C – 16

A Comparison of Heuristics Algorithms to Solve Vehicle Routing Problem with Multiple Trips and Intermediate Facility

Nur Insani
Department of Mathematics Education
Yogyakarta State University
INDONESIA
nurinsani.utomo@gmail.com

Abstract

This paper considers the design and analysis of algorithms for vehicle routing problems (VRP) with multiple trips. The VRP is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles. The objective is to deliver a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. Given the challenging difficulty of this problem class, heuristics approaches seem to offer the most promise for practical size problems. After describing two heuristics: Sequential Insertion and Nearest Neighbors, we conduct an extensive computational study of their performance. The problems set are taken from a randomly generated instance test problems. Our experimental study shows that the two heuristics performed well in different problem environments; in particular the Nearest Neighbors heuristic consistently gave better results.

Keyword: VRP with multiple trips, Sequential Insertion, Nearest Neighbors.

1. Introduction

One of the most important problems in combinatorial optimization is vehicle routing problem (VRP). The VRP determine the optimal routes to be performed by a fleet of vehicles to serve a given set of customers. Each customer is supplied by exactly one vehicle route. The total demand of any vehicle route must not exceed the vehicle capacity and the total length of any route includes customer travel times and service times must not exceed a specified bound. Since it was introduced by Dantzig and Ramser in 1959, there have been hundreds of researches devoted to solve any variant of this problem with exact or heuristics approaches. Some of the variants of the VRP problems are VRP with Time Windows (VRPTW), where customers may be served within a specified time interval and the schedule of vehicle trips need to be determined, and VRP with Multiple Trips, known as VRPM, where vehicles may work for multiple routes or trips within a given time period. In practice multiple trip scheduling is important since significant cost savings can be achieved if the number of vehicles and hence drivers is reduced. In this paper, we consider a case of VRPM, a variant in which the vehicles may perform more than one route within a working shift.

Owing to the inherent complexity of VRPM, the use of classical methods is only successful for certain types of problems and performs poorly on others. Metaheuristics offer a great advantage over classical methods. They combine various features from different procedures in an effective way to explore the solution space in order to find better solutions and these techniques have been successfully applied to VRP and other combinatorial optimization problems [9]. The first work to explicitly address multiple trips was made by Salhi in 1987 [13], in the context of vehicle fleet mix. He used a matching algorithm to allocate routes to vehicles within a refinement process and limited the double trips. This problem was also tackled by Fleischmann within a working paper [5]. A constructive and improvement heuristic was proposed by Brandão and Mercer [2]. They tackled multiple trips as part of a more extensive problem involving time windows and vehicle fleet mix. Using real test data, results showed that their heuristic produced savings of 20% when compared to the manual schedule. On the other research, they modified their heuristic to solve the classical VRPM. Their approach is based on the nearest neighbor rule and the insertion criterion to assign customers to routes within vehicles. This process is repeated until all unrouted customers are inserted.

According to Laporte [6], some of the heuristics techniques introduced in the literature were Bin-Packing assignment, Tabu Search, Genetic Algorithm, Sequential Insertion and Nearest Neighbors. The last two techniques are known as some of the best algorithm to solve VRP with multiple trips. Insertion method has an advantage in the selection of the customers, i.e. insert a new customer into a set of chosen customers, in order to get a maximal result, while the Nearest Neighbor has the advantage in determining the resulting distances. Chairul et al [3] defined the Nearest Neighbor algorithm as a method to solve the problem by considering the shortest distance. This can minimize the distance and travel time used by the vehicles. Unfortunately, there is not enough review discussing which algorithm from these two which gives the best results, i.e. shortest distance and time. By knowing which algorithm will give the best result, it will help us saving the time and extra energy to solve a problem. Putra & Insani [12] applied the Sequential Insertion and Nearest Neighbors algorithm to a special case of study, a waste collection route. They had a conclusion that the second method was giving the best results, both in distance and time. Thus, in this paper, we apply the two algorithms to some well-known randomly generated instances in order to get some more general conclusions, especially which algorithm gives the shortest travelling distance and time to serve the customers.

2. Literature Review

2.1. *Vehicle Routing Problem with Multiple Trips* (VRPM)

According to Nallusamy et.al [10], the mathematical structure of the VRP is a graph where the cities are the vertices of the graph. Connections between pairs of cities are called edges and each edge has a cost associated with it which can be distance, time or other attribute. If n is the input number of vertices representing cities, for a weighted graph G, the VRP problem is to find the cycle of minimum costs that visit each of the vertices of G exactly once. There are many mathematical formulations for the VRP with a variety of constraints that enforce the requirements of the problem. In this paper, the following notation is used:

 ISBN : 978-602-1037-00-3

- n: the number of cities to be visited; In a graph , it is the number of vertices in the network;
- i, j, k: indices of cities that can take integer values from 1 to n;
- t: time period, or step in the route between the cities;
- $x_{ijt} = \begin{cases} 1 \text{ , if the edge of the network from i to j is used in step t of the route} \\ 0 \text{ , otherwise} \end{cases}$
- d_{ij}: the distance or cost from city i to city j.

