A Comparison between Genetic Algorithm and *k*-opt local search method for the Vehicle Routing Problem

SangHeon Han*

Abstract

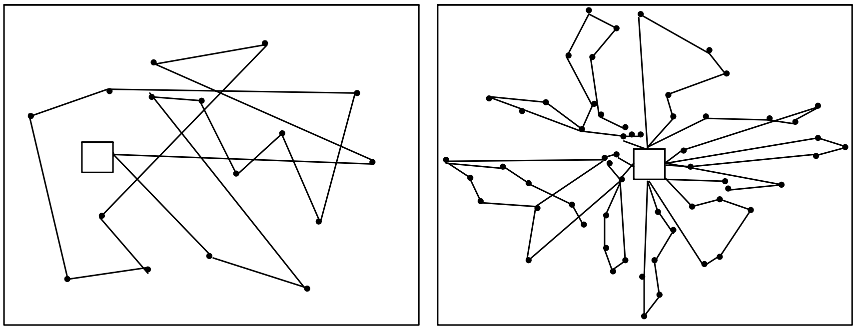
The Traveling Salesman problem (TSP) can be used to method many practical problems. The Vehicle Routing Problem (VRP) is more complicated than TSP because it requires determining which customers are assigned to each vehicles, as well as the optimal ordering of the cities within each vehicle's tour. Previous studies proposed that Genetic Algorithm, Integer Programming and several neural network approaches could be used to solve VRP. This paper compared the results for Genetic Algorithm (GA) as a Meta-Heuristic method and k-opt local search method as a heuristic method. We proposed three VRP-cases for simulations. Then, each case is solved with *k*-opt and GA in terms of performance and computing time.

1 Introduction

Besides being one of the most important problems of operations research in practical terms, the vehicle routing problem is also one of the most difficult problems to solve. It is quite close to one of the most famous combinatorial optimization problems, TSP, where only one person has to visit all the customers. The TSP is an NP-hard problem. It is believed that many never find a computational technique that will guarantee optimal solutions to larger instances for such problems. The vehicle routing problem is even more complicated. Even for small.eet sizes and a moderate number of transportation requests, the planning task is highly complex. Hence, it is not surprising that human planners soon get overwhelmed, and must turn to simple, local rules for vehicle routing. The TSP can be developed into VRP. The VRP was originally proposed by Dantzig and Ramser [2] and defined as follows: vehicles with a fixed capacity Q must deliver order quantities q_i ($i=1, ..., n$) of goods from a single depot ($i=0$) to n customers. Knowing the distance d_{ij} between customers i and j ($i, j=0, ..., n$), the objective of the problem is to minimize the total distance travelled by the vehicles in such a way that only one vehicle handles the deliveries for a given customer and the total quantity of goods that a single vehicle delivers is not larger than Q.

Figure 1 gives a graphical representation of a VRP and one possible solution. The square (in the middle of Fig 1(a) and (b)) represents the base (where the trucks start and.nish their tour) and the diamonds represent the sub-routes. Figure 1(b) shows the sub-tours of the different trucks. It should be

* Department of Management Administration Nagoya University of Commerce & Business 4-4 Sagamine, Komenoki-cho, Nisshin-shi, Aichi-ken, Japan 470-0193, Phone: +81-(0)561-73-2111, E-mail: hansangheon@gmail.com



(a) A Traveling Salesman Problem (b) A Vehicle Routing Problem

Fig 1. TSP & VRP

observed that, in this case, all the customers have been allocated.

Problem formulation

Let $G=(V, A)$ be a graph with a set V of vertices and a set A of arcs. We have $V=0\cup N$, where 0 corresponds to the depot and $N=1, \ldots, n$ is the set of customers. For the set of arcs, we have $A=(\{0\}\times N)\cup I\cup(N\times\{0\})$, where $I\subseteq N\times N$ is the set of arcs connecting the customers, $\{0\}\times N$ contains the arcs from the depot to the customers, and $N\times\{0\}$ contains the arcs from the customers to the depot. Every customer $i\in N$ has a positive demand q_i. For each arc $(i, j)\in A$ we have a cost c_{ij}. Furthermore, we assume that the vehicles are identical and have the capacity Q. All the above mentioned factors are assumed to be known in advance. Thus the model examined is deterministic.

