A Proposal of Genetic Algorithm for TSP with a Constraint on Visiting Orders

Hiroaki Tohyama†, Kenichi Ida†, and Naoko Abe†

†Department of systems and Information Engineering,
Graduate School of Engineering,
Maebashi Institute of Technology,
460-1, Kamisadori, Maebashi, Gunma, Japan
Email: tohyama@maebashi-it.ac.jp

***Abstract*— In this paper, we propose a solution method using genetic algorithm for traveling salesman problem with visiting orders. Since the order visiting cities has to be considered, we define the good character of individual as the order visiting cities. Concretely, the order visiting each city in individuals which are parents is inherited to children in the proposed crossover. The effectiveness of our method was verified by numerical experiments.**

I. Introduction

Traveling salesman problem (TSP) is a typical combinatorial optimization problem that is known to be **NP**-hard. Therefore, various approximation algorithms have been studied. Among such algorithms genetic algorithm (GA) has been applied to TSP well since the use is easy and a solution with comparatively high accuracy is obtained for a short time. TSP is the problem which finds a minimum tour in which all of given cities are visited. However, in many cases another constraint is added to TSP. For example, there is a case that the appointed time for some specific cities has been decided beforehand. TSP with such constraint is called Traveling Salesman Problem with Time Windows (TSPTW), and recently there are many studies on TSPTW[1][2][3][5][6]. In this paper, we treat TSP with a constraint on the order visiting cities. That is, it has been decided beforehand that some specific cities must be visited before other cities, respectively. Such TSP will be called Traveling Salesman Problem with Visiting Orders (TSPVO). To think such problem is needed in the situation that the result of a meeting held at a city has to be reported in a meeting held at another city. Many of the methods of GA for TSP which have been known define the good character as the edges which construct the tour, and in the crossover edges that parents possess are inherited to the children. The first crossover that inherits edges was proposed by Grefenstette[4]. Whitley[10] proposed a crossover (EX) that improved on Grefenstette's crossover. EX increases the inheritance rate of edges than the Grefenstette's one. SXX[12] and EXX[7] are the crossover in which the edges of parents are inherited completely. Although it is guaranteed that in SXX and EXX, the length of the generated tours is much the same as the length of the parent tours, they can not keep up diversity of the population. EAX[8] is developed by considering the balance between the inheritance rate of edges and the variation of generated individuals. If we pursue only the length of the tour, then the edges which construct the tour should be defines as the good character. In TSPVO, however, to inherit the order visiting cities is more natural than to inherit edges which construct the tour. In this paper, we define the good character of individual as the order visiting cities, and the good character is used as the guide which designs our solution method. Although generated individuals may have some crossing edges, this fact will be of helpful for keeping up diversity of the population. The solution candidates with crossing edges will be modified by applying the 2-opt method [7]. It cannot be said that our crossover is enough from the point of the local search since solution space is searched widely. To make up for this point, we propose a mutation which works like local search. That is, some cities are selected randomly, and if the tour is improved by changing the locations of selected cities, then the original tour is updated. Our method can obtain a good solution by balancing between crossover and mutation. This fact will be verified by the numerical experiment using some instances.

II. Traveling Salesman Problem with Visiting Orders

Let $G = (V, E)$ be a complete graph with n vertices, and $d : E \rightarrow \mathbf{Z}^+$ be a distance function which satisfies the following condition:

$$\forall c,\ c' \in V,\ d(c,\ c') = d(c',\ c).$$

A tour on G is a sequence $t : c_{i_1}, c_{i_2}, \cdots, c_{i_n}$ of cities which satisfies the following condition:

$$i_j \neq i_k,\ (1 \leq j,\ k \leq n,\ j \neq k).$$

The length l of the tour t is defined as follows,

$$l = \sum_{j=1}^{n-1} d(c_{i_j},\ c_{i_{j+1}}) + d(c_{i_n},\ c_{i_1}).$$

Let s be a vertex of G, and let $T \subseteq V^2$ be a set of pairs of vertices. Traveling salesman problem with visiting orders (TSPVO) is a problem to find a minimum tour $t : c_{i_1}, c_{i_2}, \cdots, c_{i_n}$ on G that satisfies the following conditions for given G, d, s, and T:

