path to make a tour.

1.3. Quadratic Assignment Problem (QAP)

The QAP can be described as follows, given two matrices $\boldsymbol{A}$ (distance matrix) and $\boldsymbol{B}$ (flow matrix). The objective of the QAP is to find a permutation $\boldsymbol{p}$ of the elements which minimizes the following object function F given by

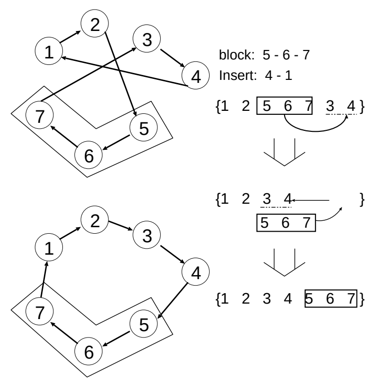

Figure 1: A schematic diagram of the Or-opt algorithm. For example, we create a block consists of the city 5 to the city 7. Then we insert the block in the path between the city 4 and the city 1.

$$F = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{p(i)p(j)} \tag{2}$$

$$\begin{aligned} \text{index}: &\ 1, \quad 2, \quad \cdots, \ n \\ \boldsymbol{p}: &\ \{p(1),\ p(2),\ \cdots,\ p(n)\} \end{aligned} \tag{3}$$

where a_{ij} and b_{ij} are the (i,j)th elements of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively, $p(i)$ is the ith element of $\boldsymbol{p}$, and n is the problem size. There are $n!$ total combinations for the QAP, therefore, it is hard to obtain the exact solution in the case of large-scale QAPs.

2. Slide-and-Insert Assignment Method with Chaotic Neurodynamics

2.1. Slide-and-Insert Assignment Method

The slide-and-insert assignment method shown in Fig. 2 is based on the Or-opt algorithm for the TSPs.

First, we arbitrarily choose a city. The element assigned to the city is the starting point of the block.

Next, we select the city that will be the insert point of the block. For example, if the insert-point is the city r, the block is inserted between the element $p(r)$ and the element $p(r+1)$.

In addition, we choose the block size which is less than or equal to 3. We need to make a space between $p(r)$ and $p(r+1)$ to insert the block. As shown in Fig. 2(a), we temporarily take out the block from the permutation $\boldsymbol{p}$, so that the cities of the block are now empty. The elements assigned to the cities next to the end of the block are reassigned to the empty cities by sliding them to create an empty space for the insert of the block as shown in Fig. 2(b). This reassignment is referred to as the slide operation.

In the slide operation, the elements move in one way to the descending order of the city numbers. In addition, if the enough numbers of successive elements for

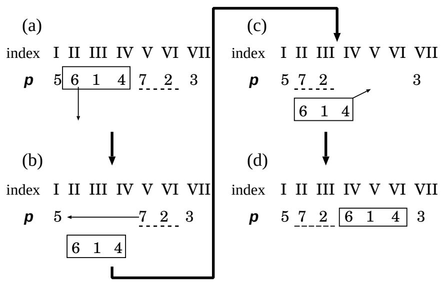

Figure 2: A schematic diagram of the slide-and-insert assignment method. In this example, the starting point of the block is the city II, and the block size is 3. (a) We temporarily take out the block from the permutation $\boldsymbol{p}$. In other words, the cities II, III, and IV are made empty. (b) The elements of the city V to the city VI are reassigned to the empty cities through the sliding operation. (c) Finally, the elements of the block are inserted into the insert point, the city IV.

the slide operation cannot prepared, we use the elements assigned at city I, II, III, and so on.

Finally, the elements of the block are inserted into the cities which became empty as a result of the slide operation as shown in Figs. 2(c) and 2(d). This operation corresponds to the insert operation.

It should be noted that the slide-and-insert assignment method sometimes alters the solution dramatically. As a result, it cannot spend enough time for the local search. Thus, we take advantage of this property. That is, we first use the 2-opt algorithm (Fig. 3) for the local search [4]. When the 2-opt algorithm is trapped in the local minimum, we then apply the slide-and-insert assignment method to escape from the local minimum.

Fig. 3 shows an example of the 2-opt algorithm. First, we select the element i. Next, we select the city j to which we assign the element i. At the same time, the element $p(j)$ that was assigned to the city j is assigned to the city $q(i)$. The objective function will be improved until the algorithm is trapped in the local minimum because the 2-opt algorithm is a local search method.

In contrast, the objective function is rarely improved by the slide-and-insert assignment method. Thus, we use random numbers in the slide-and-insert operations in order to explore a large solution space regardless of the current solution.

Moreover, we introduce the chaotic dynamics into the slide-and-insert assignment method instead of the random numbers to further control the searching space.

