IEICE Proceeding Series

A Chaotic Local Search Algorithm with Adaptive Exchange of Elements for Quadratic Assignment Problems

Akio Watanabe, Kaori Kuroda, Kantaro Fujiwara, Tohru Ikeguchi

Vol. 2 pp. 362-365
Publication Date: 2014/03/18
Online ISSN: 2188-5079

A Chaotic Local Search Algorithm with Adaptive Exchange of Elements for Quadratic Assignment Problems

Akio Watanabe†, Kaori Kuroda†, Kantaro Fujiwara†, and Tohru Ikeguchi†‡

†Graduate School of Science and Engineering, Saitama University
‡Saitama University Brain Science Institute
255 Shimo-Ohkubo, Sakura-ku, Saitama-city, Saitama, 338-8570 Japan
Email: {watanabe,kuroda,kantaro,tohru}@nls.ics.saitama-u.ac.jp,

Abstract—The quadratic assignment problem (QAP) is one of the NP-hard combinatorial optimization problems. Then, it is required to develop an effective approximation algorithm for finding near optimal solutions in realistic time. In this paper, we proposed a local search algorithm for solving QAP by introducing searching property of the Lin-Kernighan algorithm. We also applied chaotic dynamics to the proposed local search algorithm to escape from undesirable local minima. To evaluate solving performance of the proposed algorithm, we compared the performance of the proposed algorithm with those of the conventional algorithms. As a result, the solving performance of the proposed algorithm with chaotic dynamics exhibits smaller gaps from optimal solutions than the conventional algorithms.

1. Introduction

In our daily life, we are often asked to solve combinatorial optimization problems, such as drilling problem, VLSI design, scheduling, and delivery plan problem. It is important to solve these combinatorial optimization problems in terms of reduction of the cost. However, we cannot usually get an optimal solution because these combinatorial optimization problems belong to a class of NP-hard. Thus, it is necessary to develop effective approximate algorithms.

The quadratic assignment problem (QAP)[1] is one of the NP-hard problems. It formulates various real problems. The purpose of solving QAP is, for example, to find an assignment of facilities to each city that makes a total cost minimum when the relation of facilities and relation of cities are given.

On the other hand, several approximate algorithms for solving traveling salesman problem (TSP), which is a special case of QAP, have already been proposed, such as 2-opt, Or-opt and Lin-Kernighan (LK) algorithm[2]. Among them, the LK algorithm is the most powerful algorithm for finding a superior solution of TSP. The LK algorithm controls the λ-opt algorithm. It realizes an effective search by changing λ adaptively in its search process.

In this paper, at first, we propose a new algorithm with a searching property of the LK algorithm; exchange elements are adaptively decided. Then we conducted numerical experiments to compare the performance of the proposed local search algorithm with the conventional algorithms by using the benchmark problems of QAPLIB[1]. As a result, the proposed local search algorithm found superior solutions with less calculation cost than the λ-opt algorithm.

In this paper, we also introduce an algorithm of a metaheuristic strategy to escape from undesirable local minima, because the proposed algorithm for QAP has a risk of being trapped at local minima because the searching dynamics is steepest descent. It has already been reported that metaheuristic strategy with chaotic dynamics is effective for solving combinatorial optimization problem[3, 4, 5, 6]. Therefore, it is very natural to expect that chaotic dynamics can realize an effective strategy to escape from local minima. Then, we also introduce chaotic dynamics into the proposed algorithm to escape from undesirable local minima.

2. Quadratic Assignment Problems

The quadratic assignment problem (QAP) is a typical example of an NP-hard combinatorial optimization problem. It includes various combinatorial optimization problems as a special case. In the QAP, when two $n \times n$ matrices, a distance matrix $A = (a_{ij})$ and a flow matrix $B = (b_{kl})$ are given, we are asked to find an assignment $\boldsymbol{p} = \{p_{(1)}, p_{(2)}, ..., p_{(n)}\}$ that minimizes the objective function. It is the purpose of QAP. Here, distance a_{ij} means the distance between cities i and j, the flow b_{kl} means flow from the facility k to l. In QAP, the objective function is defined as:

$$F(\boldsymbol{p}) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)}. \quad (1)$$

However, it is almost impossible to get an optimal solution, because the number of assignments is $n!$ for the n-size QAP. It means that the number of possible solutions increases exponentially. If we can develop a heuristic algorithm, a superior solution can be provided in realistic time.