- $c_{i_1} = s$,

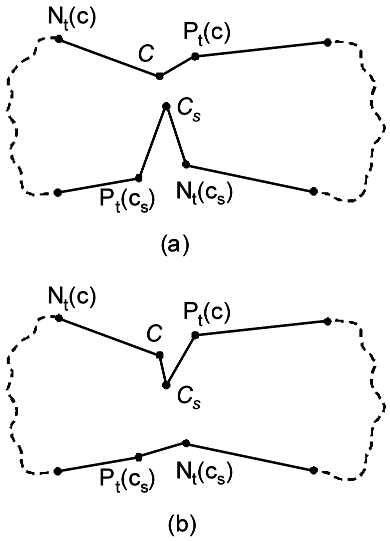

Fig. 1. An original tour (a) and the modified tour by mutation (b).

- for any $(c, c') \in T$, if $c = c_{i_j}$ and $c' = c_{i_k}$, then $j < k$,

The vertex of graph will be called city, s and the element of T will be called starting city and visiting order, respectively.

III. Proposed Method

In this paper, we propose a solution method using GA that solves TSPVO. In this section, a crossover which inherits the order visiting cities and a mutation are proposed.

A. Crossover

Let $t : c_{i_1}, c_{i_2}, \cdots, c_{i_n}$ be a tour, and $c_s (= c_{i_j})$ and $c_k (= c_{i_l})$ be cities. The visiting rank of city c_k for the city c_s on the tour t, which is denoted by $r_t(c_s, c_k)$, is defined as follows:

$$r_t(c_s,\ c_k) = \begin{cases} l - j, & \text{if } i_l > i_j, \\ n - j + l, & \text{otherewise.} \end{cases}$$

That is, the visiting rank $r_t(c_s,\ c_k)$ denotes the order visiting c_k when salesman starts from c_s on tour t. For example, let $V = \{c_1, c_2, \cdots, c_{10}\}$ be a set of cities, and let t : c_6, c_4, c_2, c_5, c_1, c_9, c_3, c_{10}, c_8, c_7 be a tour. In this case, $r_t(c_2, c_8) = 6$, and $r_t(c_3, c_4) = 5$. The procedure of our proposed crossover is as follows:

step 1 Select one city c_s from the set V, randomly.

step 2 For each city $c_i \in V$, the visiting ranks $r_{t_1}(c_s, c_i)$ and $r_{t_2}(c_s, c_i)$ are computed.

step 3 For each $i (1 \leq i \leq n)$, $R_{s,i}$ is computed as follows:

$$R_{s,i} = r_{t_1}(c_s,\ c_i) + r_{t_2}(c_s,\ c_i).$$

step 4 A tour t_c that is the sequence of cities sorted on the order which increases the value of $R_{s,i}$ is generated. That is,

$$t_c : c_{k_1},\ \ c_{k_2},\ \ \cdots,\ \ c_{k_n},$$

where $R_{s,k_1} \leq R_{s,k_2} \leq \cdots \leq R_{s,k_n}$.

step 5 For the generated tour t_c, 2-opt method is applied. The child is the tour t'_c that is generated by 2-opt method.

