DOI : 10.3745/JIPS.2009.5.4.237

A Geometrical Center based Two-way Search Heuristic Algorithm for Vehicle Routing Problem with Pickups and Deliveries

Kwangcheol Shin*

Abstract: The classical vehicle routing problem (VRP) can be extended by including customers who want to send goods to the depot. This type of VRP is called the vehicle routing problem with pickups and deliveries (VRPPD). This study proposes a novel way to solve VRPPD by introducing a two-phase heuristic routing algorithm which consists of a clustering phase and uses the geometrical center of a cluster and route establishment phase by applying a two-way search of each route after applying the TSP algorithm on each route. Experimental results show that the suggested algorithm can generate better initial solutions for more computer-intensive meta-heuristics than other existing methods such as the giant-tour-based partitioning method or the insertion-based method.

Keywords: *Vehicle Routing Problem, Heuristic Algorithm, Initial Solution*

1. Introduction

The vehicle routing problem (VRP) is a combinatorial optimization and nonlinear programming problem seeking to service a number of customers with a fleet of vehicles. VRP has been an important problem in the fields of transportation, distribution and logistics since Dantzig and Ramser [1] first proposed the problem. The vehicle routing problem with pickups and deliveries (VRPPD) is an extension of the classical VRP. Recently, research on VRPPD has peaked essentially because pickup demands for packaging and used product returns from customer locations have increased substantially due to environmental and government regulations, and the fact that integrating pickups with deliveries maximally utilizes vehicle capacity and saves money. So, VRPPD must take into account the goods that customers return to the delivery vehicle. This restriction makes the planning problem more difficult and increases travel distances or the number of vehicles. Restricted situations, in which there are no interchanges of goods between customers and all delivery demands start from the depot and all pickup demands are brought back to the depot, are usually considered in VRPPD.

The VRPPD model can be classified into three: Delivery-first and pickup-second; mixed pickups and deliveries; and simultaneous pickup and deliveries. The assumption of the delivery-first and pickup-second model is that all deliveries must be made before any pickups. This assumption was due to the fact that vehicles were rear-loaded and because rearranging delivery loads onboard to accommodate new pickup loads was difficult. However, in recent days, most vehicles have side-loading as well as rear-loading functions. To accommodate new pickup loads, rearranging delivery loads onboard is no longer a requirement. Hence, the assumption that all deliveries must be made before any pickup can occur can be relaxed, allowing for the mixed pickups and deliveries model in which deliveries and pickups may occur in any sequence on a vehicle route. When customers can simultaneously receive and send goods, it is referred to as simultaneous pickup and delivery. A VRPPD solution is feasible only if the following three conditions are satisfied: delivery-feasibility, pickup-feasibility and load-feasibility. Delivery-feasibility and pickup-feasibility mean that both the total delivery and the total pickup demands on any vehicle route do not exceed the maximum capacity of the vehicle; and load-feasibility means that the maximum capacity of the vehicle is not exceeded at any point on the route.

This study focuses on solving mixed pickups and deliveries by applying the two-phase heuristic algorithm. The first phase selects cluster seed as the farthest node (customers) among unclustered nodes and cluster nodes by using the notion of the geometrical center of a cluster. The result of the first phase satisfies the delivery-feasibility and pickup-feasibility since the total deliveries and pickups on any route is less than or equal to the maximum capacity of the vehicle. And the second phase of the algorithm applies the TSP algorithm on each cluster to find the shortest route regardless of load-feasibility and then finds routes that sat-

※ This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MEST) (No. 2009-0071125)
Manuscript received November 2, 2009; accepted November 24, 2009.
Corresponding Author: Kwangcheol Shin
* Dept. of Information and Communications Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea (kcshin2005 @kaist.ac.kr)

isfy the load-feasibility by applying the direction sensitive two-way search algorithm. The result of the second phase satisfies all VRPPD constraints of load-feasibility as well as delivery-feasibility and pickup-feasibility. The experimental results show that the suggested two-phase heuristic can generate better initial solutions than giant-tour-based partitioning or the insertion-based method for more computer-intensive meta-heuristics.

This study is organized as follows: Section 2 reviews the related literature on VRPPD and defines VRPPD, and Section 3 explains the details of the suggested two-phase algorithm. Section 4 discusses experimental results, and finally, Section 5 offers our conclusions and future direction.

2. Literature Review and Definition

The literature on VRPPD can be classified into three main categories by following its model: delivery-first and pickup second; mixed pickup and delivery; and simultaneous pickup and delivery.