2.2. Chaotic Neural Network

In this paper, we use the chaotic neural network model [5] as described in eqs. (4) to (7) [6][7].

$$\xi_{ij}(t+1) = \max_{k}\{\beta\Delta_{ijk}(t)\} \tag{4}$$

$$\zeta_{ij}(t+1) = k_r\zeta_{ij}(t) - \alpha x_{ij}(t) + \theta(1-k_r) \tag{5}$$

$$y_{ij}(t+1) = \xi_{ij}(t+1) + \zeta_{ij}(t+1) \tag{6}$$

$$x_{ij}(t+1) = \frac{1}{1+\exp(\frac{-y_{ij}(t+1)}{\epsilon})} \tag{7}$$

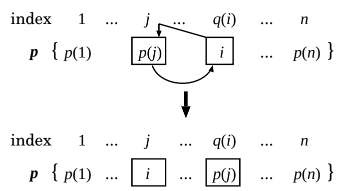

Figure 3: An example of the 2-opt algorithm where the element i is assigned to the jth location, and element $p(j)$ is assigned to the $q(i)$th location.

where $\xi_{ij}(t+1)$ is the gain effect, $\zeta_{ij}(t+1)$ is the refractory effect, β is a scaling parameter of the gain effect, k_r is a decay parameter of the refractory effect, α is a scaling parameter of the refractory effect, θ is a bias, $y_{ij}(t+1)$ is the internal state, ϵ is the steepness of the output function, $\Delta_{ijk} = F_0 - F_{ijk}$ is the gain of the objective function value, F_0 is the current value of the objective function F, and F_{ijk} is the value of F after the slide-and-insert assignment method.

In addition, we pay attention to the number of change in the assignments (changes-in-assignments) through the slide-and-insert assignment. In the example shown in Fig. 2, the elements of the city II to the city VI are reassigned. Therefore, the changes-in-assignments is 5. In the slide-and-insert assignment algorithm, $j \neq 1$ and 2. This is because when j=1, there is no change in $\boldsymbol{p}$, and when j=2, the method is equivalent to the 2-opt algorithm.

Fig. 4 shows a schematic diagram for the slide-and-insert assignment method with the chaotic neural network. As shown in the figure, we prepare $n \times n$ in the chaotic neurons configuring a chaotic neural network for the size-n QAPs. In the figure, i represents the starting point of the block, and j corresponds to the changes-in-assignment. As shown in Fig. 4, $2n$ neurons will not be updated because $j \neq 1$ and 2. As a result, we can reduce the computational time by $2n$ neurons exploiting the changes-in-assignments.

Fig. 5 shows a flow chart for the proposed method with the chaotic dynamics. First, we generate the initial permutation using the random numbers. As shown in the figure, we next select the neuron which has not been updated yet. Then, we determine the block size by evaluating the gains for the block sizes from 1 to 3. The block size will be the size that gives the largest gain. After the block size is determined, we update the internal state of the neuron. If $x_{ij}(t+1) > 0.5$ (the neuron fires), then we apply the slide-and-insert assignment method.

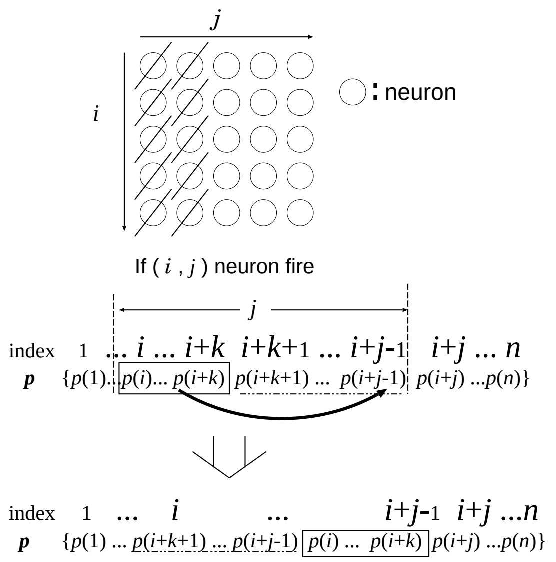

Figure 4: The slide-and-insert assignment method with the chaotic neural network. If the (i, j)th neuron fires, we set the head of the block to the ith city. We then use the slide-and-insert assignment method with the changes-in-assignment of j.

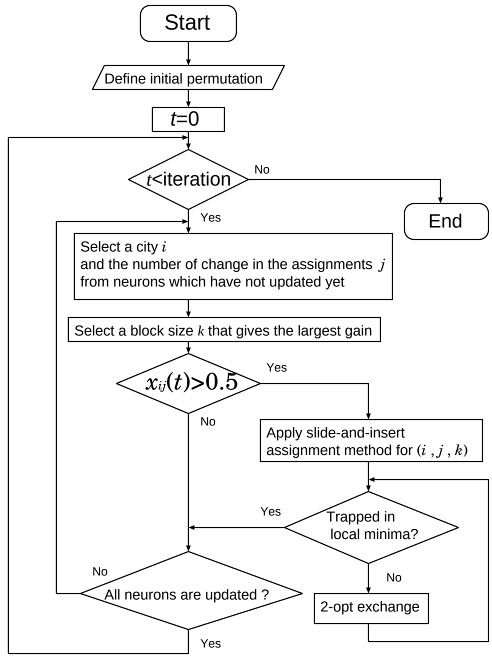

Figure 5: A flow chart of the proposed method with chaotic dynamics.

