A Heuristic Algorithm for Multiple Depots Vehicle Routing Problem with Backhauls

Jianhua Fan
Tianjin University of Technology
School of Computer Science and Technology
Tianjin, 300191, China
fan109@163.com

Xiufeng Wang and Qiushuang Chen
Nankai University
School of Information Technology and Science
Tianjin, 300071, China
wangxf@nankai.edu.cn

Abstract

An algorithm for multiple depots vehicle routing problem with backhauls is proposed and the goal is to minimize the total traveling cost in reasonable computation time. Firstly borderline customers are determined and the remaining customers are allocated to the nearest depot. Secondly, an effective method is presented to construct the initial routings and insert the borderline customers into the appropriate location. Finally, the initial routings are improved by post-optimization procedure. Some reduction computation strategies are designed to enhance algorithm efficiency. The experiment results indicate that this algorithm can produce encouraging results with less computation time.

Keywords—multiple depots, VRPB, pick-up and delivery

1. Introduction

The Vehicle Routing Problem with Backhauls (VRPB) is an extension of the Capacitated Vehicle Routing Problem (CVRP). The VRPB involves two different subsets of customers known as linehauls and backhauls. A linehaul requires a given quantity of goods from a central depot, whereas a given quantity of goods is collected from a backhaul and brought back to the depot. Mixed vehicle routings with both linehauls and backhauls are needed to serve the set of customers to minimize the total traveling cost. In this article, we consider the special case where the backhauls must be visited after the linehauls on each routing. This constraint is motivated by the fact that most vehicles are rearloaded, so it is inconvenient to rearrange onboard delivery loads to accept new pick-up loads.

* This work was supported by the National Natural Science Foundation of China (No.70572045) to Xiufeng Wang, and the Science and Technology Research Council of Tianjin (No. 05YFJMJC01300) to Qiushuang Chen.

The VRPB can produce much traveling cost saving than designing the routings for linehual and backhaul separately. The importance of VRPB can be attributed to the effort by taking advantage of the unused capacity of an empty vehicle traveling back to the depot. The Interstate Commerce Commission News estimated that backhauling can save 42 million gallons fuel a year nationally [1][2].

VRPB is extremely frequent in practical situations. A common example is that in grocery industry, supermarkets and shops are linehaul who need delivery from the depot, and grocery suppliers are backhaul who offer the goods carried back to the depot.

VRPB has recently attracted many researchers' attention and some exact and approximate algorithms are presented. Toth and Vigo develop a branch-and-bound algorithm, and the lower bound on the optimal solution is derived from a Lagrangean relaxation [3]. Yano et al [4] develop an exact algorithm based set covering for a practical VRPB application in quality stores industry. The exact algorithm can solve the small sized problems to optimization, but large sized instances cannot be solved efficiently. Hence, heuristic algorithms are becoming the main method for effectively solving the practical VRPB.

Toth and Vigo [5] develop a cluster-first route-second heuristic based on the K-tree Lagrangian relaxation approach for the symmetric and asymmetric VRPB. Meta-heuristics are a recent class of approximate methods that have been used widely in combinatorial optimization problems. Potvin [6] proposed a genetic algorithm for VRPB with time windows, and Osman [7] present a reactive tabu search heuristic for VRPB.

At present, most literatures for VRPB refer to the single depot case, and the multiple depots instances are rare. In practical application, some large-scale enterprises have several chain stores, so it is necessary to allocate the customers to the appropriate depot in an effective manner to minimize the total traveling cost. This paper presents an algorithm for the multiple depots vehicle routing problems with backhauls (MDVRPB), where each routing include both linehaul and backhaul. Reduction cost strategies are used to cut down the computation efforts effectively.

The remainder of the paper is organized as follows. In section 2 we formally give the description of the MDVRPB and the algorithm framework. Section 3 gives the detailed implement procedure of the algorithm, and the reduction computation cost strategies are introduced in Section 4. Computational results are given in Section 5, and conclusions are drawn in Section 6.

2. The description of MDVRPB and the algorithm framework

2.1. The description of MDVRPB

The MDVRPB can be formulated as follows. Let G= (V, E) be a complete undirected graph with vertex set $V=N\cup K$, $N=\{1,2,\ldots,n\}$ is the set of customers including linehaul and backhaul, and $K=\{1,2,\ldots,k\}$ represent the set of depots. Edge set $E=\{(i, j): i, j\in V, i<j\}$ is the distance between customers or between customers and depots. A nonnegative demand l_i to be delivered or b_i to be collected is associated with each vertex i in N. In each depot, there are a fleet of identical vehicles, each with capacity Q. The MDVRPB consists of finding a set of minimum cost vehicle routings such that:

- each vehicle serve one routing;
- each customer is visited exactly once;
- linehaul precede backhaul on each vehicle routing;
- the total demands of the linehaul and backhaul customers visited by a vehicle do not exceed Q separately;
- each routing originate one of the depots and return to the same depot.