2.1 Delivery-first and pickup second

The original problem addressed in VRPPD is referred to as classical VRPPD in the literature where the delivery and pickup demands are different, and on a route all the delivery demands have to be served before pickup demands. The reason for this route constraint was the inconvenience caused by 'mixed' loads for rear-loaded vehicles. Exact methods for the classical VRPPD were proposed by Toth and Vigo [2] and Mingozzi et al. [3]. Heuristics for delivery-first and pickup second have been studied in various ways and can be found in the works of Deif and Bodin [4], Goetschalckx and Jacobs-Blecha [5], Thangiah et al. [6], Duhamel et al. [7], Toth and Vigo [8], and Brandão [9].

2.2 Mixed pickup and deliveries

Problems that allow serving pickup demands before all delivery demands are served on a route are referred to as mixed VRPPD. The reasons for relaxing the route constraint are that nowadays most vehicles have side-loading and rear-loading functions, and the fact that allowing mixed loads on a route reduces costs. As far as we know, all available algorithms developed for mixed cases are heuristics such as those in the works of Min [10], Dethloff [11-12], Angelelli and Mansini [13], Tang and Galvão [14-15], and Nagy and Salhi [16-17]. Most heuristics are based on classical procedures including insertion methods, edge and vertex exchanges, and customer relocations.

2.3 Simultaneous pickup and delivery

Both the classical and mixed VRPPD require that delivery and pickup customers be different. However, in today's context a customer may have both delivery and pickup demands. Problems that address this situation are referred to as vehicle routing problems with simultaneous delivery and pickup (VRPSDP). VRPSDP was introduced by Min [10]. Heuristics to solve the problem were developed by Dethloff [11], Chen and Wu [18], and Tang and Galvão [15].

2.4 Problem definition

This study focuses on solving mixed pickups and deliveries. The vehicle routing problem with pickup and delivery, VRPPD, is defined as follows: Let $G=(V,A)$ be a complete and directed graph with vertex set $V=\{0,1,\cdots,n\}$, where vertex 0 represents the depot, and each remaining vertex represents a customer. The arc set is defined as $A=\{(i,j):i,j\in V,i\neq j\}$. A fleet of *m* identical vehicles of capacity *Q* is based at the depot. Each customer *i* has a non-negative pickup demand p_i and a non-negative delivery demand d_i, satisfying $p_i+d_i>0$ for all *i*. Initially, all delivery demands are located at the depot and ultimately all pickup demands must arrive at the depot. The VRPPD consists of designing at most *m* vehicle routes starting and ending at the depot, such that each pickup and each delivery is performed by one vehicle, the total load of a vehicle along a route never exceeds *Q*, and the total routing cost is minimized.

A VRPPD solution is feasible only if three conditions are satisfied: delivery-feasibility, pickup-feasibility and load-feasibility. Delivery-feasibility and pickup-feasibility mean that both the total delivery and the total pickup demands on any vehicle route do not exceed *Q*. Load-feasibility means that *Q* not be exceeded at any point on the route.

2.5 Methods for finding Initial solutions

To get the final solution of a VRPPD, generally two steps are required the first procedure to get initial solutions and the second procedure to optimize the initial solutions.

This study focused on the first procedure and there are two frequently used algorithms to find an initial solution. The first one is the giant-tour partition-based method on the concept of multiple giant-tours [21]. The giant-tour partition was used in the work of [17] and its time complexity is O(n^2). The second one is the insertion-based heuristic algorithm that first makes routes by using the Clarke-Wright insertion algorithm [23] for delivery nodes only, and then applying the 1-INS algorithm [24] in order to insert pickup nodes one by one into the routes. The time complexity of the second one is O(n^4).

To get better initial solutions of VRPPD, this study suggested a novel way by introducing a two-phase heuristic routing algorithm which consists of a clustering phase and uses the geometrical center of a cluster and route establishment phase by applying a two-way search of each route after applying the TSP algorithm on each route. As the experimental result section shows, the suggested algorithm can generate better initial solutions than other existing methods such as the giant-tour-based partitioning method or the insertion-based method.

3. The Suggested Two-phase Heuristic Algorithm

In this section, the suggested two-phase heuristic algorithm with the notion of geometrical center to produce an initial solution will be explained in detail.

3.1 The notion of geometrical center

In our algorithm, we introduce the notion of a geometrical centre (GC) of a cluster as follows: Let $l_i = \{v_0, v_1, \cdots, v_k\}$ be cluster *i*, where v_j is a member of cluster *i*. GC of l_i can be defined as:

$$GC(l_i) = \left(\frac{\sum_{j=0}^{k} v_j^x}{k}, \frac{\sum_{j=0}^{k} v_j^y}{k}\right),$$ where v_j^x and v_j^y are *x* and *y* coordinates of a node v_j.

