OPTIMIZATION METHOD FOR THE MULTI-ROUTE SEARCHING PROBLEM

Eiko SEKINE Yoshio HAMAMATSU

Dep. of Systems Eng., Faculty of Eng., Ibaraki University
4-12-1 Nakanarusawa, Hitachi, IBARAKI 316-8511 Japan.
Phone: +81-294-38-5180 E-mail :eiko@dse.ibaraki.ac.jp

Summary A new optimization method for the multi-route searching problem is proposed. The problem that the vehicles go to the goal through each city from a starting point is discussed. In this problem, the combinations of the routes become enormous, because the combinations of the vehicles and the visited cities need to be considered. The proposed method consists of two parts. (1) Dividing of the area, and (2) Deciding of the routs. In first part, the area is divided using the geometric information. In the second part, the orders of visiting the cities are decided using Nearest Neighbor Heuristics. As the result of the simulations, the solutions are better than that of GA. Furthermore, the computation time of using the proposed method is below 1 second, as against that the computation time of using GA is about 20 minutes.

INTRODUCTION

Japan is going to become the unusual society of having the elderly people around the world. Today, the facilities that are used by the elderly and handicapped person such as day-care centers have the buses with a lift. This lift can carry the wheelchair into the bus. The Delaware Department of Transportation now uses . Around 20 buses manage about 250 trip in a day. In this example, the scheduling of traveling of the vehicles is carried out using a kind of DSS (Decision Support System) which is based on the dynamic programming. However, this system cannot cope with the case such as canceling, because it takes about half a day to get the approximate optimal solution. We compared the combination of this problem and the combination of TSP (Traveling Salesman Problem)[1] which is typical of this kind of problem. We realized that the combination of this problem is 55,458,451 in case of the number of the cities is 10, as against that the combination of TSP is 19,958,400 in case the number of the cities is 12. Because, we need to consider the combinations of the vehicles and the visited cities.

In this paper, we are proposing a new method, which can rapidly search the optimal route using the geometric information.

ASSUMPTION

Fig. 1 shows a simplified location of cities. The vehicles should go to the goal 'G' through each city from a starting point 'S'. To make the problem clear, we will itemize the conditions as the follows.

(1) All the vehicles start from the starting point 'S' should visit each city to carry the customers to the goal 'G'.
(2) Each vehicle shouldn't visit less than one city, and every city should be visited once by only one vehicle.
(3) There are n cities numbered from 1 to n, and there is only one customer at each city.
(4) The distance between two cities is measured as a straight line.

Let the coordinates of 'S' be (0,0), and let the coordinates of 'G' be (a, a). Therefore, the coordinates of each city (x, y) should satisfy the relations 0<x<a, and 0<y<a. We are searching the multi-route for $\lceil \frac{n}{m} \rceil$ vehicles, where m is the capacity of the bus, which satisfies $2 \leq m < n$. The optimal solution will be obtained when the total summation of the traveling distance of each vehicle is the minimum.

OPTIMIZATION METHOD

The proposed optimization method consists of two parts. (1) Dividing of the area, and (2) Deciding of the routs. In first part, we calculate the perspective ratio of cities that will be described later, and divide the area using the value of that ratio. The cities in a divided area should be visited by only one vehicle. The two parts of the detail of this method are explained in the following sections.

DIVIDING THE AREA

In this problem, the shortest route is the diagonal route between the starting point and the goal, while the longest route is the route along the outermost sides that represent right-angle sides. We define the perspective ratio as the ratio between the distance d_i and l_i, (see Fig. 1).

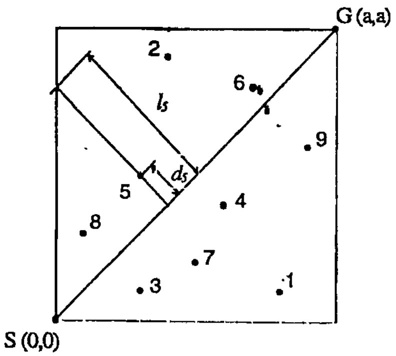

Fig. 1. Location of cities.

$$P_i = d_i/l_i \qquad (1)$$

Where d_i is the perpendicular distance between the focused city, i, and the shortest route. l_i is the distance between the shortest route and the longest route along the distance d_i passing through the focused city. The perspective ratio is zero in case the city is located on the shortest route, and it is 1 when the city is on the longest route. The value of l_i is positive if the city is located to the left of the shortest route; and it is negative for the city that is located on the right of the shortest route. Fig.2 illustrates the concrete example of the perspective ratio when the cities are located as in Fig. 1. To divide the area, the perspective ratios need to be arranged in decreasing order of the value (see Fig. 3). When the capacity of the bus is three, we can decide the cities where each vehicle should visit by dividing the nine cities into three groups from the left side of Fig. 3. Fig. 4 illustrates the example of dividing the area when the cities are located as in Fig. 1. In this case, the first vehicle should visit the cities that are numbered 2, 8, 5, and the second vehicle should visit 6,4,7, and the third vehicle should visit 3,9,1. This is considered as a fundamental way of thinking to dividing the area.

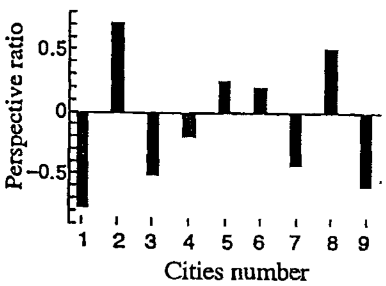

Fig. 2. Perspective ration of cities.