In this paper, we assume that the number of customers, customer position, customer demands, vehicle traveling speed, and vehicle capacity are known in advance. The distance between customers and between customers and depots are assumed to be known, to be symmetric, and to satisfy the triangle inequality.

2.2. The algorithm framework for MDVRPB

The main idea of the algorithm for MDVRPB of this paper is as following. Firstly, determine the customers set which each depot serve. The customers are divided into borderline customers and non borderline customers according to some criterion. Each non borderline customer is assigned to the nearest depot, and the borderline customers are left unassigned. Secondly, in each depot, initial routings are constructed satisfying the constraints to serve the customers assigned to the depot, and each borderline customer is then inserted into an existing routing or an empty routing. Finally improve all the routings. The main steps of algorithm are as follows:

Step 0: Set the borderline parameter $\gamma=\gamma_0$ and a step size Δ, in this paper let $\gamma_0=0.7$ and $\Delta=0.1$.

Step 1: Determine the borderline customers set according to the value of γ and assign the non borderline customers to their nearest depots.

Step 2: Construct the initial routings for each depot, and in each routing all the linehaul precede the backhaul.

Step 3: Insert the borderline customers into the most appropriate routings or empty routings (when there is not available routing).

Step 4: Improve the routings.

Step 5: If $\gamma=\gamma_{max}$, stop, else set $\gamma=\gamma+\Delta$, go to step 1.

3. Implement of the algorithm

In this section, we provide the mathematical notations and the procedure of the algorithm.

3.1. Mathematical notation

The mathematical notation needed is as follows:

$N=\{1,2,\ldots,n\}$: the set of customers, including linehaul and backhaul customers

$K=\{1,2,\ldots,k\}$: the set of depots

D_{ij}: the distance between customer i and j, $i, j\in N$

d_i^k: the distance between customer i ($i\in N$) and depot k ($k\in K$)

l_i: the demand to be delivered to customer i

b_i: the demand to be collected from customer i

Q: the capacity of vehicle

P_i: the nearest depot to customer i

P_i': the second nearest depot to customer i

E_i: the set of depots which can serve customer i

A(i): the most economical depot to serve customer i ($A(i)\in E_i$)(which need computation to determine)

γ: a prescribed positive value for the borderline parameter, $0<\gamma\leqslant 1$

B: set of borderline customers

$V(\gamma)=|B|$: the number of borderline customers corresponding to the value of γ

C_i^k: the least extra cost of serving customer i from depot k

R_k: the set of routings served from depot k

r_j^k: the jth routing served from depot k, $j\in\{1,\ldots,|R_k|\}$

$Q(r_j^k)$: the sum of demand to be delivered to linehaul customers in routing r_j^k

$Q'(r_j^k)$: the sum of demand to be collected from backhaul customers in routing r_j^k

$L(r_j^k)$: the length of routing r_j^k

3.2. The determination of borderline customers

In this paper, we use the method of Salhi [8] to define the customers. A borderline customer is the one who locate nearly half way between his two nearest depots, so it's needed to determine by which depot the customer is served. The procedure for determining is as follows:

for each customer $i \in N$ do
  begin
    compute $\theta = d_i^{Pi} / d_i^{Pi'}$
    if $\theta \leqslant \gamma$ then customer i is allocated to its nearest depot P_i and set $E_i = \{ P_i \}$
    else customer i is determined as a borderline customer and set $B = B \cup \{i\}$, $E_i = \{ P_i, P_i' \}$
  end
$V(\gamma) = |B|$

For example, supposing that $\gamma = 0.7$, if $\theta = d_i^{Pi} / d_i^{Pi'} = 0.8$, customer i is a borderline customer which locate nearly half way between P_i and P_i', and the nearest depots set is $E_i = \{P_i, P_i'\}$. Else if $\theta = d_i^{Pi} / d_i^{Pi'} = 0.5$, the distance between i and P_i is more little than that between i and P_i', so i is not a borderline customer, which should be served by depot P_i.