3.2 The first phase: make clusters satisfying delivery-feasibility and pickup-feasibility

The proposed algorithm works in the manner of cluster first and route second. The objective of the first phase is to make clusters that satisfy delivery-feasibility and pickup-feasibility. To achieve this objective, the proposed algorithm finds the farthest node from the depot among unclustered nodes as a seed for cluster generation. And then the geometric center (GC) of the cluster (if there is only a node in a cluster, the center of the cluster is the same as the location of the node) is calculated. Find the closest node from the center of the cluster among unclustered nodes and include the node into the cluster, if the inclusion does not violate delivery-feasibility and pickup-feasibility, and update the center of the cluster. If the inclusion of a node into the cluster violates either delivery or pickup feasibility, stop to expand the cluster and find the farthest node from the depot among unclustered nodes again and repeat the above processes until there are no unclustered nodes left. Table 1 shows the pseudo code of the first phase.

Table 1. Pseudo code of the first phase

```
set i = 1 and Q = vehicle_capacity
while (unvisited node exists)
  v_k = the farthest node among unclustered nodes from the depot
  make a cluster l_i with v_k
  set capacity of l_i^d = Q, l_i^p = Q
  add v_k to l_i
  if v_k needs delivery service, reduce l_i^d by q_k
  else reduce l_i^p by q_k
  calculate the GC(l_i)
  v_j = the closest node among unclustered nodes from the
      GC(l_i)
  while (TRUE)
     if v_j needs delivery service and l_i^d + q_j > Q
        break while loop
     if v_j needs pickup service and l_i^p + q_j > Q
        break while loop
     add v_j to l_i
     recalculate GC(l_i)
     if v_j needs delivery, reduce l_i^d by q_k
     else reduce l_i^p by q_k
     v_j = the closest node among unclustered from GC(l_i)
  end while
  i = i + 1
end while
```

3.3 The second phase: establish routes satisfying load- feasibility

The second phase is for route establishment and uses the result of the first phase since the result of the first phase satisfies the delivery-feasibility and pickup-feasibility. To build up the route that satisfies the load-feasibility, at first, the proposed algorithm applies the TSP (Traveling Salesman Problem) algorithm without discriminating delivery and pickup customers. In the case the TSP algorithm does not satisfy the load-feasibility, the suggested algorithm adopts another process to satisfy load-feasibility by skipping the nodes, which causes violation of load-feasibility, and visiting the skipped nodes when enough space for the demands of the skipped nodes is available. This process for satisfying load-feasibility causes longer travel distances. We apply this process in clockwise and counter-clockwise directions since the route of VRPPD is sensitive to its direction and we select the one that has shorter travel distance as the final result.

Figures 1 and 2 show an example of the process. The routing result of VRPPD is sensitive to the clockwise or counter-clockwise direction of the vehicle. The left side of Figure 1 depicts the result of applying the TSP algorithm on a cluster and the right side shows the result of satisfying the load-feasibility if the capacity of the vehicle is 7 tons and each demand of pickup and delivery is 1 ton. Figure 1 shows that the vehicle should detour in order to not violate the load-feasibility by skipping some nodes and that it should visit these nodes when space for the demands of the nodes is available. Figure 2 shows another routing result of visiting customers of reverse direction and in this case, satisfying the load-feasibility is not needed. We select the second one as the final result of the second phase since the travel distance of the second one is shorter than the first one.

Table 2 shows the pseudo code part of the second phase. (Full code of the second phase is just to change line 2 of the procedure with counter_clockwise_TSP(l_i) and compare the travel distance of each cluster and select the shorter one as the final result.)

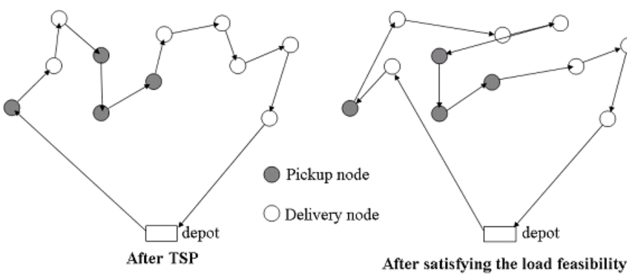

Fig. 1. Example of the route establishment process (Clockwise direction)

