The Third AFSS(1998). 75 – 79

An Approximate Solution of Travelling Salesman Problem Using a Smoothing Method

Tomoyuki ARAKI and Fujio YAMAMOTO

Dept. of Information and Computer Sciences,
Kanagawa Institute of Technology
1030 Shimoogino, Atsugi-shi, Kanagawa 243-0292 Japan
Tel: +81-462-91-3164 Fax: +81-462-42-8490
E-mail: {araki, yamamoto}@ic.kanagawa-it.ac.jp

Abstract

It is well known that traveling salesman problem (for short, TSP) is one of most important problems for optimization, and almost all optimization problems result in TSP. This paper describes on an effective solution of TSP using genetic algorithm.

The features of our method are summarized as follows: (1) By using division and unification method, a large problem is replaced with some small ones. (2) Smoothing method proposed in this paper enables us to obtain a fine approximate solution globally. Accordingly, demerits caused by division and unification method are decreased. (3) Parallel operation is available because all divided problems are independent of each other.
Keywords: traveling salesman problem, genetic algorithm, smoothing method, parallelism

1. Introduction

Many studies on algorithm to solve traveling salesman problem (for short, TSP) effectively have been done for a long time. It is well known that TSP is one of the most fundamental problems for optimization and almost all optimization problems can be result in TSP.

This paper proposes on an approximate and effective solution of TSP based on genetic algorithm (for short, GA)[1]. The features of the proposed method are as follows:

(1) By using a division and unification method, large problems are divided into small ones. Therefore, scale of problems are reduced.
(2) The Division and unification method have harmful effects, that is, there is some possibility that solutions are led into local solutions. As a method to improve it and get fine solutions globally, we proposes a *smoothing method*.
(3) We can optimize each divided problem in parallel because each divided problem is independent with each other. Accordingly, the proposed method in this paper is suitable for applying to parallel computers.

There are two kinds of TSP. One is symmetrical TSP, and the other is asymmetric TSP. Generally, symmetric TSP is harder to solve than asymmetric TSP. This paper deals with symmetric TSP.

As recent results[5] of symmetric TSP, 325 cities' problem (Balas and Christofides, in 1981), 2392 cities' problem (padberg and Rinaldi, in 1992), 4461 cities' problem (Applegate, Bixby, Chvatal and Cool, in 1993) and 7397 cities' problem (Applegate, Bixy, Chvatal andCool, in 1994) were solved, respectively.

The outline of this paper is as follows:

At first, we describe a simple method based on GA. Here we propose a new concept, which is called *degree of similarity among chromosomes*. By using this concept when we select a pair of chromosomes for crossover, we can get descendants with diversity avoiding local solutions as much as we can.

Secondly, we describe a simple hybrid method based on GA and hill-climbing method (for short, HC). We apply HC both at the beginning of our GA described in the above and at the end of our GA. Hereafter, we call this method GA+HC. We verified that GA+HC was executed two times faster than our GA, and we could get optimal solution until about forty cities experimentally. However, we could get no solution for more than forty cities. Accordingly, the size of problem which we can get optimal solutions is about forty cities.

Thirdly, we try to divide a large problem into small ones using division method and then we apply GA+HC to divided small ones, and lastly we unify all optimized solutions. The result which we applied this method to the benchmark pr2392 (which is included in TSPLIB[6,7]) was about 15% longer than the optimal solution.

Lastly, we propose a new method, which is called a *smoothing method*. In the division and unification method described in the above, no optimization is applied among small divided problems. The smoothing method enables us to optimize relation of cities which are included in problems adjacent to each other. Concretely, after optimization in each divided regions, the smoothing method execute again GA+HC at boundaries between adjacent regions. Generally we can make the distance be shorter using the smoothing method. We obtained experimental results which included less than 6% for all benchmarks.

2. Proposal on selection-of-difference method

Hereafter, let integers greater then zero(0) represent genes, and they correspond to cities.

