2014 International Symposium on Nonlinear Theory and its Applications
NOLTA2014, Luzern, Switzerland, September 14-18, 2014

# Performance analysis of adaptive tabu search for quadratic assignment problems

Akio Watanabe†, Yutaka Shimada‡, Kantaro Fujiwara‡, Takaomi Shigehara† and Tohru Ikeguchi‡

†Graduate School of Science and Engineering, Saitama University
255 Shimo-Ohkubo, Sakura-ku, Saitama-city, Saitama, 338-8570 Japan
‡Department of Management Science, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku, Tokyo, 162-8601 Japan

**Abstract**—The quadratic assignment problem is one of the NP-hard combinatorial optimization problems. In this paper, we analyzed solvable performance of an adaptive local search algorithm for solving quadratic assignment problems which has similar searching characteristic to the Lin-Kernighan algorithm for solving traveling salesman problems. We also analyzed the proposed adaptive local search algorithm with tabu search dynamics. Using results obtained from the analysis, we proposed a new adaptive local search algorithm with less variable depth. Evaluating the performances of the proposed algorithms, we found that the new adaptive local search algorithm has high performance with less calculation costs by modified for tabu search.

## 1. Introduction

In our life, we are often asked to solve combinatorial optimization problems such as drilling problems, VLSI design, scheduling, and delivery plan problems. When we solve these combinatorial optimization problems, one of the most important issues is to reduce costs. If we can obtain the best solutions, we can minimize the costs. However, we cannot usually get a best solution because these combinatorial optimization problems belong to a class of NP-hard. Thus, it is necessary to develop effective heuristic algorithms, even though obtained solutions are not guaranteed to be best.

The quadratic assignment problem (QAP)[1] is one of the NP-hard problems. The goal of QAP is how to find a best assignment. It formulates various real life problems. For example, in case of facility assignments, the goal is how to assign facilities to each city with a minimum total cost when the flow between the facilities and distance between the cities are given.

On the other hand, several heuristic algorithms for solving traveling salesman problem (TSP), which is a special case of QAP, have already been proposed, such as the 2-opt, the Or-opt and the Lin-Kernighan (LK) algorithms[2]. Among them, the LK algorithm is one of the most powerful algorithms for finding superior solutions of TSP. The LK algorithm controls the number of exchanged links $\lambda$ in the $\lambda$-opt algorithm. It realizes an effective search by changing $\lambda$ adaptively during its searching processes.

We have already proposed an algorithm for solving QAPs that has a similar searching strategy as the LK algorithm: namely the number of exchanged elements are adaptively decided[3]. The proposed adaptive local search algorithm has higher performance. We also introduced tabu search strategy[6, 7] to escape from undesirable local minima, because the proposed adaptive local search algorithm for QAPs has a steepest descent down hill dynamics, which has a drawback of being trapped at undesirable local minima.

In this paper, we modified the adaptive local search algorithm with tabu search[3] to obtain much higher performance. We conducted numerical experiments to show that the performance of the modified adaptive local search algorithm is superior using the benchmark problems of QAPLIB[1].

## 2. QAP

The QAP[1] is a typical example of an NP-hard combinatorial optimization problem. It includes various real life combinatorial optimization problems. For example, TSP is a special case of QAP.

In QAP, when two $n \times n$ matrices, a distance matrix $\boldsymbol{A} = (a_{ij})$ and a flow matrix $\boldsymbol{B} = (b_{ij})$, are given, we are asked to obtain an assignment $\boldsymbol{p} = \{p(1), p(2), \ldots, p(n)\}$ that minimizes an objective function where, distance $a_{ij}$ means the distance between cities $i$ and $j$, the flow $b_{kl}$ means the flow from the facility $k$ to the facility $l$. The objective function of QAP is defined as follows

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

Although it is easy to define QAPs, it is almost impossible to get an optimal solution of QAPs. The reason is that the number of possible assignments becomes $n!$ for an n-size QAP, then the number of feasible solutions increases exponentially if we use an exhaustive search. Then, it is important to develop a good heuristic algorithm in a reasonable time frame, even though obtained solutions are not guaranteed to be optimal.