We have the following variables: For each customer $i\in N$, y_i is the load of the vehicle when it arrives at the customer. Now the problem is to determine which of the arcs $(i, j)\in A$ are used by routes. For each arc $(i, j)\in A$, the decision variable x_{ij} is equal to 1 if arc (i, j) is used by a vehicle and 0 otherwise. Formally

$$\text{Minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

$$\text{Subject to} \quad \sum_{j\in V} x_{ij}=1 \qquad \forall i\in N \tag{2}$$

$$\sum_{j\in V} x_{ji}=1 \qquad \forall i\in N \tag{3}$$

$$x_{ij}=1\Rightarrow y_i-q_i=y_j \qquad \forall (i, j)\in I \tag{4}$$

$$q_i\leq y_j\leq Q \qquad \forall i\in V \tag{5}$$

$$x_{ij}\in\{0, 1\} \qquad \forall (i, j)\in A \tag{6}$$

We minimize the total costs that consist of travel costs and a.xed cost c of vehicles (included in the travel cost c_0 between depot and.rst customer). The object is, firstly minimize the number of routes or vehicles, and then the total distance of all routes. By equation (2), (3) and (6), we require that every customer be visited exactly once. Equation (4), (5) enforce that the loads of the vehicles when arriving at the customers are feasible. The purpose of this paper is to compare the solution of VRP using k-opt method and GA.

The organization of this paper is as follows. In section 2, the approaches and methodology are discussed in detail and then the results are presented and discussed in Section 3. Then the conclusion of this paper is described in section 4..

2 The Problem solving Methodology

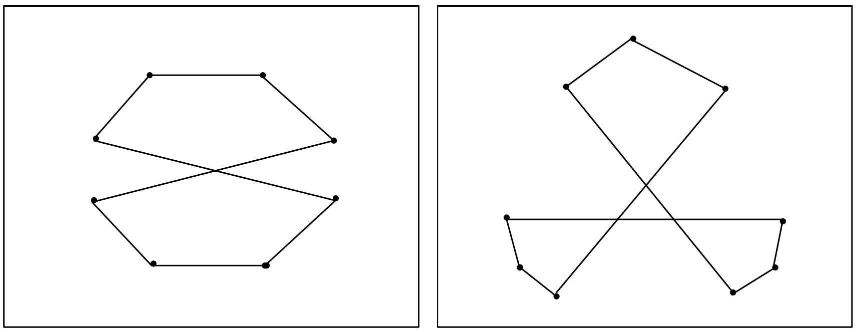
In this paper, VRP is solved using two approaches. The first approach is a local search with k-change neighborhoods. k-opt is the most widely used heuristic method for the traveling salesman problem. k-opt is a tour improvement algorithm, where in each step k links of the current tour are re-placed by k links in such a way that a shorter tour is achieved.

It has been shown Chandra [1] that k-opt may take an exponential number of iterations and that the ratio of the length of an optimal tour to the length of a tour constructed by k-opt can be arbitrarily large when $k \leq n/2-5$. Such undesirable cases, however, are very rare when solving practical instances [9]. Usually high-quality solutions are obtained in polynomial time. This is, for example the case for the Lin-Kernighan heuristic [4], one of the most effective methods for generating optimal or near-optimal solutions for the symmetric traveling salesman problem. High-quality solutions are often obtained, even though only a small part of the k-change neighborhood is searched.

In the original version of the heuristic, the allowable k-changes (or k-opt moves) are restricted to those that can be decomposed into a 2- or 3-change followed by a sequence of 2-changes. This restriction simplifies implementation, but it need not be the best design choice.

Once a tour has been generated by some tour construction heuristic, we might wish to improve that solution. There are several ways to do this, but the most common ones are the 2-opt and 3-opt local searches. Their performances are somewhat linked to the construction heuristic used. Other ways of improving our solution is to do a tabu search using 2-opt and 3-opt moves.

The second approach is to use Standard GA to solve VRP. GA is a metaheuristic search method based on population genetics. The basic concepts are developed by Holland (1975) [7], while the practically of using the GA to solve complex problems is demonstrated in Dejong (1975) [3] and Goldberg (1989) [5]. References and details about genetic algorithms can also be found for example in Alander



(a) 2-opt move (b) 3-opt move

Fig 2. k-opt move

(2000) [8] and Mühlenbein (1997) [6] respectively.