Using ordinal subtour exchange method, for crossover, if we select parents of next generation from chromosomes which have short distance of tour, the order of genes of all chromosomes is getting similar to each other gradually. Therefore, there is high possibility that solutions fall into local solutions. To get descendants with diversity avoiding local solutions as much as possible, we propose a new concept which is called *similarity among chromosomes*. And we propose a method to select parents of next generation using similarity among chromosomes. The method is called *selection-of-difference method.* When using selection-of-difference method, we select parents of next generation by obeying the following criteria:

(1) The shortest chromosomes are selected as the first parents.

(2) After the first generation, parents are selected from chromosomes which satisfy the following conditions:

(2.1) *Unlike* the parents of previous generation.

(2.2) the selected chromosomes have the shortest distance in unlike chromosomes.

Where *unlike* means that a chromosome does not have any subtour which have less than half the number of same genes for its parents.

[Example 1] In the case of eight cities, there is a parent and there are three children as follows:

parent: (0, 1, 2, 3, 4, 5, 6, 7)

child(1): (0, 1, 2, 3, 4, 7, 6, 5)
child(2): (1, 7, 5, 0, 6, 4, 3, 2)
child(3): (0, 1, 2, 7, 5, 6, 4, 3)

For each child(1) and child(2), there is at least one subtour which the number of same genes as parent is greater than four(=8/2). Therefore, child(1) and child(2) is not unlike. However, child(3) is unlike. //

By adopt the subtour exchange method using similarity among chromosomes, that is, selection-of-difference method, we can obtain descendants with diversity avoiding local solutions.

3. Estimation of selection-of-difference method

We estimate our selection-of-difference method. Here we use mutation with the above method.

Table 1 shows the estimation result for a 20 cities' problem.

Table 1 Convergence rate for the optical solution.

Crossover rate [%]	Convergence rate for the optimal solution[%] # of parents: 10	# of parents: 12
30	30.8	45.8
35	34.2	49.4
40	44.2	55.2
45	45.2	53.6
50	41.8	53.6
55	50.2	59.4
60	52.8	54.8
65	50.0	49.4

The best convergence rate is obtained for the case that the number of parents is 12 and probability for crossover is 55 %.

Table 2 shows execution time. Here we used SUN SPARCstation 5 (110MHz).

Table 2 Execution time for 20 cities and 30 cities.

# of Cities	Execution time [sec.]
20	14
30	65

Until 30 cities' problem, we can obtain solution in about one minute. However, we can not obtain any solution for problems over 40 cites' within several minutes. Consequently, GA that are mentioned above is not suitable for practical use because the scale of problem we can solve is too smaller.

3. Using Hill-Climbing method

In the previous section, it is shown that our GA can not solve large problems. In this section, a hybrid method is introduced. The method we adopt is hill-climbing method (for short, HC). HC is suitable for local search as compared with GA is suitable for global search. We apply HC method before GA and after GA (Hereafter, we notate this method as "GA+HC".). Estimation result of GA+HC is shown in Table 3.

Table 3 Execution time of GA+HC.

# of Cities	Congvergence rate for optimal solution[%]	Execution time[sec.]
20	100	7
40	97.9	14

In the case of 20 cities, convergence rate improves from 59.4% to 100%. And execution time improves from 14 sec. to 7 sec. In the case of 40 cities, we can obtain solutions within about 14 sec. although we can not obtain solution within several minutes using the previous method in section 2. However, for over 50 cities' problem, we can not obtain the optimal solution using GA+HC at all. Consequently, the limit of the size of problem we can solve using GA+HC is about 40 cites.

4. Using Division and unification method and proposal on smoothing method

According to the result in section 3, the limit of the size of problem we can solve using GA+HC is about 40 cities for practical use. In this section, using division and unification method (for short, DU), we try to obtain approximate solutions for larger than 40 cities' problem.

4.1 Division method

Fig. 1 illustrates division method. In this figure, 0,1,2,3 and 4 represent names of cities, respectively. If width of the region which cites exit is longer than its height, division is done vertically. Similarly, if height of the region is longer than its width, division is done horizontally.

4.2 Unification method

Fig. 2 illustrates unification method. When unification is done, either path(0-1) or path(3-4) is selected and unified.

(1) length of path(0-2-1) - length of path(0-1)
(2) length of path(3-2-4) - length of path(3-4)

In the case of Fig. 2, by comparing the above (1) with (2), (1) is selected because difference is longer than (2).