## 3. $\lambda$-exchange algorithm

The $\lambda$-exchange algorithm is one of the most typical heuristic algorithms for searching better solutions of QAP. It is described as follows.

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

**Step2** $\lambda$ facilities are chosen from $\boldsymbol{p}$, and their assignments 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 $\lambda$-exchange algorithm with large $\lambda$ can find superior solutions because the size of neighborhood increases, when $\lambda$ becomes large. However, if $\lambda$ becomes large, calculation costs also increase exponentially. It means that it is almost impossible to finish a search in realistic time.

## 4. An adaptive $\lambda$-exchange algorithm with tabu search

To overcome a computational complexity of the $\lambda$-exchange algorithm, we have already proposed an adaptive $\lambda$-exchange algorithm[3]. In the proposed algorithm[3], solutions are searched by exchanging elements with changing $\lambda$ adaptively. It is implemented by deciding the number of exchanged elements adaptively, which is a similar strategy to the LK algorithm[2] applied to TSP.

### 4.1 Local search algorithm[3]

First, let us explain the adaptive local search algorithm[4, 5] in the following.

**Step1** Facilities are assigned to cities randomly. Let us describe the present solution as $\boldsymbol{p}$, and the best solution in the past search as $\boldsymbol{p}_{\text{best}}$.

**Step2** A new iteration starts. The cities $s_1$ and $s_2$ are chosen from the cities $c_i(i = 1, 2, ..., n)$ which reduce the cost most when the assigned facilities are exchanged each other. The city $s_1$ is a start point of the exchange. Then, a new solution $\boldsymbol{p}'$ by exchanging the facilities assigned to $s_1$ and $s_2$ is obtained. If $F(\boldsymbol{p}_{\text{best}}) > F(\boldsymbol{p}')$, $\boldsymbol{p}_{\text{best}}$ is updated to $\boldsymbol{p}'$. Let $d = 2$. Here, $d$ is the number of cities which are already exchanged in this iteration.

**Step3** If $d = L$, go to **Step5**. Here, $L$ is a predefined parameter for the search depth. Then, the city $s_{d+1}$ is chosen from the cities $c_i(i = 1, 2, ..., n)$ that has not been exchanged yet ($\{c_i\} \setminus \{s_k\}(k = 1, 2, ..., d)$), which reduces the cost most when the assigned facility is exchanged with $s_1$. Then, the solution $\boldsymbol{p}'$ is updated by exchanging the facilities which assigned $s_1$ and $s_{d+1}$. Then, $d$ is increased by 1.

**Step4** If $F(\boldsymbol{p}_{\text{best}}) > F(\boldsymbol{p}')$, $\boldsymbol{p}_{\text{best}}$ is updated to $\boldsymbol{p}'$. Otherwise, $\boldsymbol{p}_{\text{best}}$ is not updated. Return to **Step3**.

**Step5** The present solution $\boldsymbol{p}$ is updated to $\boldsymbol{p}_{\text{best}}$. If $\boldsymbol{p}$ is not improved at this iteration, the search is terminated, otherwise, return to **Step2**.

In **Step3** of the proposed local search algorithm, we set the parameter $L = n$. It means that we search solutions from depth 2 to depth $n$ adaptively in all iteration.

### 4.2 Introduction of tabu search[6, 7]

The main ideas of the tabu search is described as follows among all the neighboring solutions, the tabu search moves the proposed solution to a best solution. If there are no improving moves, the tabu search chooses a move that least degrades the objective function. To avoid returning to the local optimum just visited, the reverse move is now forbidden. This is realized by storing this move in a data structure often called a tabu list.

We explain the adaptive local search algorithm with the tabu search[3] as follows.

**Step1** Facilities are assigned to cities randomly. Let us describe the present solution as $\boldsymbol{p}$, and the best solution in the past search as $\boldsymbol{p}_{\text{best}}$. Then, the tabu list is initialized. A size of the tabu list is set $T$. The tabu list is utilized to restrict only the choice of the cities in **Step2**.