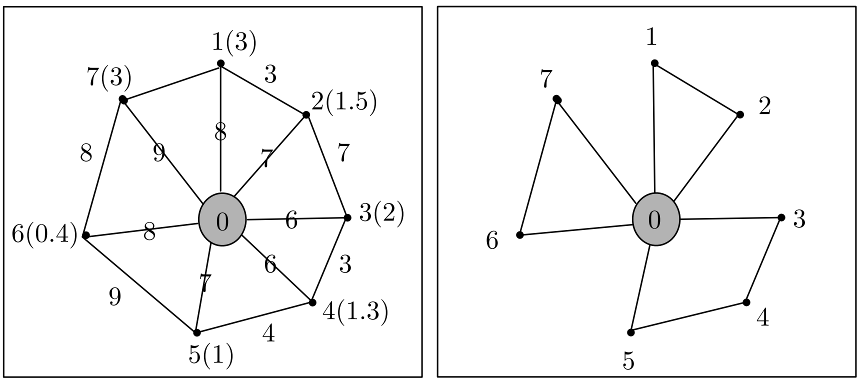
The creation of new generation of individuals involves primarily four major steps or phases: representation, selection, recombination (crossover), and mutation. The representation of the solution space consists of encoding significant features of a solution as a chromosome, defining an individual member of a population. Typically pictured by a bit string, a chromosome is made up of a sequence of genes, which capture the basic characteristics of a solution.

Although theoretical results that characterize the behavior of the GA have been obtained for bit-string chromosomes, not all problems lend themselves easily to this representation. This is the case, in particular, for sequencing problems, like the vehicle routing problem, where an integer representation is more often appropriate. We are aware of only one approach by Thangiah (1995) [11] that uses bit string representation in vehicle routing context.

A basic scheme of a typical algorithm is as follows:

Randomly create an initial population
While not (termination condition) **do**
  Evaluate each member's fitness
  Kill the bottom $x\%$ elements of the population
  Let the.tness reproduce themselves
  Randomly select two members/parents (many other selection methods are also used)
  Perform crossover on the selected elements to generate two children (many variations of crossover exist)
  Perform mutation
Endwhile

Like in other GAs applications, the members of a population in our GA for VRP are string entities of an artificial chromosome. The representation of the solution we present here is an integer string of



(a) delivery pattern (b) sub-routes in the solution
Fig 3. The example of VRP

If we have the following solution:

Route No. 1 is $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$

Route No. 2 is $0 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 0$

Route No. 3 is $0 \rightarrow 6 \rightarrow 7 \rightarrow 0$

The chromosome string of Fig 2(b) represent the solution as below:

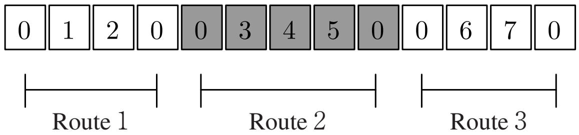


Fig 4. A chromosome representation

length N, where N is the number of customers including depots in question. Each gene in the string or chromosome, is the integer node number assigned to that customer originally. And the sequence of the genes in the chromosome is the order of visiting the customers.

In the fig 3., a number 0 indicates the delivery center (Depot), and the number written on each like corresponds to the distance between depot and customers and between customers. The numbers 1, 2, 3, 4, 5, 6, 7 correspond to customers. Moreover, the portion of the visited route is called a sub-route, for example route (0, 1, 2, 0) in the fig 2.(b) is a sub-route of all the feasible routes. Besides, the numbers in brackets correspond to the quantities required by each customer.

Note that we link the last customer visited in route i with the.rst customer visited in route $i+1$ to form one string of all the routes involved. Furthermore, we put any bit like 0 in the string to indicate the end of a route. To decode the chromosome into route configurations, we simply insert the gene values into routes sequentially (Fig 4.). In the above-mentioned expression, the length (the length of a sequence) of the chromosome becomes a variable length instead of a fixed length. The other GA operations (crossover & mutation) in VRP can be find in S.H.HAN [10].