The following is an example of one linear programming formulations of the VRP:

The objective function (Z) is to minimize the sum of all costs (distances) of all of the selected elements of the tour:

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n} d_{ij} x_{ijt}$$

The tour is subject to the following constraints. For all values of t, exactly one arc must be traversed, hence:

$$\sum_{i} \sum_{j} x_{ijt} = 1 \text{ for all } t$$

For all cities, there is just one other city which is being reached from it, at some time, hence:

$$\sum_{j} \sum_{t} x_{ijt} = 1 \text{ for all } i$$

For all cities, there is some other city from which it is being reached, at some time, hence

$$\sum_{i} \sum_{t} x_{ijt} = 1 \text{ for all } j$$

When a city is reached at time t, it must be left at time $t+1$, in order to exclude disconnected sub-tours that would otherwise meet all of the above constraints. These sub-tour elimination constraints are formulated as:

$$\sum_{i} x_{ijt} = \sum_{k} x_{jkt+1} \text{ for all } j \text{ and } t$$

In addition to the above constraints the decision variables are constrained to be integer values in the range of [0,1]:

$$0 \leq x_{ijt} \leq 1$$

VRP problems are known as NP-hard problem, which means that the computing time becomes higher in non-deterministic polynomial with as the size of the problem increases. In this case, the more cities visited, the higher the size of the problem. The goal in solving the VRP problem is to minimize the total distance traveled and minimize the number of vehicles used while still meeting the inventory at each location.

3. Discussion

3.1. Nearest Neighbor (NN) Method

The basic principle of the Nearest Neighbor method that is developing the travel route, each vehicle serves the closest city from the last city that has been visited before. In other word, the method solves the problem by determining the closest point with the shortest distance. It is a simple method to solve these problems and has an early solution.

According to Chairul et al [3], here are the Nearest Neighbor algorithm steps:

Step 1 - Choose a starting point as the starting point (0). The starting point is usually the depot or it is selected based on predetermined rules, then go to step 2.

Step 2 - Determine the nearest point (i) from the starting point, and then connect these two points, go to step 3.

Step 3 - Set the last customer ($i - 1$) as a starting point, go to step 2. If the entire customers are already on the network, then go to step 4.

Step 4 - Stop the process.

3.2 Sequential Insertion (SI) Method

According to Laporten in Mustika [8], there are two ways to solve VRP: 1). Combine the existing routes with saving criterion and 2) try sequently insert a customer into the existing route (arcs) using cost insertion criteria. The second method is proven as the most popular method in the literature to solve the VRP as well as scheduling problems. Chairul et.al. [3] stated that this method give a feasible solution, a set of feasible route which is constructed from several sequences of insertion of customers that are not in the any existing routes (arcs). Feasibility is checked for the limitation of the time window and the capacity of the vehicle. Customer and arc that give most small cost and feasible solution is subsequently selected. This procedure is repeated until all customers have been assigned.

The steps of problem solving sequential insertion is as follows:

Step 1 - Choose a starting point as the starting point (0) is selected based on the rules that have been predetermined, go to step 2.

Step 2 - Calculate the travel distance of each vehicle to each customer and calculate the travel time as well, go to step 3.

 ISBN : 978-602-1037-00-3

Step 3 - Calculate the remaining capacity of the vehicle, if the remaining capacity meets the demans of the next customer then go to step 4, otherwise go to step 9.
Step 4 - If it has entered the 2nd customer then go to step 5, if not go to step 6.
Step 5 - Insert the next customer in the existing arc, then go to step 6.
Step 6 - Select the customer who have the shortest distance, then go step 7.
Step 7 - Calculate the travel distance touring, the completion time and the list of customers who have been served. Go to step 8.
Step 8 - If the demand to be delivered to the customer is not all fulfilled yet then back step 2, if it is all done then go to step 10.
Step 9 - Back to the depot, create a new tour, t = t + 1, return to step 2.
Step 10 - All requests has been sent to the customer. Stop the procedure.

3.3 Computational Results & Analysis

In this paper, we used the same randomly generated instances to test the quality of the two algorithms. The problem was implemented in MATLAB 7.0 with Pentium IV processor system. The results from the tests of each algorithm are compared in order to give an indication of the quality of the algorithms. Table 1 lists the algorithms for which all numerical results are available to perform the comparison. Here we used randomly generated coordinates of 150 cities in total. Table 2 shows the details of the tested instances.