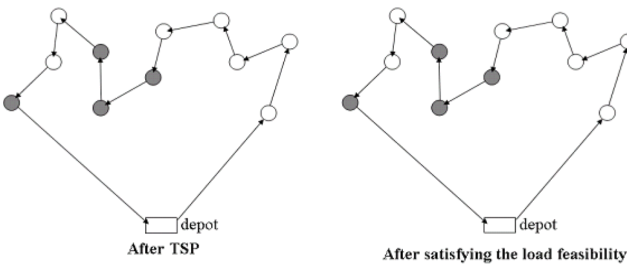

Fig. 2. Example of the route establishment process (Counter clockwise direction)

Table 2. Pseudo code of the second phase

```
for (i=1; i<=number of clusters; i=i+1)
  r_i =clockwise(TSP(l_i))
  for (j=1; j<=number of nodes in r_i ; j=j+1)
      if pickup or delivery demand of node r_i (v_j) violates the
            load-feasibility
         add r_i (v_j) to skipped_node_list
      else if available capacity of the vehicle >= sum of demands
            of skipped_node_list
         make a link from r_i (v_{j-1}) to the first node in
            skipped_node_list
         make a link from the first to the last node one by one in
            skipped_node_list
         make a link from the last node in skipped_node_list to
            r_i (v_j)
      else make a link from r_i (v_{j-1}) to r_i (v_j)
  end for
end for
```

3.4 Analysis of the suggested algorithm

The time complexity of the first phase is O(n^2) for finding the farthest node among unclustered nodes takes O(n) and repeat this process until there are no unclustered nodes left. The second phase adopts the Lin-Kernighan heuristic [19] to solve the TSP which has O($n^{2.2}$) of time complexity. So, if we let l be the number of clusters and k_l be the number of nodes in cluster l, the complexity of the second phase is O($lk_l^{2.2}$). If there is just a route (the worst case), then l is equal to 1 and k_l is equal to n and the time complexity goes to O($n^{2.2}$). As a result, the overall time complexity of the suggested algorithm is O(max(n^2,$n^{2.2}$)) =O($n^{2.2}$).

4. Experimental Results

The algorithms of this study were written in Java and executed on an IBM® PC of Intel® Core®2 Quad CPU 2.83GHz and 4GB RAM.

We tested our solution for the CVRP on the Augerat A, B, and P benchmark dataset [20]: for the instances in class A, both customer locations and demands were random. The instances in class B were clustered instances. The instances in class P were modified versions of instances from the literature. To generate test data for the case of mixed pickups and deliveries of three VRPPD problems for each VRP instance, every second (50%), fourth (25%) or tenth (10%) customer on the list was set to a pickup node and assigned a supply demand equal to the original delivery demand.

For the comparisons, we implemented two algorithms that are used frequently in VRPPD literatures. The first one is the giant-tour partition-based method on the concept of multiple giant-tours [21]. To make the results of the giant-tour partitioning load feasible, the method of reinserting the depot [22] is used. The giant-tour partition was used in the work of [17] and its time complexity is O(n^2). The second one is the insertion-based heuristic algorithm which first make routes by using the Clarke-Wright insertion algorithm [23] for delivery nodes only, and then applying the 1-INS algorithm [24] with α and β value as 1.5 and 1, respectively, in order to insert pickup nodes one by one into the routes. The time complexity of the second one is O(n^4).

Table 3 summarizes the results in terms of average routing cost, which are accumulated distances of all vehicles, and computational time, respectively. As Table 3 shows, the suggested algorithm gives the best results except for 10% of pickup nodes in A and B dataset. As averages, the suggested algorithm gives better solution qualities than giant-tour partition by 6%, 2% and 6% and insertion by 4%, 1% and 6% for each dataset, respectively. With regards to the computing time, since all of these methods have polynomial time complexities, all methods take less than a second on average as Table 3 shows. This means that these methods can be used before applying subsequent computer-intensive optimizing procedures without time related burdens.

5. Conclusion

In this study, we focused on generating an initial solution for the vehicle routing problem with pickups and deliveries (VRPPD). We suggested a two-phase heuristic algorithm which at first clusters routes by using the notion of the geometrical center of each router and then we applied the two-way search algorithm just after applying the TSP algorithm on the results of the first phase. The suggested algorithm shows better results over existing algorithms such as the giant-tour partition-based method and the insertion-based method. Based on the experimental results, the suggested method can be applied to generate initial solutions for VRPPD.

References

[1] Dantzig GB and Ramser JH (1959). The truck dispatching problem. Manage Sci 6: 80-91.

[2] Toth P and Vigo D (1997). An exact algorithm for the vehicle routing problem with backhauls. Transport Sci 31: 372-385.