2.1. Conventional local search algorithm

When a solution set is provided by transforming the present solution $\boldsymbol{p}$, we call it the neighborhood of $\boldsymbol{p}$. The local search for QAP is a repetitive operation to search $\boldsymbol{p}'$ that satisfies $F(\boldsymbol{p}') < F(\boldsymbol{p})$. Once a better solution $\boldsymbol{p}'$ is found, the present solution $\boldsymbol{p}$ is really updated to $\boldsymbol{p}'$.

The λ-opt algorithm is the most typical heuristic algorithm for searching better solution of QAP. We illustrate the λ-opt algorithm as follows.

Step1 Facilities are assigned to cities randomly. Let us describe the solution as $\boldsymbol{p}$.

Step2 λ facilities are chosen from $\boldsymbol{p}$, and are exchanged each other. Let us describe a solution of the neighborhood of $\boldsymbol{p}$ as $\boldsymbol{p}'$. If $\boldsymbol{p}'$ satisfies $F(\boldsymbol{p}') < F(\boldsymbol{p})$, $\boldsymbol{p}$ is updated to $\boldsymbol{p}'$.

Step3 **Step2** is repeated until the update of $\boldsymbol{p}$ stops.

Generally, the λ-opt algorithm with large $\lambda's$ can find superior solutions because the number of candidate solutions increases, when λ becomes large. However, if λ becomes large, calculation costs also increase exponentially. It means that it is almost impossible to finish a search in realistic time.

2.2. Move strategy

The λ-opt algorithm updates the solution $\boldsymbol{p}$ at **Step2**. Here two possible update algorithms exist.

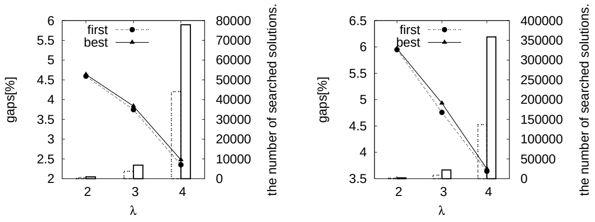
First admissible move strategy
The solution $\boldsymbol{p}$ is updated to $\boldsymbol{p}'$ which is firstly found that satisfies $F(\boldsymbol{p}') < F(\boldsymbol{p})$ from the neighborhood of a present solution.

Best admissible move strategy
The solution $\boldsymbol{p}$ is updated to $\boldsymbol{p}'$ which has the minimum cost from the neighborhood of a present solution.

In Fig.1, we show the results of the average gaps from the optimal solution and the number of the searched solution for these two strategies when the value of λ is changed. The number of trials is 100. We used two instances "Tai15a" and "Tai20a" from QAPLIB[1].

The solid lines express gaps provided as a result of search, and the bars express the number of the searched solutions. The dashed lines describe results of the first admissible move strategy, and the solid lines describe results of the best admissible move strategy.

As a result, the λ-opt algorithm found better solutions by increasing λ. However, the number of searched solutions increases exponentially. Although the best admissible move strategy searches more solutions than the first admissible move strategy, their obtained solutions are not significantly different.



(a) Tai15a. (b) Tai20a.

Figure 1: Average gaps from the optimal solution and the number of the searched solutions for (a)Tai15a and (b)Tai20a in case of applying the λ-opt ($\lambda = 2, 3, 4$) algorithm with the first admissible move and the best admissible more strategies.

3. Proposed algorithm

3.1. Approach

As shown in the previous section, the λ-opt algorithm finds superior solutions by increasing λ. However, the best admissible move strategy which enumerates the neighborhood with a fixed number of exchange elements λ, is less effective than the first admissible move strategy.

