A Lagrangian Heuristic for the Vehicle Routing Problems with the Private Fleet and the Common Carrier

Kuancheng HUANG
Associate Professor
Dept. of Transportation Technology and Management
National Chiao Tung University
No. 1001, Ta Hsueh Road, Hsinchu City 300
Taiwan
Fax: +886-3-5720844
E-mail: kchuang@cc.nctu.edu.tw

Cheng-Po HSU
Research Associate
Dept. of Transportation Technology and Management
National Chiao Tung University
No. 1001, Ta Hsueh Road, Hsinchu City 300
Taiwan
Fax: +886-3-5720844
E-mail: kevin74625.tem93@nctu.edu.tw

Abstract: The delivery of goods from a warehouse to local customers is an important decision problem of a logistics operator. As the demand is fluctuating, the common carrier may be needed if the private fleet is insufficient. In this study, we focus on the vehicle routing problem with private fleet and common carrier (VRPPC). In order to balance computational load and solution quality, a heuristic algorithm is developed based on several classical mathematical programming techniques. The VRPPC is first formulated in the form of the set covering problem, and the Lagrangian relaxation is used as the backbone in designing the iterative algorithm. In addition, a concept similar to column generation is used to update the solution space, a partial set of routes. Based on the numerical experiment, the solution quality of the heuristic algorithm is stable, and the computation time is acceptable under the current highly dynamic environment.

Key Words: vehicle routing problem, common carrier, set covering, Lagrangian relaxation

1. INTRODUCTION

The Vehicle Routing Problem (VRP) has been widely addressed in prior research works. In general, many companies operate their private fleet for delivering the shipments to the local customers with an aim to minimize the shipping costs and raise the service level. However, demand is fluctuating in reality, and it thus is difficult to determine the size of private fleet. If the fleet is too large, some assets become idle when the demand is relatively low. On the contrary, if the fleet size can not deal with the high demand, some customers may not be served in time.

Therefore, it is necessary to integrate the usage of the private fleet and the common carriers to minimize the shipping cost. This paper studies the Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC). It determines whether each customer should be served by the private fleet or by the common carrier when companies can not accommodate all demand by the private fleet only.

This paper develops a heuristic algorithm that can generate a good solution for the VRPPC within a reasonable time. The heuristic algorithm is based on several mathematical programming techniques, including Set Covering Problem (SCP), Lagrangian Relaxation (LR), and Column Generation (CG).
Laporte and Semet (2001) stated that future research of the VRP steers toward the algorithms that are simpler with lighter computation load and applicable to large scale problems with a more general problem structure, even if the precision of solutions can be sacrificed to some extent. Thus, the focus of this paper is the algorithm efficiency for the VRPPC, and the goal is to obtain good approximate solutions within a reasonable time.

2. LITERATURE REVIEW

The introduction and the literature review of the VRPPC are presented in the first and second section. Then, the models and solution techniques used in developing the solution algorithm in this paper are provided in the third section.

2.1 Brief introduction to VRPPC

Chu (2005) pointed out that when demand exceeds the capacity of the private fleet, a company has to consider using the common carriers to handle the surging demand. From the perspective of a distribution center, an integer programming model was formulated considering the choice of private vehicles in various types and common carriers. Besides, the saving heuristic was developed to find the solution. Since then, the defined problem was referred to as the vehicle routing problem with private fleet and common carriers (VRPPC) by researchers. Table 1 provides the summary of the differences between the classic capacitated vehicle routing problem (CVRP) and the VRPPC.

<table>
<thead>
<tr>
<th></th>
<th>CVRP</th>
<th>VRPPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>The company provides service by the private fleet only.</td>
<td>The company serves the customers by choosing from the private fleet and the common carriers simultaneously.</td>
<td>The company has the opportunity to reduce the cost by employing the outside common carriers.</td>
</tr>
<tr>
<td>The operation cost is probably higher due to the lack of responsiveness to demand variation.</td>
<td>The company has the opportunity to reduce the cost by employing the outside common carriers.</td>
<td>The problem definition is closer to reality, but the related literature is limited.</td>
</tr>
<tr>
<td>The problem definition is simpler, and the related literature is abundant.</td>
<td>The problem definition is closer to reality, but the related literature is limited.</td>
<td>The feasible solution usually exists given the service capability provided by the common carriers.</td>
</tr>
<tr>
<td>The feasible solution does not exist when the capacity is insufficient.</td>
<td>The feasible solution usually exists given the service capability provided by the common carriers.</td>
<td></td>
</tr>
</tbody>
</table>

2.2 Brief introduction to VRPPC

According to Fisher (1995), the development of the VRP algorithms can be divided into three stages. The first stage is from 1960 to 1970, and the developed heuristics are simple, including various local improvement methods and greedy approaches, such as the Saving Algorithm, the Exchange Algorithm, and the Sweep Algorithm. The second stage is from 1970 to 1980, the heuristics were mainly based on mathematical programming: for instance, the Branch and Bound Method, the Lagrangian Relaxation, the Cutting Plane Method, and the Column Generation. The third stage is from 1990 to the present, the developed heuristics incorporate
the concepts of artificial intelligent, and many algorithms have been developed, such as the Tabu Search, the Simulated Annealing, the Genetic Algorithm, and the Great Deluge Algorithm. In general, the performance of these so-called meta-heuristics is better than the heuristics developed in the early stages. Regarding the VRP related algorithms, Laporte et al. (2000), Cordeau et al. (2002) and Toth and Vigo (2002) serve as excellent references.