For example, if the $(2, 5)$-neuron fires, the element is assigned to the city II is the start point of the block, and assigned elements from the city II to the city VI are reassigned using the slide-and-insert assignment method. Because $\boldsymbol{p}$ is changed largely, the searching space may also be changed dramatically. Thus, we should check whether the sub-optimum solution exists in the current solution space by the 2-opt algorithm. We continue to apply the 2-opt exchanges until the algorithm is trapped to the local minimum.

One iteration is completed when all the neurons in the network are updated.

3. Simulation Results

We show the numerical simulation results for the proposed method with the chaotic neurodynamics. The results with the random numbers instead of the chaos are also shown for comparison. The random number was generated by the rand{} function in C. For both methods, we execute 1000 iterations and 30 trials.

We evaluate the average error rate (AER) from the optimum solution defined by

$$\mathrm{AER} = \frac{1}{30}\sum_{t=1}^{30}(F_{min}(t) - F_{opt}) \times 100 \ [\%] \tag{8}$$

Table 1: The parameters for the chaotic neural network

Size	Instance	Parameters				
		α	β	k_r	ϵ	θ
12	nug12	0.5	0.005	0.8	0.05	2
	chr12a	0.5	0.000025	0.8	0.05	2
	chr12b	0.5	0.000025	0.8	0.05	2
	rou12	0.5	0.000025	0.8	0.05	2
	had12	0.5	0.000025	0.8	0.05	2
	tai12a	0.5	0.000025	0.8	0.05	2
20	nug20	0.5	0.005	0.8	0.05	2
	chr20a	0.5	0.0002	0.8	0.05	2
	chr20b	0.5	0.00015	0.8	0.05	2
	rou20	0.5	0.000025	0.8	0.05	2
	had20	0.5	0.005	0.8	0.05	2
	tai20a	0.5	0.000005	0.8	0.05	2
30	nug30	0.5	0.00075	0.8	0.05	2
	lipa30a	0.5	0.0075	0.8	0.05	2
	lipa30b	0.5	0.000025	0.8	0.05	2
	kra30a	0.5	0.00003	0.8	0.05	2
	kra30b	0.5	0.00003	0.8	0.05	2
	tho30	0.5	0.000025	0.8	0.05	2

where $F_{min}(t)$ is the minimum objective function obtained during the t-th trial, and F_{opt} is the optimum objective function.

The network parameters for the chaotic neural network used in the simulation for each problem are shown in Table 1.

Table 2 shows the AER for each problem with the chaotic dynamics and random numbers.

As shown in Table 2, the proposed method with the chaotic dynamics is better than that with the random numbers. However, as shown in Table 1, the value of β are different for each problem. This is because the gains obtained by the slide-and-insert assignment method are different depending on the problems. Therefore, the dynamics of the neuron in the neural network may change according to the characteristics of the problems.

4. Conclusions

In this paper, we have proposed the slide-and-insert assignment method for the QAPs based on the Or-opt algorithm for the TSPs. We have also introduced the chaotic dynamics to the proposed method.

Numerical simulation results have shown that the proposed method with the chaotic dynamics is more effective than that with the random numbers. However, the searching ability is affected by the value of β. Therefore, we need to investigate the dynamics of the proposed method in order to determine the optimal parameter set.

Acknowledgment

This work was supported by Kakenhi (20300085).

Table 2: The AER

Size	Instance	AER	
		Random [%]	Chaos [%]
12	nug12	0.12	0.00
	chr12a	1.62	0.00
	chr12b	0.00	0.00
	rou12	0.13	0.00
	had12	0.00	0.00
	tai12a	0.00	0.00
20	nug20	0.54	0.00
	chr20a	12.65	0.00
	chr20b	7.98	0.00
	rou20	1.00	0.53
	had20	0.01	0.00
	tai20a	1.96	0.33
30	nug30	1.51	0.10
	lipa30a	1.59	0.00
	lipa30b	5.68	0.00
	kra30a	2.79	0.00
	kra30b	1.62	0.015
	tho30	1.48	0.55

References

[1] R. E. Burkard, S. E. Karisch and F. Rendl, "QAPLIB - A Quadratic Assignment Problem Library," avaible via world wide web to http://www.opt.math.tugraz.ac.at/qaplib

[2] "TSPLIB95," http://www.iwr.uniheidelberg.de/iwr/compt/soft/TSPLIB95/TSPLIB.html, 1995

[3] T. Matsuura and T. Ikeguchi, "Chaotic search for traveling salesman problems by using 2-opt and Or-opt algorithms," ICANN 2008, Part II, LNCS 5164, pp. 587–596, 2008

[4] E. Taillard, "Robust tabu search for the quadratic assignment problems," Parallel computing, vol.17, pp. 443–455, 1991.

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

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

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