Therefore, in the proposed algorithm, we do not search solutions provided by neighboring enumeration such as the best admissible move strategy, but by exchanging elements with changing the λ adaptively. It makes the solution search process with a large number of exchange elements with less calculation cost. It is implemented by deciding exchange elements adaptively, which is a similar strategy as the LK algorithm used for TSP.

3.2. Algorithm

We explain the proposed local search algorithm in the following.

Step1 Facilities are assigned to cities randomly. Let us describe the present solution as $\boldsymbol{p}$.

Step2 Let us describe the best solution in the past search as $\boldsymbol{p}_{best}$. We choose the facility E_1 which is the start point of the exchange from the facilities $f_i (i = 1, 2, ..., n)$. Then let $d = 1$.

Step3 The facility E_{d+1} is chosen which reduces the cost most when the facility is exchanged with E_d from facilities $f_i (i = 1, 2, ..., n)$ except facilities that has already been exchanged $E_k (k = 1, 2, ..., d)$. Then, a new solution $\boldsymbol{p}'$ by exchanging E_d and E_{d+1} is obtained. Then d is increased by 1.

Step4 If $F(\boldsymbol{p}') < F(\boldsymbol{p}_{best})$, $\boldsymbol{p}_{best}$ is updated to $\boldsymbol{p}'$. If $d = n$, go to **Step5**, otherwise return to **Step3**.

Step5 $\boldsymbol{p}$ is updated to $\boldsymbol{p}_{best}$, and this iteration is finished. Then, the procedure returns to **Step2** and the search is repeated with selecting another facility as E_1. If $\boldsymbol{p}$ is not updated even if any facility is chosen as E_1 at **Step2**, the search is terminated.

3.3. Meta heuristics using chaotic neural network

It has been reported that a metaheuristic strategy with chaotic dynamics is useful for solving combinatorial optimization problems. Then, we also applied the metaheuristic strategy with the chaotic neural network to the proposed local search algorithm.

Chaotic neural network is a network constructed by chaotic neurons[7]. The dynamics of the ith chaotic neuron used in this paper are described by the following equations.

$$\xi_i(t+1) = \beta\Delta_i(t), \quad (2)$$

$$\zeta_i(t+1) = -\alpha\sum_{d=0}^{t} k^d x_i(t-d) + \theta, \quad (3)$$

$$\eta_i(t+1) = \begin{cases} -wx_\mu(t) & (\text{if } f_i = E_d), \\ -wx_\nu(t) & (\text{if } f_i = E_{d+1}), \\ 0 & (\text{otherwise}), \end{cases} \quad (4)$$

$$y_i(t+1) = \xi_i(t+1) + \zeta_i(t+1) + \eta_i(t+1), \quad (5)$$

$$x_i(t+1) = \begin{cases} \frac{1}{1+\exp(-\frac{y_i(t+1)}{\epsilon})} & (\text{if } f_i = E_d \text{ or } E_{d+1}), \\ 0 & (\text{otherwise}), \end{cases} \quad (6)$$

where β is the scaling parameter for the gain effect; k is the decay parameter of refractoriness; α is the scaling parameter of refractoriness; w is a synaptic weight; θ is the threshold of the chaotic neuron; ν is a index of firing neuron at time t; μ is a index of firing neuron at time $t+1$; $\Delta_i(t)$ is the difference of the objective function; $\xi_i(t+1)$, $\zeta_i(t+1)$ and $\eta_i(t+1)$ are the internal states of the ith chaotic neuron at time $t+1$. The output of the ith chaotic neuron is $x_i(t+1)$. If $y_i(t+1) = \xi_i(t+1) + \zeta_i(t+1) + \eta_i(t+1)$ is the largest among all the neurons, the ith chaotic neuron fires. By using chaotic dynamics of Eqs.(2)-(6), we modified the proposed algorithm. In the modified algorithm, we used the chaotic neural network constructed by n chaotic neurons for n-size problem.