**Step2** A new iteration starts. Let us describe the new best solution in this iteration as $\boldsymbol{p}_{\text{ite}}$. The cities $s_1$ and $s_2$ are chosen from the cities $c_i(i = 1, 2, ..., n)$ that are not stored in the tabu list which reduce the cost most when the assigned facilities are exchanged each other. Then, the cities $s_1$ and $s_2$ are stored in the tabu list. The city $s_1$ is a start point of the exchange. Then, a new solution $\boldsymbol{p}'$ by exchanging the facilities which are assigned to $s_1$ and $s_2$ is obtained. Then, $\boldsymbol{p}_{\text{ite}}$ is updated to $\boldsymbol{p}'$. Let $d = 2$. Here, $d$ is the number of cities which are already exchanged for the assignment of facilities in this iteration.

**Step3** If $d = L$, go to **Step5**. Here, $L$ is the parameter for the limit of depth of solution search. Then, the city $s_{d+1}$ is chosen from the cities $c_i(i = 1, 2, ..., n)$ except cities that has already been exchanged $s_k(k = 1, 2, ..., d)$ which reduces the cost most when the assigned facility is exchanged with $s_1$. Then, the solution $\boldsymbol{p}'$ is updated by exchanging the facilities which are assigned to $s_1$ and $s_{d+1}$. Then, $d$ is increased by 1.

**Step4** If $F(\boldsymbol{p}_{\text{ite}}) > F(\boldsymbol{p}')$, $\boldsymbol{p}_{\text{ite}}$ is updated to $\boldsymbol{p}'$. Otherwise, $\boldsymbol{p}_{\text{ite}}$ is not updated. Return to **Step3**.

**Step5** If $F(\boldsymbol{p}_{\text{best}}) > F(\boldsymbol{p}_{\text{ite}})$, $\boldsymbol{p}_{\text{best}}$ is updated to $\boldsymbol{p}_{\text{ite}}$. Next, $\boldsymbol{p}$ is updated to $\boldsymbol{p}_{\text{ite}}$. Return to **Step2**.

### 4.3 Modification of tabu search

In **Step3** of section 4.2, we set the parameter $L = n$. However, if we introduce the tabu search, the value of $L$ = 2, often appropriate.

Then, we do not fix the value of $L$. Namely, we varied the value of $L$ during the search. We used the following definition:

$$L = 1 + \left\lfloor \frac{M}{T} \right\rfloor, \tag{2}$$

where $M$ is the maximum number of calculating the objective functions during the search and $T$ is the number of calculating objective functions from the initial state to the present state. In this paper, we set $M = 1 \times 10^6$ and $T$ increases form 0 to $M$ during the search. If $L$ is lager than $n$, we set $L = n$. Then, in the first phase of the searching process, the value of $L$ becomes $n$. At the final phase of the searching process, the parameter $L$ converges to 2 because the value of $\dfrac{M}{T}$ converges to 1.

## 5. Numerical results

### 5.1 Results of adaptive local search

We evaluated the performance of the adaptive local search algorithm using benchmark problems from QAPLIB[1]. For each instance, different 100 initial solutions were prepared. We evaluated the performance of two local search algorithms: the 2-exchange and the adaptive local search algorithm. To evaluate the performance, we used the gap defined by the following equation:

$$\mathrm{gap}[\%] = \frac{\text{found best solution} - \text{optimal solution}}{\text{optimal solution}} \times 100. \quad (3)$$

Table 1 shows the average gaps and the number of calculating the objective functions until the search is stopped at local minima. For all instances, the proposed local search algorithm can find better solutions than the 2-exchange algorithm. Moreover, the numbers of calculating the objective function of the adaptive local search algorithm are almost the same as the 2-exchange algorithm. The results indicate that the adaptive local search[3] algorithms could find better solutions efficiently.