4.3 GA+HC with DU

The method that GA+HC is executed after division method, and after that, unification method is executed is notated as "GA+HC with DU". The size of divided region is smaller than or equal to 40 cities. Using this method, we can obtain solutions for large scale problems. However, divided regions are independent with each other. Furthermore, no optimization is done among divided regions. Therefore, the larger the size of problem, the more approximation error get growth (See Table 4.).

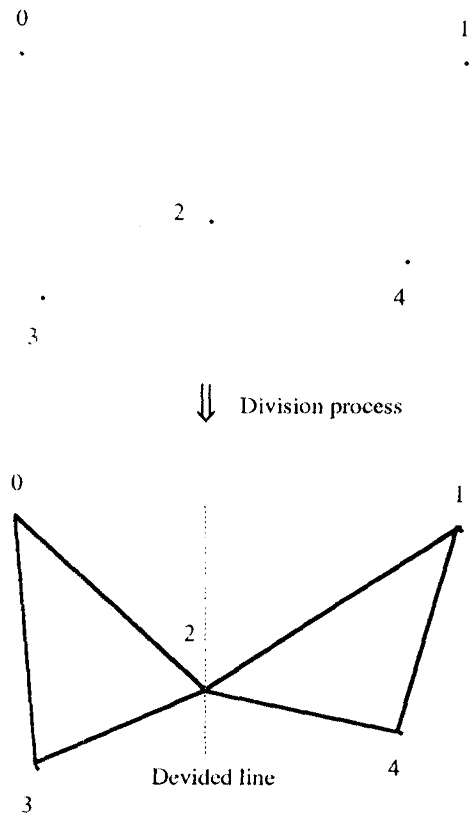

Fig. 1 Division process

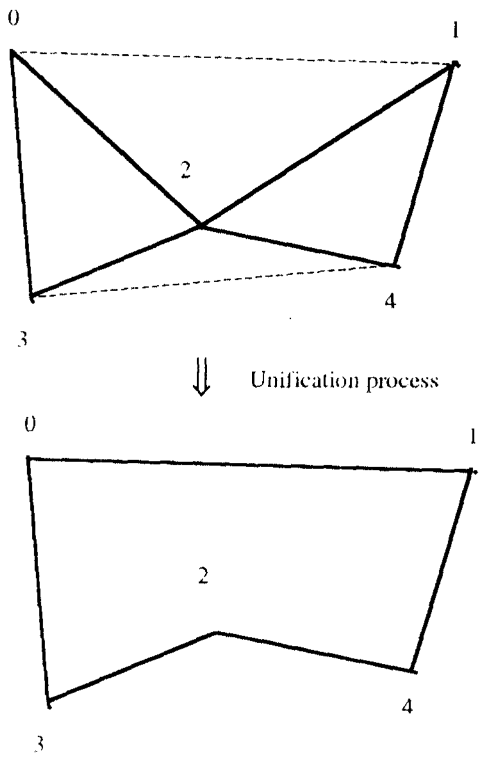

Fig. 2 Unification process

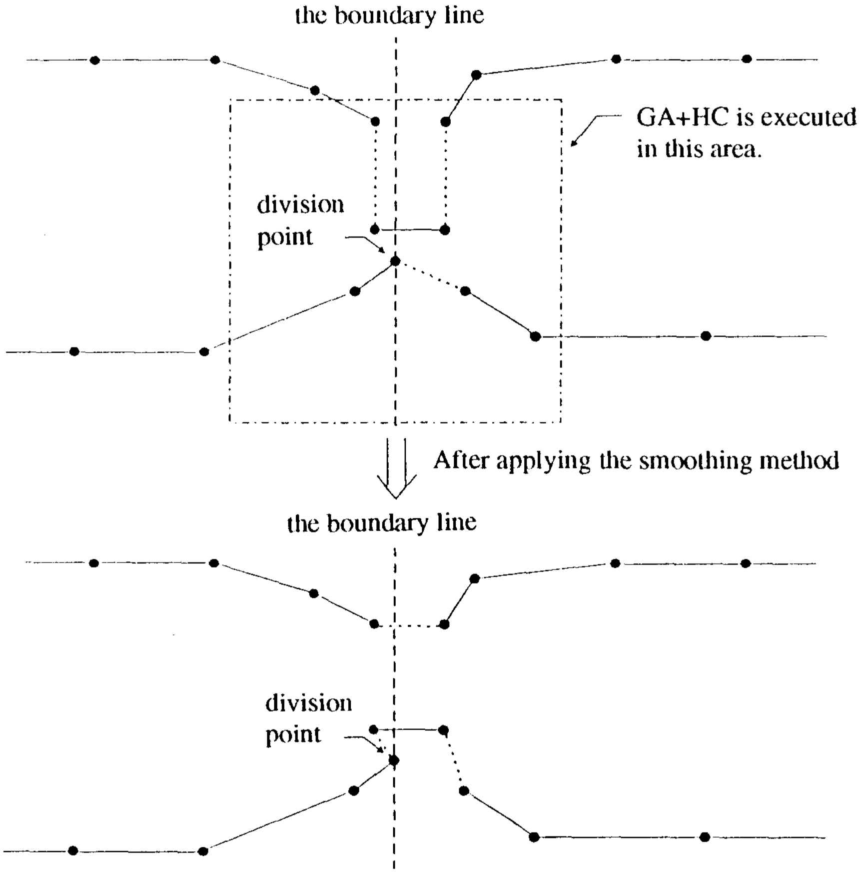

Fig. 3 Smoothig method (SM)

Table 4 Estimation results

Benchmark	Minimum distance	with DU Error[%]	with DU and SM Error[%]	with DU and SM Time[sec]
eli51	426	2	0	23
eli76	538	5	2	52
kroA100	21282	11	0	53
kroD100	21294	11	4	60
eli101	629	8	2	91
lin105	14379	14	1	50
pcb442	50779	14	4	417
pr2392	378032	15	6	2463(=41min.)

4.4 Proposal on smoothing method

As mentioned in the above, if we use DU for large scale problems, approximation error is increasing. To solve it, we propose a new method - *Smoothing method* (for short, SM) - . As illustrated in Fig. 3, SM is the method that GA+HC is executed on division lines. Therefore, the obstacle caused by that all divided regions are independent with each other is decreased. Hereafter, we notate this method as "GA+HC with DU and SM".

4.5 Estimation

Table 4 includes results that "GA+HC with DU" and "GA+HC with DU and SM" are applied to benchmark datum in TSPLIB[6,7].