[3] Mingozzi A, Giorgi S and Baldacci R (1999). An exact method for the vehicle routing problem with backhauls. Transport Sci 33:315-329.

[4] Deif I and Bodin L (1984). Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In: Kidder A (ed). Proceedings of the Babson conference on software uses in transportation and logistic management, Babson Park, FL, USA, pp 75-96.

[5] Goetschalckx M and Jacobs-Blecha C (1989). The vehicle routing problem with backhauls. Eur J Oper Res 42:39-51.

[6] Thangiah SR, Potvin J-Y and Sun T (1996). Heuristic approaches to vehicle routing with backhauls and time windows. Comput Oper Res 23:1043-1057.

[7] Duhamel C, Potvin J-Y and Rousseau J-M (1997). A tabu search heuristic for the vehicle routing problem with backhauls and time windows. Transport Sci 31:49-59.

Table 3. Experimental Results

		Solution quality (distance)			Computing time (milli second)		
		Giant-tour partition	Insertion	Centroid two-way	Giant-tour partition	Insertion	Centroid two-way
Augerat A	10%	1148.5	**1092.1**	1101.6	1.6	32.0	15.3
	25%	1085.8	1035.0	**1023.3**	1.8	41.4	19.7
	50%	1019.5	1045.5	**919.1**	2.0	88.3	25.2
Average		1084.6	1057.5	**1014.7**	1.8	53.9	20.1
Augerat B	10%	997.4	**960.7**	986.0	1.6	34.1	16.5
	25%	896.7	909.2	**876.2**	1.7	43.0	20.4
	50%	814.2	816.9	**787.7**	2.1	88.9	27.8
Average		902.8	895.6	**883.3**	1.8	55.3	21.6
Augerat P	10%	669.3	658.2	**636.3**	2.3	48.5	31.3
	25%	637.9	625.0	**604.8**	2.7	68.5	44.1
	50%	623.5	650.1	**574.7**	3.4	171.0	74.9
Average		643.6	644.4	**605.2**	2.8	96.0	50.1

[8] Toth P and Vigo D (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. Eur J Oper Res 113:528-543.

[9] Brandão J (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. Eur J Oper Res 173:540-555.

[10] Min H (1989). The multiple the vehicle routing problem with simultaneous delivery and pick-up points. Transport Res Part A 5:377-386.

[11] Dethloff J (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. Oper Res Spektrum 23:79-96.

[12] Dethloff J (2002). Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pickup applied to the vehicle routing problem with backhauls. J Oper Res Soc 53:115-118.

[13] Angelelli E and Mansini R (2002). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In: Klose A, Speranza MG, Van Wassenhove LN (Eds). Quantitative Approaches to Distribution Logistics and Supply Chain Management, Springer-Verlag, Berlin, pp. 249-267.

[14] Tang FA and Galvão RD (2002). Vehicle routing problems with simultaneous pick-up and delivery service. Opsearch 39:19-33.

[15] Tang FA and Galvão RD (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Comp Opns Res 33:595-619.

[16] Nagy G and Salhi S (1998). The many-to-many location-routing problem. TOP 6:261-275.

[17] Nagy G and Salhi S (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. Eur J Opl Res 162:126-141.

[18] Chen JF and Wu TH (2006). Vehicle routing problem with simultaneous deliveries and pickups. J Opl Res Soc 57:579 - 587.

[19] Lin S and Kernighan B (1973). An effective heuristic algorithm for the traveling salesman problem. Opns Res 21: 498-516.

[20] Augerat P (1995). Approche polyèdrale du problème de tournées de véhicules. PhD thesis, Institut National Polytechnique de Grenoble-Inpg. http://www.branchandcut.org/VRP/data/, accessed 1 April 2009.

[21] Salhi S, Sari M, Saidi D and Touati N (1992). Adaptation of some vehicle fleet mix heuristics. Omega 20, 653-660.

[22] Mosheiov G (1994). The travelling salesman problem with pick-up and delivery. Eur J Oper Res 79:299-310.

[23] Clarke G and Wright J (1964). Scheduling of vehicles from a central depot to a number of delivery points. Opns Res 12: 568-581.

[24] Nagy G and Salhi S (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. J Opl Res Soc 50: 1034 - 1042.

Kwangcheol Shin
He received his B.S. in 1996 and M.S. in 1998, both from ChungAng University, Korea. He also completed his Ph.D. degree in Computer Science and Engineering at the same University in 2004. Currently, he is a research professor of the Department of Information and Communications Engineering at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His research interests include RFID and Sensor Network.