Table 1: Results of average gaps [%] and the number of calculating the objective functions during the search ($N$).

| | 2-exchange | | adaptive local search | |
|---|---|---|---|---|
| instance | gap[%] | $N$ | gap[%] | $N$ |
| Lipa40a | 1.51 | 18,002 | 1.45 | 16,290 |
| Lipa60a | 1.11 | 62,711 | 1.04 | 45,531 |
| Lipa80a | 0.85 | 150,321 | 0.79 | 98,545 |
| Lipa40b | 18.5 | 18,954 | 18.0 | 17,096 |
| Lipa60b | 20.1 | 63,932 | 19.7 | 46,262 |
| Lipa80b | 21.7 | 156,357 | 21.3 | 102,415 |
| Sko49 | 3.09 | 53,543 | 2.48 | 52,070 |
| Sko56 | 2.84 | 84,685 | 2.20 | 81,312 |
| Sko64 | 2.69 | 129,951 | 2.05 | 120,102 |
| Sko81 | 2.30 | 297,594 | 1.70 | 247,296 |
| Sko90 | 2.15 | 426,973 | 1.63 | 351,138 |
| Tai40a | 5.05 | 18,392 | 4.51 | 16,655 |
| Tai60a | 4.72 | 63,649 | 4.21 | 47,759 |
| Tai80a | 3.67 | 160,781 | 3.27 | 100,605 |
| Tai100a | 3.46 | 312,196 | 3.01 | 178,868 |
| Tai40b | 10.4 | 33,524 | 9.10 | 33,781 |
| Tai60b | 8.66 | 127,776 | 6.00 | 121,278 |
| Tai80b | 6.01 | 318,749 | 4.53 | 283,466 |
| Tai100b | 5.28 | 676,071 | 4.01 | 578,063 |

### 5.2 Results of adaptive local search with tabu search

We evaluated the performance of three algorithms with tabu search: (i) the 2-exchange algorithm with the tabu search, (ii) the adaptive local search algorithm [4, 5] with the tabu search and constant $L$ (= $n$) and (iii) the adaptive local search algorithm with the tabu search and variable $L$. The 2-exchange algorithm with the tabu search corresponds to the adaptive local search with constant $L = 2$.

Figure 1 shows temporal change of gaps during the search. We firstly obtain the solutions by the proposed adaptive local search algorithm without the tabu search until improvement stops. Next, the solution is improved by the adaptive local search algorithm with the tabu search until the searches are executed for $1 \times 10^6$. For each instance, different 30 initial conditions were prepared. Figure 1 is the result of the average of 30 trials.

In Tai** instances (Fig.1 (f),(g),(h),(i)), the adaptive local search algorithm with constant $L$ found superior solutions to the 2-exchange algorithm. On the other hand, in Lipa** instances (Fig.1 (a),(b)) and Sko** instances (Fig.1 (c),(d)), the 2-exchange algorithm found superior solutions to the adaptive local search algorithm with constant $L$.

In Tai** instances (Fig.1 (f),(g),(h),(i)), the adaptive local search algorithm with variable $L$ shows almost the same as the performance algorithm with constant $L$. In addition, in Lipa** instances (Fig.1 (a),(b)) and Sko** instances (Fig.1 (c),(d)), the algorithm with variable $L$ has the same performance as the 2-exchange algorithm. These results indicate that the algorithm with variable $L$ can find better solutions effectively for all instances.

## 6. Conclusion

We proposed an adaptive local search algorithm for solving QAPs. The proposed adaptive local search algorithm decides the number of exchanged elements adaptively. In addition, the algorithm could be improved by the tabu search. In this paper, we modified the proposed adaptive local search algorithm by introducing strategy of variable depth. Then, the modified algorithm could search solutions more effectively. Quantitative analysis by numerical experiments show that the proposed algorithm has higher performance for solving QAPs.

The authors would like to thank Mr. S. Ogawa and AGS Corp. for their kind encouragements on this research. 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.

## References

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

[2] S. Lin and B. Kernighan: Operational Research, Vol.21 pp.498–516, 1973.

[3] A. Watanabe, K. Kuroda, K. Fujiwara and T. Ikeguchi: IEICE Technical Report, Vol. 113, pp. 55–60, 2014.

[4] A. Watanabe, K. Kuroda, K. Fujiwara and T. Ikeguchi: NOLTA2013, pp. 362–365, 2013.

[5] A. Watanabe, K. Kuroda, K. Fujiwara and T. Ikeguchi: IEICE Society Conference, AS-1-2, 2013.

[6] F. Glover: ORSA Journal on Computing, Vol.1, pp. 90–206, 1989.

[7] F. Glover: ORSA Journal on Computing, Vol.2, pp. 4–32, 1990.