According to the results in Table 4, by using DU, accuracy is growing bad as the number of cites getting large. It is a natural result that the number of regions is getting large as the number of cites is getting large. In the case of pr2392, the error rate is the worst. However, the result is improved by using SM. the error rate without SM is 15%. It is improved to 9% by using SM. On the other hand, execution time of pr2392 using GA+HC with DU and SM is about 41 minutes. It seems to be within permissible range for practical use although it depends on applications.

As mentioned in the above, from experimental point of view, we can make the distance be shorter using the SM. And execution time can be allowed in practical use.

5. Conclusions

This paper describes on some methods to solve TSP. Among these methods, for large scale problems, GA+HC with DU and SM is effective, and it is verified experimentally.

When using GA+HC with DU and SM, each divided problem is independent with each other. Accordingly, the method is suitable for applying to parallel computation. In future works, we will try it and estimate effectiveness using a parallel computer.

References

1. Kitano, H., Genetic algorithm, SANGYOU-TOSYO, CO, Ltd (1993). (In Japanese)
2. Kubo, M., Invitation to travelling salesman problem I, The Operations Research Society of Japan, Vol. 39, No.1, pp.25-31(1994). (In Japanese)
3. Kubo, M., Invitation to travelling salesman problem II, The Operations Research Society of Japan, Vol. 39, No.2 pp.91-96(1994).(In Japanese)
4. Kubo, M., Invitation to travelling salesman problem III, The Operations Research Society of Japan, Vol. 39, No.3, pp.156-157(1994). (In Japanese)
5. Yamamoto, Y. and Kubo, M., Invitation to travelling salesman problem, ASAKURA-SYOTEN, CO, Ltd(1997). (In Japanese)
6. http://www.iwr.uniheodelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html
7. http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html