By the same procedure, one more child is generated from parents t_1 and t_2. For example, let t_1: c_3, c_5, c_4, c_2, c_1, c_6 and t_2: c_3, c_6, c_5, c_4, c_2, c_1 be a pair of parents, and let c_3 be the selected city. In this case,

$$R_{3,1} = r_{t_1}(c_3,\ c_1) + r_{t_2}(c_3,\ c_1) = 4 + 5 = 9,$$
$$R_{3,2} = r_{t_1}(c_3,\ c_2) + r_{t_2}(c_3,\ c_2) = 3 + 4 = 7,$$
$$R_{3,3} = r_{t_1}(c_3,\ c_3) + r_{t_2}(c_3,\ c_3) = 0 + 0 = 0,$$
$$R_{3,4} = r_{t_1}(c_3,\ c_4) + r_{t_2}(c_3,\ c_4) = 2 + 3 = 5,$$
$$R_{3,5} = r_{t_1}(c_3,\ c_5) + r_{t_2}(c_3,\ c_5) = 1 + 2 = 3,$$
$$R_{3,6} = r_{t_1}(c_3,\ c_6) + r_{t_2}(c_3,\ c_6) = 5 + 1 = 6.$$

A new tour $t_c : c_3, c_5, c_4, c_6, c_2, c_1$ is generated by sorting cities on the order which increases the value of $R_{3,l}$. For generated tour t_c, 2-opt method is applied, and the child t'_c: c_3, c_2, c_1, c_6, c_5, c_4 is generated.

B. Mutation

The idea of our mutation is as follows (Figure 1):

1) Select a city c_s randomly, and c_s is removed from tour t.
2) Insert c_s to immediately before (or after) the neighboring city of c_s.
3) If the length of the modified tour is shorter than the original tour, then the modified tour is substituted for the original one.

We enter into detail on our mutation. Let $V = \{c_1, c_2, \cdots, c_n\}$ be a set of cities, and let t : c_{i_1}, c_{i_2}, $\cdots$, c_{i_n} be a tour. On the tour t, the city which is visited immediately before city c is denoted by $P_t(c)$, and the city that is visited immediately after c is denoted by $N_t(c)$. Let m and $repN$ be positive integers, and let $N_m(c)$ be a set of the neighboring m cities for c. Our proposed mutation is the procedure in which the following is repeated $repN$ times:

step 1 Select one city $c_s \in V$ randomly.

step 2 For each $c \in N_m(c_s) - \{P_t(c_s)),\ c_s,\ N_t(c_s)\}$, the following are computed:

$$l_t(0, c) = d(P_t(c), c_s) + d(c_s, c),$$
$$l_t(1, c) = d(c, c_s) + d(c_s, N_t(c)).$$

step 3 For a pair of i and c that minimize $l_t(i, c)$,

1) if $i = 0$ and $l_t(0, c) + d(P_t(c_s), N_t(c_s)) < d(P_t(c_s), c_s) + d(c_s, N_t(c_s)) + d(P_t(c), c)$, then the tour t is modified as follows: go to c via c_s from $P_t(c)$, and go to $N_t(c_s)$ directly from $P_t(c_s)$.

2) if $i = 1$ and $l_t(1, c) + d(P_t(c_s), N_t(c_s)) < d(P_t(c_s), c_s) + d(c_s, N_t(c_s)) + d(c, N_t(c))$, then the tour t is modified as follows: go to $N_t(c)$ via c_s from c, and go to $N_t(c_s)$ directly from $P_t(c_s)$.

TABLE I

THE RESULTS OF EXPERIMENTATION 1 (* IS THE LENGTH OF THE OPTIMAL SOLUTION)

Instance		Compared Method		
		MGA	NGA	pGA
eil101	Best	629	629	629
	Worst	630	629	629
629*	Ave.	629.02	629.00	629.00
	Freq.	98	100	100
kroA200	Best	29368	29368	29368
	Worst	29445	29368	29384
29368*	Ave.	29375.29	29368.00	29369.23
	Freq.	72	100	89

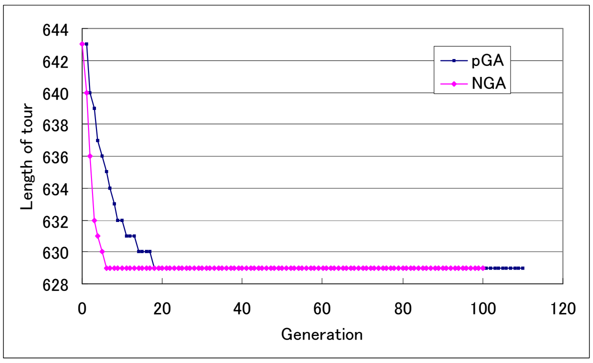

Fig. 2. The average length of the best tour in population of each generation on eil101.