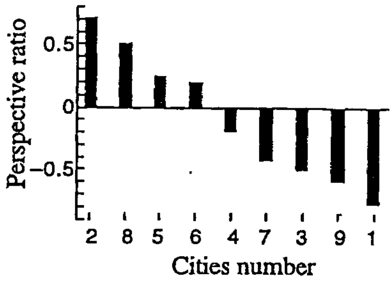

Fig. 3. Perspective ration of cities which is arranged in a decreasing order of value.

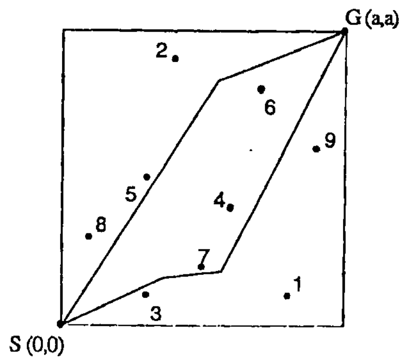

Fig.4 Example of dividing the area.

In deciding the route, we will decide the order of visiting the cities. This kind of problem has a primary factor of increasing the computation time. It is that the combinations of the routes become enormous and we should compare the distance of each route. Thus, we will decide the orders of visiting the cities using Nearest Neighbor Heuristics[1] to dispense with the comparison of the distances.

SIMULATION RESULTS

In this chapter, the proposed method is compared with GA (Genetic Algorithms)[2][3]. At first, we examine the precision of the two methods using the exact solution of the enumerative method as a standard. As an example, we simulated the 15 different patterns of locations, where the number of the cities of each pattern equals to 16 excluding the starting point and the goal. These patterns are made by generating random coordinates' axes of the cities' locations.

Fig. 5 shows the results of the simulations. Where, we defined the error ratio as

$$Error \quad ratio = \frac{S_{p/GA} - S_e}{S_e}. \qquad (2)$$

Where $S_{p/GA}$ is the solution of this method, or the solution of GA, and S_e is the exact solution. The vertical axis represents the error ratio values and the horizontal axis represents the pattern's number. The black dots represent the error ratios of this method. The dotted lines represent the maximum and minimum of the error ratios when the same examples were solved with GA. We figured out that approximately all the solutions of this method are better than that of GA. In this example, the average of error ratios of this method is about 11.7%. This is smaller than the minimum error ratio using GA. Therefore, the solutions of the proposed method are more closed to the exact solutions.

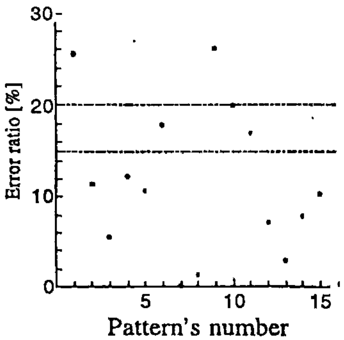

Fig.5. Comparison results of error ratio between the proposed method with GA. The number of cities is 16.

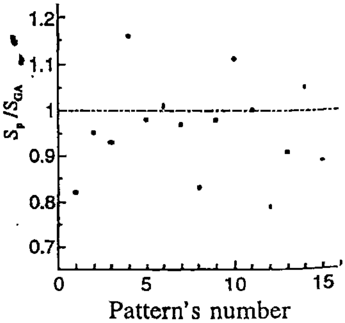

Fig.6. Comparison results of the solution between the proposed method with GA. The number of cities is 30.

Next, we examine the precision of the two methods in case of more numbers of cities are considered. However, we cannot obtain the exact solution of the enumerative method in available time. Thus, we compare the two methods using GA as a standard.

In this simulation, we increase the number of the cities to 30, where 15 different patterns of locations are generated as explained above. Fig. 6 shows the comparison result between this method with GA. The vertical axis represents the ratio of the solution of this method, S_p, to the solutions of GA, S_{GA}. The horizontal axis represents the pattern's number. The dotted line represents the solutions that can be obtained equally by the two methods. If the value is less than 1, it means that the solution of this method is less distance than that of GA. The better solutions than GA are obtained for eleven patterns of fifty. Therefore, the solution of the proposed method can be considered better than that of GA. Furthermore, the computation time of using this method is below 1 second, as against that the computation time of using GA is about 20 minutes, with Sun SPARCstation 20.

CONCLUSION

The merits of the proposed method are that it is faster and more precise compared to GA. Moreover, many methods to approximately obtain the optimal solution need to examine some parameters. For example, in case of GA, we should decide the optimal values of the ratio of crossover and the ratio of mutation for obtaining the satisfactory results. Whereas this method doesn't have such parameters to be examined, and it is easy to be used to obtain accurate solutions. In actuality, it is expected that the cancellations or additional dispatching orders after the starting of vehicles will happen. The proposed method may be applied to such situations because of its merit of operation speed. However, this is an ideal case in which the traveling time is proportional to the distance. In practice, the traveling times will depend on the layout of the roads and the actual traffic conditions. Therefore, for practical application, we need to extend the method to be obtained the optimum routs considering this point.

REFERENCES

(1)Gerhard, R. (1994). The Traveling Salesman-Computational Solutions for TSP Applications. :Springer-Verlag, New York.

(2)Holland, J. (1975). Adaptation in Natural and Artificial Systems. : Univ. Michigan Press.

(3)Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Mass.: Addison-Wesley.