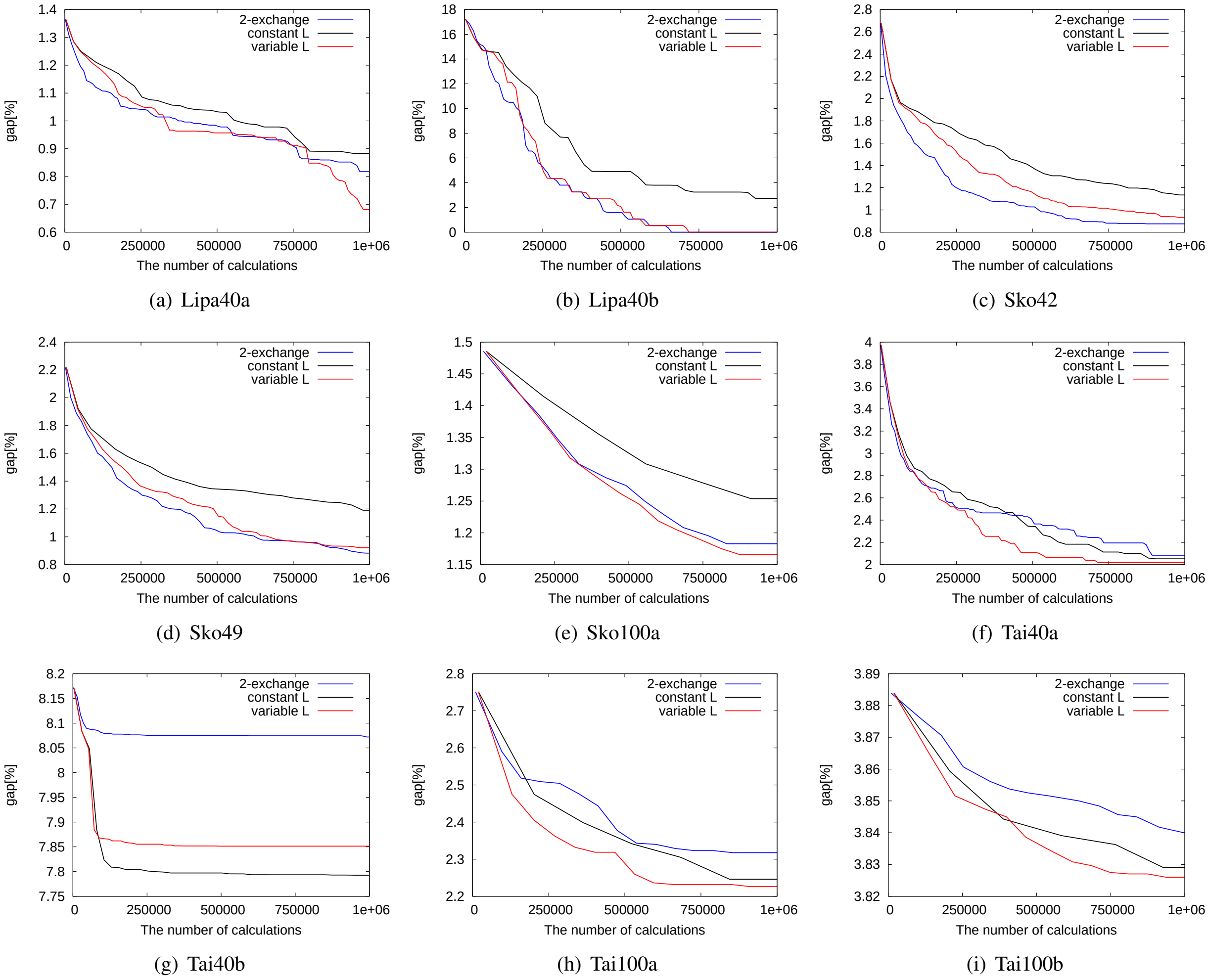


Figure 1: Temporal change of gaps. Abscissae show the number of calculating the objective functions during the search. Results by the 2-exchange algorithm with tabu search are shown by blue lines, those by the adaptive local search algorithm with constant $L$ are shown by black lines and those by the adaptive local search algorithm with variable $L$ are shown by red lines.