C. proposed method

Our proposed method is as follows:

Initial Population: $popN$ individuals (tours) are generated randomly. The 2-opt method is applied for all generated individuals.

Selection of Parents: $popN/2$ individual pairs are selected randomly.

Generation of Children: For each pair of parents, two children are generated. That is, the probability of the crossover equals 1. For all generated children, 2-opt method is applied. Moreover, mutation is applied for each generated tour.

Existence Selection: Individuals which are left to the next generation are selected by elite selection from all parents and children.

IV. PERFORMANCE EVALUATION

In this section, we show the results of two experiments to evaluate the performance of our method.

Experimentation 1: We experimented on the instances with the empty set T. That is, we evaluated the performance of our method for the simple TSP without visiting orders. In this experimentation, we compared the proposed method with existing methods by using two benchmarks eil101 and kroA200 [9]. We adopt Maekawa's method[7] and Nagata's method[8] as the subjects for comparison, and these comparison methods are called MGA and NGA, respectively. The proposed method are called pGA below.

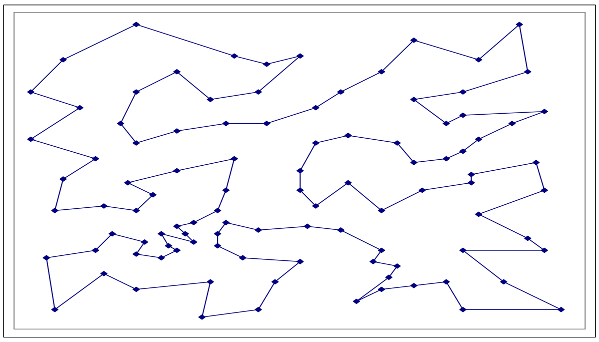

Fig. 3. The optimal tour of eil101 obtained by pGA.

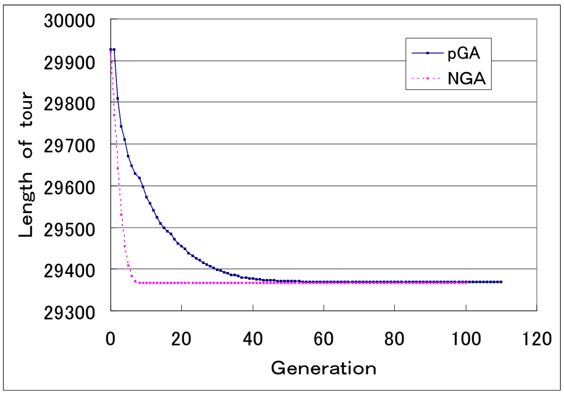

Fig. 4. The average length of the best tour in population of each generation on KroA200.

Experimentation 2: We evaluated the performance of our method by using eil101 to which visiting orders are added. The proposed method was compared with NGA. We experimented on eil101 to which five and eight visiting orders are added. Population size were set to 100 and 200 if instances are eil101 and kroA200, respectively. Population size is denoted by $popN$ below. The number of generations was set to 300. In the proposed method, the probability of crossover, the parameter m, and $repN$ are set to 1.0, 20, and $popN$, respectively.