3 Numerical Study and Results

Our numerical experiments were run on a Pentium Core2-Duo 3GHz Processor, Windows 7 Operating System using the Program Language C and Borland C++Builder. We tested two approaches; k-opt method and Genetic Algorithm so as to evaluate the performance treating 6-problems. We found that the speed of convergency is very sensitive to the setting of GA-parameters. However, the computational study on set of benchmark problems indicated that our GAbased meta-heuristic is capable of generating optimal solutions for small-size problems as well as high-quality solutions for large-size problems. The algorithm outperforms any of the previous heuristics in terms of solution quality. The computational times of the algorithm are very reasonable for all problem instances from the heuristic viewpoint. In addition, the numerical experiment used a delivery plan problem which is shown as an example (Table 1).

We performed the simulation 3 times for each 5 cases of the problem with random data. Each case is constructed as numerical data of the customers 50, 100, 150, 200 and 250. Figure 5. to Figure 9. show

Table 1. The example data of VRP (20 cities)

Node	abscissa (X)	ordinate (Y)	amount of delivery
1	86	63	1564
2	9	46	1036
3	17	47	1392
4	31	24	2726
5	100	68	3593
6	89	66	2988
7	70	85	3468
8	54	35	3902
9	67	23	1349
10	51	31	1201
11	47	28	1466
12	91	19	2101
13	31	93	3088
14	16	99	2845
15	65	81	1162
16	20	30	1342
17	60	14	3137
18	27	72	3460
19	20	36	2138
20	11	28	3667

the coordinates of the Depot：(50, 50)
the capacity of truck：5000

Table 2. Results of Simulations

the number of customers	k–opt			GA		
	types	distance	time (sec.)	run	distance	time (sec.)
50	2 -ops	2561. 32	2. 00	1000	2564. 74	17. 00
	3 -ops	2762. 43	1. 00	1500	2561. 53	25. 00
	4 -ops	2348. 65	1. 00	2000	2600. 83	37. 00
mean		2557. 47	1. 33		2575. 70	26. 33
100	2 -ops	6277. 57	2. 00	1000	5891. 61	10. 00
	3 -ops	5680. 87	2. 00	1500	5811. 88	18. 00
	4 -ops	5723. 76	1. 00	2000	5740. 61	26. 00
mean		5894. 07	1. 67		5814. 70	18. 00
150	2 -ops	8759. 97	3. 00	1000	8778. 68	17. 00
	3 -ops	8528. 78	2. 00	1500	8532. 62	25. 00
	4 -ops	8839. 12	3. 00	2000	8492. 73	37. 00
mean		8709. 29	2. 67		8601. 34	26. 33
200	2 -ops	11573. 98	2. 00	1000	11696. 88	24. 00
	3 -ops	11732. 27	4. 00	1500	11660. 40	45. 00
	4 -ops	11683. 38	3. 00	2000	11539. 49	73. 00
mean		11663. 21	3. 00		11632. 26	47. 33
250	2 -ops	15191. 26	4. 00	1000	15243. 75	32. 00
	3 -ops	15582. 31	3. 00	1500	15413. 07	51. 00
	4 -ops	15403. 43	2. 00	2000	15042. 89	69. 00
mean		15392. 33	3. 00		15233. 24	50. 67

vehicles' tour distance and sub-tours in each cases as developed by our program. Table 2. show the result that mean of execution time and the tour distance for each problem. Since GAs contain a random element theoretically, it cannot guarantee that the result of each time of the solution method using GAs is surely superior to the result of the other method.

Moreover, in the case of problems 2, 3, 4 and 5, in the above mentioned experiment, a result better than the k-opt method was searched more than twice in GAs execution. However, the case of the problem 1, we can find that k-opt method is better result than GA. It is considered to be a problem of the random search direction of GA to huge search space.

k-opt method is not only perform better than GA in case of small space but also the shorter computing time. On the other hand, we can.nd that GA is more efficient on the average for the big scale problem. In general, it is found that GA is superior to k-opt method in terms of performance (evaluated by fitness function) not considering computing time. k-opt method is much simpler to understand and speedy like a sweep algorithm.

4 Conclusions and Remarks

VRP is one of the classic problems associated with TSP, and it has been applied in various fields. Since, VRP is di.cult to solve in real time, heuristic methods have been adapted to solve it. In this paper, in order to compare k-opt method as a heuristic with GA as meta-heuristic, we tested each method as applied to NP-hard class optimization problems such as VRP. The influence on solutions, such as crossover and mutation, are very sensitive elements in GA.