In the proposed algorithm mentioned in section 3.2, we choose the facility E_{d+1} that makes the total cost minimum by the exchange at **Step3**. On the other hand, in the following algorithm, we choose the facility which is assigned to a firing neuron at **Step3.2**. If the chaotic neuron once fires, it becomes hard to fire for a while due to refractoriness. Therefore if the facility is chosen as E_d, it becomes hard to be chosen again for a while. Thus we avoid the same facility to be a target of the exchange repeatedly, and we search the solution which has not been searched. The algorithm is as follows.

Step1 Facilities are assigned to the cities randomly. Let us describe the solution as $\boldsymbol{p}$. Then we suppose $t = 1$, $\zeta_i(1) = 0$, $\eta_i(1) = 0$ for $i = 1, 2, ..., n$.

Step2 Let us describe the best solution in the past search as $\boldsymbol{p}_{best}$. The facility E_1 is chosen as the start point of the exchange from the facilities $f_i (i = 1, 2, ..., n)$. Then let $d = 1$.

Step3.1 For any chaotic neurons not assigned to E_d, the internal state $\xi_i(t+1)$ is updated by Eq.(2). In Eq.(2), $\Delta_i(t)$ is improvement of the objective function by exchanging f_i and E_d. Then, the total internal states $y_i(t+1)$ is updated by Eq.(5). Then d is increased by one.

Step3.2 From $f_i (i = 1, 2, ..., n)$, the facility E_{d+1} whose total internal states $y_i(t+1)$ of the assigned neuron is the largest is chosen. Then, a new solution $\boldsymbol{p}'$ by exchanging E_d and E_{d+1} is obtained.

Step3.3 Let $\xi_\mu(t+1) = \xi_\nu(t+1)$ where μ is the neuron index assigned to E_d and ν is the neuron index assigned to E_{d+1}. Then, $y_\nu(t+1)$ is updated, and $x_i(t+1)$ of all neurons are updated.

Step4.1 Let t be increased by one. The internal states $\xi_i(t+1)$ and $\zeta_i(t+1)$ of all the neurons are updated by Eq.(2) and Eq.(3).

Step4.2 If $F(\boldsymbol{p}') < F(\boldsymbol{p}_{best})$, $\boldsymbol{p}_{best}$ is updated to $\boldsymbol{p}'$. If $d = n$, go to **Step5**, otherwise return to **Step3**.

Step5 The present solution $\boldsymbol{p}$ is updated to $\boldsymbol{p}_{best}$, and this iteration is terminated. Return to **Step2**, the new iteration starts with selecting another facility as E_1.

4. Numerical results

We evaluate performance of the proposed algorithm using benchmark problems of QAPLIB. For each instance, different 100 initial solutions are prepared. We also compared the performance of a random assignment, 2-opt, 3-opt, the proposed local search algorithm and the proposed local search algorithm with chaotic dynamics. Here, the random assignment means that facilities are assigned randomly. When the proposed algorithm with chaotic dynamics is applied, solutions are improved by the proposed local search algorithm without chaotic dynamics (section 3.2) until the solution falls into local minima, then the solutions are improved by the proposed algorithm with chaotic dynamics (section 3.3) until the solutions are not updated for $40n$ iterations. To evaluate the performance, we used average gaps. The gap is defined by the following equation.

$$\text{gap}(\%) = \frac{\text{found solution} - \text{optimal solution}}{\text{optimal solution}} \times 100. \quad (7)$$

In the proposed local search algorithm with chaotic dynamics, the parameters are set to $k = 0.75, \alpha = 0.25, w = 0.25, \beta = 0.002, \theta = 0$ and $\epsilon = 0$.

Table 1 shows the average gaps for the algorithms. For all instances, the proposed local search algorithm can find better solutions than the 3-opt algorithm. In addition, the proposed local search algorithm with chaotic dynamics can find better solutions than that without chaotic dynamics. This result shows that chaotic dynamics is effective for escaping from local minima.