A. The results of experimentation 1

In table I, the results of experimentation 1 are shown. The value of $\min(\{b_i\})$, $\max(\{b_i\})$, $(\sum b_i/100)$, and the frequency of the optimal tours (that is, the number of times that the optimal tour is obtained out of 100 trials) are shown, where b_i is the length of the best tour obtained in the ith trial. On the method MGA the results in [7] are quoted. The average lengths of the

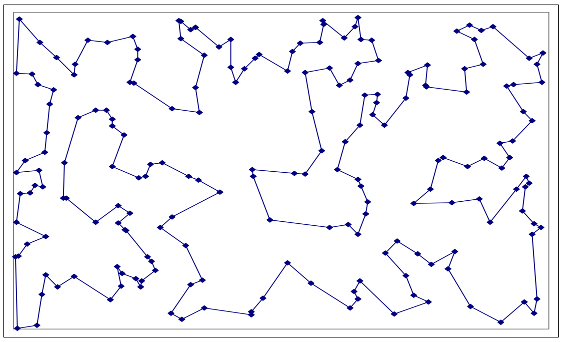

Fig. 5. The optimal tour of KroA200 obtained by pGA.

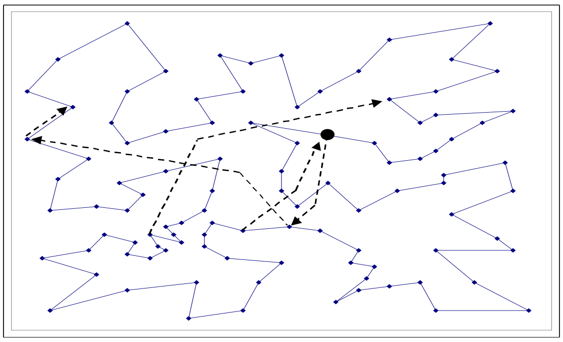

Fig. 6. The optimal tour of eil101(T5) obtained by pGA. The large vertex and dotted arrows denote the starting city s and visiting orders, respectively.

best tour in the population of each generation on 100 trials of eil101 and kroA200 are shown in figure 2 and 4, respectively. The frequency of the optimal solutions which could be obtained by pGA is more than the frequency for MGA's. On the values of maximum and average, the close values to the optimal solution could be obtained. For kroA200, the accuracy of pGA could not be better than the accuracy of NGA. In figure 3 and 5, the optimal tours for eil101 and kroA200 obtained by pGA are shown, respectively.

B. *The results of experimentation 2*

In table II, the results of experimentation 2 are shown. T5 and T8 are sets of five and eight visiting orders, respectively. Eil101 to which a set T of visiting orders is added is denoted by eil101(T). Freq.♯1, Best, Worst, Ave., and Freq.♯2 denote the frequency of feasible solutions obtained out of 30 trials, the minimum value, the maximum value, the average, and the frequency of the best tours in the obtained feasible solutions, respectively. In figure 6, the best solution of eil101(T5) obtained by pGA is shown. By dotted arrows visiting orders are denoted. That is, for each arrow, the head city has to be visited after the tail city when s is the starting city. On both eil101(T5) and eil101(T8), pGA was able to obtain many feasible solutions compared with NGA. This is because although the tendency for NGA to violate the constraint on the order visiting cities has been seen since NGA possesses the property that the accuracy of solutions is pursued, pGA does not violate the constraint and the order visiting cities in pair is inherited in the crossover. Although pGA could not obtain the best tour, this result is because importance was attached to obeying the constraint on the order visiting cities. Since to obey the constraint on the order visiting cities is contrary to pursue the accuracy of solutions, these balances are important.