Furthermore, we can claim that k-opt method is superior than GA in the case of small scale problems and GA is more e.ective to large scale problems even if it takes more times. GA takes much more computational time than other heuristics, but it is su.cient in real world. Moreover, our focus is not the speed of algorithm but the accuracy of the result. The two considered algorithms are complementary. When the performances of one of them is excellent, the performances of the other are poor, and vice versa. This suggests that better results, in general, could be obtained by exploiting the features of both exact and meta-heuristic algorithms. The study of hybrid strategies can lead to the development of new algorithms (which are in part exact and in part meta-heuristic), that could perform better than the two algorithms compared in this paper. In further studies, we are trying to investigate the performance for practical examples as well as to re.ne and improve the algorithm.

References

[1] B. Chandra, H. Karloff and C. Tovey 1994 New results on the old k-opt algorithm for the TSP, In Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.150–159.

[2] Dantzig G. B. and Ramser J. H. 1959. The Truck Dispatching Problem, *Management Science* 6(1), pp.80–91.

[3] Dejong K. A. 1975. Analysis of the behaviour of a classic of genetic adaptive system, ph.D. thesis, Department of Computer and Communication Sciences, University of Michigan.

[4] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys (eds.). 1985 *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, New York.

[5] Goldberg D. E. 1989. *Genetic Algorithm in Search, Optimization, and Machine Learning*. Addison-Wesley.

[6] Heinz Mühlenbein. 1997. *Genetic algorithms*. In *Local Search in Combinatorial Optimization*, John Wiley & Sons, Chichester, pp.137–172.

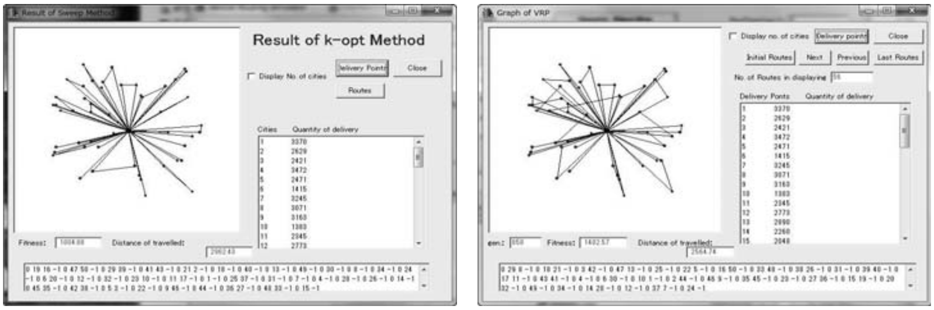
[7] Holland J. H. 1975. *Adaption in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

[8] Jarmo T. Alander. 2000. An indexed bibliography of genetic algorithms in operations research. Technical Report series No.94-1-OR, University of Vaasa, Vassa, Finland.

[9] M. Okada 1999, Studies on Probabilistic Analysis of λ-opt for Traveling Salesperson Problems, ph.D. thesis, Nara Institute of Science and Technology.

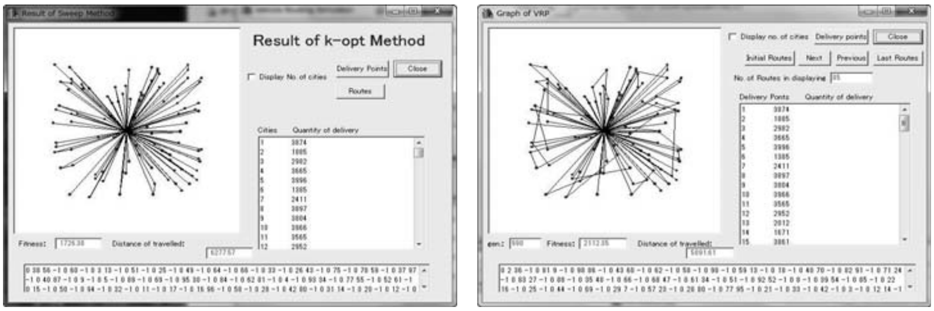
[10] S. Han & Y. Tabata, 2003. A Hybrid Genetic Algorithm for the Vehicle Routing Problem with controlling lethal gene, Joural of Asian Pacific Mangement Review, 7, 3, pp.405–425.