To compare the performance of the algorithms, iti is also important to evaluate the number of execution of the local algorithm. Then we used the number of executions that the algorithms tried to search solutions as a measure of the performance evaluation. Here, the solutions search means the number of calculating the objective function in the searching process.

Table 2 shows the number of execution of each algorithm. The results show that the proposed local search algorithm and the proposed local search algorithm with the chaotic dynamics searched more times than the 2-opt algorithm but less than the 3-opt algorithm for large n-size QAPs. Then the results indicate that proposed algorithms could find better solutions efficiently.

Table 1: Results of average gaps. Bold faces indicate the best result.

instance	random	2-opt	3-opt	proposed	proposed+CNN
Lipa20a	7.1	2.8	2.6	2.4	**2.3**
Lipa20b	31.3	14.6	13.5	11.3	**10.9**
Lipa30a	4.8	2.0	1.9	**1.7**	**1.7**
Lipa30b	29.1	16.9	15.1	14.1	**14.0**
Lipa50a	3.1	1.3	1.2	**1.1**	**1.1**
Lipa50b	28.8	18.7	17.6	**16.2**	**16.2**
Tai60a	18.1	4.7	3.5	**3.2**	**3.2**
Tai60b	66.4	8.3	6.3	**4.6**	**4.6**
Tai100a	13.9	3.1	2.4	2.1	**1.9**
Tai100b	51.0	4.5	4.3	**3.0**	**3.0**

5. Conclusion

We proposed a new local search algorithm for solving QAP. The proposed local search algorithm decides exchange elements adaptively. As a result, the proposed local search algorithm could effectively find better solutions. Moreover, we can improve the performance of the proposed local search algorithm by introducing chaotic dynamics.

The authors would like to thank Mr. S. Ogawa and AGS Corp. for their kind encouragements on this research. The research of K.K. was supported by Grant-in-Aid for JSPS Fellows (No. 24-9982). The research of K.F. was supported by Grant-in-Aid for Young Scientists (B) (No 24700221) from MEXT of Japan. The research of T.I. was partially supported by Grant-in-Aid for Exploratory Research (No. 24650116) from JSPS.

Table 2: The number of times that execute a solution search.

instance	2-opt	3-opt	proposed	proposed+CNN
Lipa20a	2,129	9,490	9,764	14,260
Lipa20b	2,289	9,319	10,887	15,358
Lipa30a	7,595	35,261	33,683	48,229
Lipa30b	7,656	35,879	37,107	52,941
Lipa50a	35,341	282,685	183,550	257,467
Lipa50b	37,791	491,817	205,159	269,214
Tai60a	65,366	600,561	364,210	477,880
Tai60b	129,015	1,098,410	462,659	576,948
Tai100a	318,186	4,766,530	1,351,850	1,623,410
Tai100b	673,497	9,816,840	1,697,860	1,983,210

References

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

[2] S. Lin and B. Kernighan: "An effective heuristic algorithm for the traveling salesman problem," Operational Research, Vol.21 pp.498–516, 1973.

[3] M. Hasegawa, T. Ikeguchi and K. Aihara: "Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems," Physical Review Letters, Vol.79, No.12, pp.2344–2347, 1997.

[4] T. Kimura and T. Ikeguchi, "A new algorithm for packet routing problems using chaotic neurodynamics and its surrogate analysis," Neural Computing and Applications, Vol16, No.6, pp.519-526, 2007.

[5] T. Matsuura and T. Ikeguchi, "Chaotic motif sampler: detecting motifs from biological sequences by using chaotic neurodynamics," Nonlinear Theory and Its Applications, Vol 1, No.1, pp.207-220, 2010.

[6] S. Motohashi, T. Matsuura, T. Ikeguchi, and K. Aihara: "The Lin-Kernighan algorithm driven by chaotic neurodynamics for large scale traveling salesman problems," Lecture Notes in Computer Science, Vol.5769, pp.563-572, 2009.

[7] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," Physics Letters A, Vol.144, No.6, pp.333–340, 1990.