3.3. Construction of the initial routings

Jacobs [2] showed that the VRPB routing consists of three parts fundamentally. The first is a path from the depot through linehaul, ending at the linehaul interface customer. The second component is the interface link between linehaul and backhaul. The third is a path from the depot through backhaul, terminating at the backhaul interface. The set of linehaul customers on the linehaul path comprises a sector of the plane anchored at the depot. The set of backhaul customers on the backhaul path also comprises a sector of the plane anchored at the depot. The best savings from backhauling can be attained by minimizing the angles of the linehaul and backhaul sectors as well as the angle between the linehaul and backhaul sectors. According to this property, we propose a new routing construction method. For each depot, the initial routing construction procedure is as follows:

Step 0: Several sector regions are formed around the depot to represent each of the VRPB routings. The total demands of linehaul and backhaul customers in each sector region do not exceed the vehicle capacity Q separately, namely $\sum l_i \leqslant Q$ and $\sum b_i \leqslant Q$.

Step 1: Compute d_i^k for each linehaul customer and backhaul customer in each sector region.

Step 2: For each sector region, linehaul customers are sequenced by increasing distance to the depot (d_i^k) and backhaul customers by decreasing distance, and the linehaul path and backhaul path are formed.

Step 3: The initial routing is constructed by uniting the linehaul path and backhaul path.

Step 4: Repeat Step 1 to Step 3 until all the routings are formed. A routings construction example is depicted in Figure 1. There are four routings in the figure, each of which includes linehaul path and backhaul path.

3.4. Insertion of the borderline customers

For each borderline customer i, a strategy is needed to determine which depot to serve the customer i from the two nearest depots (P_i, P_i'). Firstly, we compute the least insertion cost in these two depots for customer i, and then choose the routing which have the least insertion cost. If there are not available existing routings to insert into, a new routing including the borderline customer is formed.

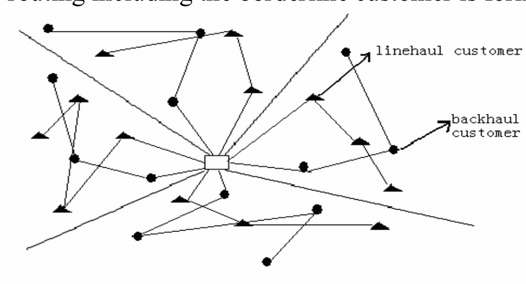

Figure 1. The initial routings construction

When inserting a borderline customer into an existing routing, customer type needs to be judged. If the customer is linehaul, locations allowed to insert into are as follows:

- between the depot and the first linehaul customer;
- between two linehaul customers;
- between the last linehaul and the first backhaul ;

The procedure for backhaul customer is similar. An example is depicted in Figure 2, linehaul customer can only be inserted into any position from the depot to customer d and backhaul customer can be inserted into position from customer c to the depot.

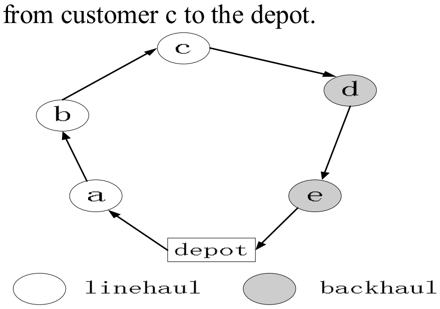

Figure 2. Borderline customer insertion

Computation of the least insertion cost C_i^k for linehaul customer is as follows:

Step 0: j=1

Step 1: if $Q(r_j^k) + l_i \leq Q$, compute the least insertion cost into routing r_j^k for customer i as following

$$EXC(r_j^k) = L(r_j^k \cup \{i\}) - L(r_j^k) = D_{i1,i} + D_{i,i2} - D_{i1,i2}$$

Step 2: if $j < |R_k|$, $j = j+1$, go to Step 1; else, go to Step 3

Step 3: if the existing routings $r_j^k \in R_k$ ($j=1,2,\ldots|R_k|$) are available, set $C_i^k = \min\{EXC(r_j^k), j=1,2,\ldots|R_k|\}$, else, go to Step 4

Step 4: A new routing (j+1) originating from depot k is formed only including i, set $C_i^k = 2d_i^k$, $R_k = R_k \cup \{j+1\}$

The insertion of borderline customer is as follows:

for each borderline customer i ($i \in B$), do

0-7695-2874-0/07 $25.00 © 2007 IEEE

begin
compute C_i^k ($k \in E_i$)
set $C_i^{A(i)}=\min\{ C_i^k, k \in E_i \}$
allocate customer i to the corresponding routing of depot A(i),
set B=B-{i}
end

The initial routings including all customers are formed.