[11] Thangiah Sam. 1995. Vehicle routing with timewindows using genetic algorithms. In Application *Handbook of Genetic Algorithms*, New Frontiers, vol.2, pp.253–277, CRC Press, Boca Raton, 1995.



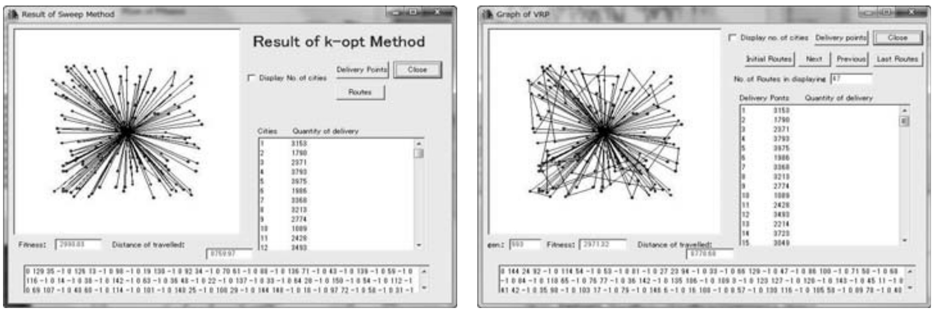
(a) *k*-opt (distance : 2557.47) (b) GA (distance : 2575.70)

Fig 5. Result of Problem (50 customers)



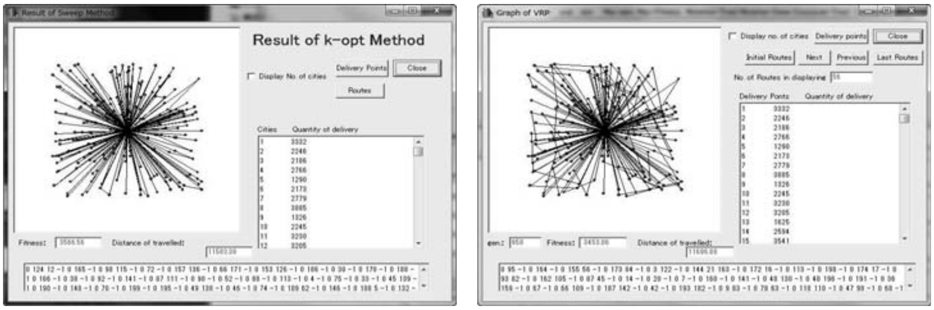
(a) *k*-opt (distance : 5894.07) (b) GA (distance : 5814.70)

Fig 6. Result of Problem (100 customers)



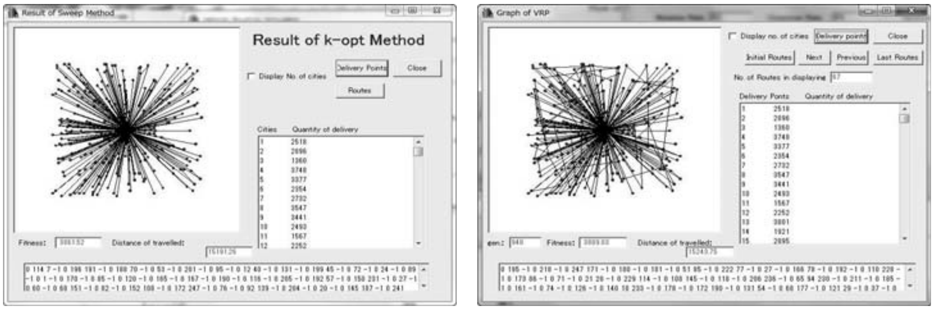
(a) *k*-opt (distance : 8709.29) (b) GA (distance : 8601.34)

Fig 7. Result of Problem (150 customers)



(a) *k*-opt (distance : 11663.21) (b) GA (distance : 11632.26)

Fig 8. Result of Problem (200 customers)



(a) *k*-opt (distance : 15392.33) (b) GA (distance : 15233.24)

Fig 9. Result of Problem (250 customers)