Algorithm number	Algorithm name
1	Neural Network algorithm
2	Sequential Insertion algorithm

Table 1: list of algorithms that are numerically compared

Instances	Number of cities allocated
1	10
2	10
3	10
4	10
5	10
6	20
7	20
8	20
9	20
10	20

Table 1: list of algorithms that are numerically compared

 ISBN : 978-602-1037-00-3

Table 3 shows the results of NN & SI algorithms. Algorithms were applied to the given set problems iteratively. We took the optimal results of the distance after performing approximately 10 iterations.

Instances	Algorithm 1 (in km)	Algorithm 2 (in km)
1	4275	4356
2	2560	2490
3	561	601
4	7844	7933
5	432	466
6	817	898
7	6651	6855
8	4874	5279
9	1189	1341
10	3980	4347

Table 3: Results of NN vs SI algorithms

CONCLUSION & SUGGESTION

Conclusion

From the results obtained, it can be clearly inferred that NN yields better solutions to the VRPM. This result might have been obtained in a lesser computational time than the exact methods like Branch and Bound, Branch and Cut and Cut and Solve techniques. It is not the global optimal solution and might turn out to be a close to global optima or a local optima. However, this paper does not show the completion time yet. Moreover, the solution may further fall in value after a large number of iterations and more cities involved. This may cost us further computational time and usage of computer memory. We can conclude that Neural Neighbors gives a better solution between the efficiency and the optimality of the final result obtained, than Sequential Insertion algorithm The work can be further extended by generating bigger set of problems and using a combination of k-clustering algorithm to a big area of cities.

REFERENCES

[1] Angelelli, E., & Speranza, M., G., (2002). *The Periodic Vehicle Routing Problem with Intermediate Facilities*. European Journal of Operational Research, Vol. 137, pp. 233-247

[2] Brandão, J., Mercer, A., *A tabu search algorithm for the multi-trip vehicle routing and scheduling problem*, European Journal of Operational Research, Volume 100, Issue 1, 1 July 1997, Pages 180-191, ISSN 0377-2217,

[3] Chairul A., dkk. (2014). *Penentuan Rute Kendaraan Distribusi Produk Roti Menggunakan Metode Nearest Neighbour dan Metode Sequential Insertion*. Jurnal Online Institut Teknologi Nasional, Vol. 01, No. 04

[4] Fatharani, A., dkk. (2013). *Penentuan Rute Kendaraan Pengangkut Sampah dengan Menggunakan Metode Nearest Neighbour (Studi Kasus PD. Kebersihan Kota Bandung)*. Jurnal Online Institut Teknologi Nasional, Vol. 01, No. 01

[5] Fleischmann, B. (1990). *The vehicle routing problem with multiple use of vehicles.* Working paper. Fachbereich Wirschaftswissenschaften, Universitat Hamburg.

[6] Laporte G., (1983). *A Branch and Bound Algorithm for Capacitated Vehicle Routing Problem.* Operation Research Specktrum 5: 77-85.

[7] Lisye F., dkk. (2009). *Penentuan Rute Truk Pengumpulan dan Pengangkutan Sampah di Bandung*. Jurnal Teknik Industri, Vol. 11, pp. 51-60

[8] Mustika, R., (2008). *Usulan Rute Kendaraan dengan Menggunakan Algoritma Sequential Insertion di PT. Coca-cola Bottling Indonesia*, Bachelor Thesis. Institut Teknologi Nasional. Bandung.

[9] Moin, N.H. (----).*Hybrid Genetic Algorithms for Vehicle Routing Problems with Time Windows*. University of Malaya. Malaysia.

[10] Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., & Parthiban, P. (2010). Optimization of multiple vehicle routing problems using approximation algorithms. *arXiv preprint arXiv:1001.4197*.

[11] Petch, R.J, Salhi, S., *A Multi-Phase Constructive Heuristic for the Vehicle Routing Problem with Multiple Trips*, Discrete Applied Mathematics, Volume 133, Issues 1–3, 15 November 2003, Pages 69-92, ISSN 0166-218X, http://dx.doi.org/10.1016/S0166-218X(03)00434-7.

[12] Putra & Insani. (2014). *The Efectivity Of Sequential Insertion Method And Nearest Neighbour Method In Determining Waste Collection Route*. Graduate Thesis. Yogyakarta State University.

[13] Salhi, S. (1987). The *Integration Of Routing Into The Location-Allocation And Vehicle Composition Problems.* Ph.D. Thesis. University of Lancaster. pp. 198–208.

[14] Singer, B. (2009). The Multiple Trip Vehicle Routing Problem.

ISBN : 978-602-1037-00-3

[15] Suprayogi & Yusuf P. (2008). *Algoritma Sequential Insertion untuk Memecahkan Vehicle Routing Problem dengan Multiple Trips, Time Window, dan Simultaneous Pickup Delivery*. Performa, Vol. 7, No. 1:88-96

[16] Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem*. Bologna: Universita degli Studi