3.5. Improvement of routings

The initial routings are improved by applying some post-optimization procedure, which include three phases:

- Intra-routing post-optimization

In each routing, we use the extension of two-exchange method[9] to improve the routing, which is depicted in Figure 3. Given two arcs (a, b) and (d, e)(with $a \neq e$ and $b \neq d$), a two-exchange is obtained by replacing arcs (a, b) and (d, e) with arc (a, d) and (b, e). For each VRPB routing, the feasible exchanges of two arcs belonging to the routing must consider the precedence constraint between linehaul and backhaul. We call linehaul arc (resp. backhaul arc) an arc whose endpoints are both linehaul (resp. backhaul). We call interface arc an arc starting from a linehaul and leading either to a backhaul or to the depot.

If one of the exchanged arcs is a linehaul arc, then the other arc can only be a linehaul or an interface arc. Otherwise, if one of the exchanged arcs is an interface or a backhaul arc, then the other arc can only be a backhaul arc. As depicted in Figure 3, Figure 3(1) is a feasible exchange and Figure 3(2) is an infeasible exchange.

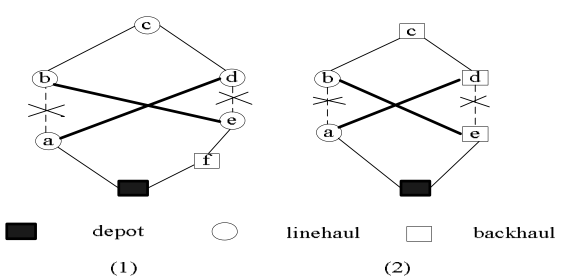

Figure 3. Two-exchange method of VRPB

- Inter-routing post-optimization within depot

We use following methods to implement it:

One-move procedure: deletes a linehaul (or backhaul) customer from one routing, and inserts this linehaul (or backhaul) customer between two linehaul (or backhaul) customers or the last linehaul and the first backhaul customer in another routing.

One-exchange procedure: interchanges two linehaul (or backhaul) customers in two different routings. The new locations for inserting these two customers must satisfy the precedence constraint of linehaul and backhaul.

- Inter-routing post-optimization between depots

We use the one-move and one-exchange method to perform inter-routing improvement between depots.

4. Reduction computation cost strategies

The initial routing constructed by step 0 to step 3 needs less computation effort, but have low solution quality and need to be improved. As the number of customers increase, the search neighborhoods of step 4 become very large. It is necessary to adopt some methods to avoid performing unnecessary calculations at the expense of a negligible loss in solution quality. In this paper, we adopt the strategies below to reduce the computation cost.

4.1. The reduction cost between depots

If the distance between customer i and depot k is far larger than the distance between customer i and depot m, the appropriate depot to serve customer i is depot m, so we don't consider exchanging customer i between depot k and m. We consider those customers who are situated within a certain area to reduce computation time. With customer i (served by depot m) and depot k and m, we propose the exchange procedure as follows:

- if $i \in B$ and $E_i=\{k,m\}$, customer i is the borderline customer between depot k and m, and can be moved into the appropriate routings within depot k.
- if $i \in B$ and $E_i \neq \{k,m\}$ or $i \in N-B$, compute d_i^k and d_i^m , if $d_i^k<2d_i^m$, an appropriate insertion location for i is searched within depot k, else i is far from depot k, it's unnecessary to exchange between depot k and m.

4.2. The reduction cost with γ

Here we use the method [8] to reduce computation.

- if $\gamma_1 < \gamma_2$ and $V(\gamma_1)=V(\gamma_2)$, there is no need to run the algorithm for γ_2, because the borderline customers partition is the same, the running results may be the same.
- if $V(\gamma_1)=0$ and $\gamma_1 < 1$, there is no need to run the algorithm for other valued of $\gamma \geqslant \gamma_1$.
- if $\gamma_1 < \gamma_2$ and customer i was not a borderline customer when $\gamma=\gamma_1$, then customer i can't be borderline customer when $\gamma=\gamma_2$, thus to avoid repeated computation.

5. Computational experiments

Most literatures consider VRPB or MDVRP problem, and to our knowledge, no algorithm for combination of VRPB and MDVRP has been seen. So no benchmark instances exist for this class of problem in this paper. We suppose the customer locations are uniformly generated in a square with $x \in [0,100]$ and $y \in [0,100]$, and the capacity of vehicle(Q) to be 25. The algorithm performance is

0-7695-2874-0/07 $25.00 © 2007 IEEE

analyzed from traveling cost and computation time.

5.1. Comparison with MDVRP

To prove the traveling cost saving by backhauling, we apply the algorithm of this paper into MDVRPB and MDVRP. In MDVRPB each routing includes linehaul and backhaul simultaneously, whereas in MDVRP each routing can only include linehaul or backhaul. We suppose there are three depots, and the coordinates are (40,65), (55,25) and (90,55) separately. The demand of customers are integral, uniformly generated in [1,15]. The number of customer is 25, 50 and 100 separately, and in each instance backhaul customer proportion is 10%, 30% and 50% separately. Each instance runs 10 times and the average results are illustrated in table 1.