TABLE II

THE RESULTS OF EXPERIMENTATION 2 (* IS THE LENGTH OF THE BEST SOLUTION)

<table>
<tr><th colspan="2" rowspan="2">Instance</th><th colspan="2">Compared Method</th></tr>
<tr><th>NGA</th><th>pGA</th></tr>
<tr><td rowspan="5">eil101(T5)
640*</td><td>Freq. # 1</td><td>13</td><td>30</td></tr>
<tr><td>Best</td><td>640</td><td>640</td></tr>
<tr><td>Worst</td><td>648</td><td>662</td></tr>
<tr><td>Ave.</td><td>641.23</td><td>642.37</td></tr>
<tr><td>Freq. # 2</td><td>8</td><td>10</td></tr>
<tr><td rowspan="5">eil101(T8)
647*</td><td>Freq. # 1</td><td>4</td><td>23</td></tr>
<tr><td>Best</td><td>647</td><td>649</td></tr>
<tr><td>Worst</td><td>649</td><td>681</td></tr>
<tr><td>Ave.</td><td>647.50</td><td>667.61</td></tr>
<tr><td>Freq. # 2</td><td>3</td><td>0</td></tr>
</table>

V. CONCLUSION

For solving Traveling Salesman Problem with visiting orders efficiently, we proposed a method of GA on which the good character is defined as the order visiting cities, and the performance was evaluated. It was shown that for the simple TSP without the constraint on the order visiting cities, comparatively high accuracy could be obtained by proposed method. For TSP with the constraint on the order visiting cities, proposed method could obtain the feasible solutions stably, although the existing method could not obtain them stably. However, the proposed method could not obtain the sufficient accuracy. For the increase of accuracy, in the crossover, it is known that the balance of the inheritance of edges is important[8]. The reason why a high accuracy could not be obtained by proposed method is because the inheritance of edges which the parents possess was not considered. In order to raise the accuracy, an improvement in which the inheritance of edges is considered is necessary.

REFERENCES

[1] Cavo, R. W., A new heuristic for the traveling salesman problem with time windows, *Transportation Science*, vol.34, no.1, pp.113–124, 2000.

[2] Dumas, Y., Desrosiers, J., Gelinas, E., and Solomon, M. M., An optimal algorithm for the traveling salesman problem with time windows, *Operations Research*, vol.43, no.2, pp.367–371, 1995.

[3] Gendreau, M., Hertz, A., Laporte, G., and Stan, M., A generalized insertion heuristic for the traveling salesman problem with time windows, *Operations Research*, vol.43, no.3, pp.330–335, 1998.

[4] Grefenstette, J., Gopal, R., Rosmaita, B., and Gucht, D., Genetic Algorithms for the Traveling Salesman Problem, *Proceedings of the 1st International Conference on Genetic Algorithms*, pp.160-165, 1985.

[5] Kim, J. S., and Suh, B. K., An algorithm for the TSP with Time Windows and Lateness Cost, *Journal of industrial and Systems Engineering*, 22, pp.13–22, 1999.

[6] Mingozzi, A., Bianco, L., Ricciadelli, S., Dynamic Programming Strategies for the Traveling Salesman Problem with Time Window and Precedence Constraints, *Operations Research*, 45, pp.365–377, 1997

[7] Maekawa, K., Tamaki, H., Kita, H., and Nishikawa, Y., A Method for the Traveling Salesman Problem Based on the Genetic Algorithm, *T. SICE*, vol.31, no.5, pp.598-605, 1995 (in Japanese).

[8] Nagata, Y., and Kobayashi, S., The Proposal and Evaluation of a Crossover for Traveling Salesman Problems, *J. JSAI*, vol.14, no.5, pp.848-859, 1999 (in Japanese).

[9] TSPLIB95, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/index.html (1995).

[10] Whitley, D., Starkweather, T., and Fuquay, D., Scheduling Problems and Traveling Salesman:The Genetic Edge Recombination Operator, *Proc.3rd ICGA*, pp.133-140, 1989.

[11] Yagiura, M., and Ibaraki, T., On Genetic Crossover Operators for Sequencing Problems, *T. IEE Japan*, vol.114-C, no.6, pp.713-720, 1994 (in Japanese).

[12] Yamamura, M., Ono, T., and Kobayashi, S., Character-Preserving Genetic Algorithms for Traveling Salesman Problem, *J. JSAI*, vol.7, no.6, pp.1049-1059, 1992 (in Japanese).