Table 1. Comparison with MDVRP

Size	Backhaul proportion (%)	MDVRPB		MDVRP	
		Number of routings	Traveling cost	Number of routings	Traveling cost
25	10	7.0	553.4	7.0	546.3
	30	6.0	568.5	8.0	572.4
	50	5.0	524.3	7.0	562.7
50	10	11.1	942.9	12.4	1042.7
	30	10.2	987.2	12.7	1032.1
	50	9.2	971.5	11.7	1075.3
100	10	21.7	1792.5	22.3	1865.2
	30	19.2	1813.5	21.1	1905.6
	50	16.3	1852.8	22.5	1923.8

Table 1 illustrates the potential benefits of backhauling. The MDVRPB reduces the number of routings and traveling cost, with an average traveling cost saving of 4.9% and a maximal saving of 9.6%.

5.2. Comparison of computation time

To prove the efficiency of reduction computation cost, we test the above data using algorithm without reduction computation cost (Algorithm A) and with reduction computation cost (Algorithm B). The running results of two algorithms are depicted in table 2.

Table 2. Comparison of computation time

Size	Backhaul proportion (%)	Algorithm A		Algorithm B	
		Traveling cost	Time (s)	Traveling cost	Time (s)
25	10	543.4	6.8	552.1	6.4
	30	561.6	6.1	569.3	5.9
	50	557.6	5.7	553.8	5.3
50	10	1026.4	18.3	1030.7	10.5
	30	996.2	17.2	1008.7	10.1
	50	1049.5	16.3	1042.5	9.4
100	10	1913.6	49.2	1926.8	21.5
	30	1886.4	47.1	1905.1	20.3
	50	1908.2	44.3	1897.1	18.7

It's shown that the algorithm with reduction computation cost cut down a lot of computation time, but the average loss in quality is only around 0.4%.

6. Conclusion

This paper presents an algorithm for multiple depots vehicle routing problem with backhaul. Firstly, we determine the borderline customers and the remaining customers are allocated to the nearest depot. Secondly, we present an effective method to construct the initial routings, and the borderline customer is inserted into the routings. Finally, we improve the initial routings by some post-optimization strategies. In order to cut down the computation time, we present some reduction computation cost method. The experiment results indicate the algorithm produce encouraging performance. One limitation of the present approach is that it only consider the same vehicle, does not take into accounts the different vehicle types, which are the topics for further research. Furthermore, this approach has not been applied into the actual situation due to some constraint, so the future work is to combine the algorithm with practice effectively.

References

[1] A. Kearney, *Measuring and Improving Productivity in Physical Distribution*, a Report Prepared for the National Council of Physical Distribution Management, NCPDM, Oak Brook, IL, 1984.

[2] B. Jacobs, *The Vehicle Routing Problem with Backhauls: Properties and Solution Algorithms,* Materials Handling Research Centre Technical Report MHRC-TR-88-13, Georgia Institute of Technology, Atlanta,1993.

[3] P. Toth and D. Vigo, " An exact algorithm for the vehicle routing problem with backhauls", *Transportation Science*, vol. 31(4), pp. 372-385, 1997.

[4] C. A. Yano, T. J. Chan and L. Richter et al, "Vehicle routing at quality stores" , *Interfaces*, vol. 17(2), pp. 52-63, 1987.

[5] P. Toth and D. Vigo, "A heuristic algorithm for the symmetric and asymmetric vehicle routing problem with backhauls," *European Journal of Operational Research*, vol.113, pp. 528-543, 1999.

[6] J. Y. Potvin, C. Duhamel and F. Guertin, "A genetic algorithm for vehicle routing problem with backhauling," *Applied Intelligence*, vol. 6, pp. 245-355, 1996.

[7] H. Ibrahim and A. Niaz, "A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls," *Journal of Scheduling*, vol. 5, pp. 263-285, 2002.

[8] S. Salhi and M. Sari, "A Multi-level composite heuristic for the multi-depot vehicle fleet mix problem,", *European Journal of Operational Research*, vol. 103, pp. 95-112, 1997.

[9] G.A.P. Kindervater and M.W.P. Savelsbergh. Vehicle routing: handling edge exchanges. In: Aarts, E.H., Lenstra, J.K.(Eds.), Local Search in Combinatorial Optimization. Wiley, Chichester, 1997.

Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)
0-7695-2874-0/07 $25.00 © 2007 IEEE