The prior research works for the VRPPC are very few. In particular, to the best of our knowledge, there is no work for the exact-solution algorithm. Chu (2005) developed a saving heuristics to solve the VRPPC. Bolduc et al. (2007) improved the heuristic of Chu (2005) to design the SRI (Selection, Routing and Improvement) heuristics, in which two more sophisticated initial solutions with better customer exchange and a 4-opt procedure are applied to improve the search solutions. Afterwards, other researchers mainly focused on the meta-heuristics. Bolduc et al. (2008) revised one constraint in Chu (2005) to relax the requirement of the full utilization of the private fleet. In addition, they developed the RIP (Randomized construction Improvement Perturbation) heuristics by modifying the SRI heuristic based on an HVRP (Heterogeneous Vehicle Routing Problem) modeling. Given the VRPPC model in Bolduc et al. (2008), Cote et al. (2009) developed a Tabu Search (TS) heuristic to solve this problem under the context of homogeneous vehicles. The initial solutions are obtained by the Least-cost Insertion Method and improved by the TS procedure. Euchi and Chabchoub (2010) further modified the TS algorithm of Cote et al. (2009) for the version of the VRPPC with multiple types of vehicles.

Cordeau et al. (2002) highlights that there are four important issues while designing or comparing heuristics: accuracy, speed, simplicity and flexibility. Although the solutions generated by meta-heuristics can be very close to optimal, the main concerns are their implementation complexity and computation time. Thus, we attempted to find a balance among the multiple goals and developed a heuristics based on mathematical programming, in which the formulation of the Set Covering Problem (SCP), the technique of the Lagrangian Relaxation (LR), and the concept of Column Generation (CG) are used. In particular, if flexibility is critical from the operational point of view, the set covering approach can be a suitable alternative due to its flexibility for modeling operational considerations and constraints.

2.3 Review of Modeling and Solution Techniques

The first sub-section describes the VRP formulation based on the SCP and discusses the advantages and limitation of this formulation. The second sub-section explains why and how the concept of CG can be incorporated into the solution algorithm for the VRP. The third sub-section provides the literature review of prior research works that combine the CG and the LR, one solution technique proved to be an effective approach for solving the SCP.

2.3.1 Set Covering Model

There have been many solution algorithms developed for the SCP due to its wide applications. For instance, Balas and Carrera (1996) adopted the approach of branch and bound with the integer constraint relaxed. Beasley (1990) and Caprara et al. (1999) used the Lagrangian relaxation, which keeps the binary variables and uses the sub-gradient method to update the Lagrangian multipliers. The SCP formulation for the VRP is as follows.
\[ \text{Min} \sum_{r \in R} c_r x_r \]  
\[ \sum_{r \in R} a_{ir} x_r \geq 1 \quad \forall i \in I \]  
\[ x_r : \text{Binary} \quad \forall r \in R \]  

- \( i \): index for customers \((i = 1 \text{ to } n; \ n \) is customer number; \( I \) is set of all customers.)
- \( r \): index for routes \((R \) is set of all routes.)
- \( c_r \): cost of route \( r \)
- \( a_{ir} \): binary constant; \( a_{ir} = 1 \) if customer \( i \) is in route \( r \), and \( a_{ir} = 0 \) if otherwise.
- \( x_r \): binary decision variable representing that route \( r \) is selected.

The objective function (1) is to minimize the total expense by combining the costs of the selected routes (sets). Constraint (2) ensures that each customer is incorporated in at least one of the selected routes. The greater-or-equal-to sign is used in (2), instead of the equal sign. The advantage of choosing the SCP rather than the set partitioning problem (SPP) is that the SCP is generally easier to solve than the SPP. However, as it is possible that one customer can be included in multiple selected routes, extra steps must be taken to correct this situation. Finally, in Constraint (3), the binary variable \( x_r \) represents the selection of the routes.

Although many types of VRPs can be formulated by this flexible SCP formulation, it is impossible to enumerate all of the routes for large scale problems. Thus, many researches have used the technique of column generation to deal with this issue.

2.3.2 Column Generation Approach for VRP

Column Generation (CG), also known as Dantzig-Wolfe Decomposition, is a linear relaxation approach. It employs the duality theory of linear programming to generate the variables (the columns). CG can avoid enumerating the numerous variables, most of which are unlikely to be in the solution. The solution procedure of CG involves the primal problem and the sub-problem. The former is basically the SCP formulation, but the integer constraint relaxed as (6).

\[ \text{Min} \sum_{r \in R} c_r x_r \]  
\[ \sum_{r \in R} a_{ir} x_r \geq 1 \quad \forall i \in I \]  
\[ x_r \geq 0 \quad \forall r \in R \]  

\( R \) should be the set including all possible routes for serve the customers in set \( I \). However, it is impossible to enumerate all routes to derive the \( a_{ir} \)'s and calculate its corresponding cost \( c_r \). Suppose the dual variable of the constraint in (5) is denoted by \( \pi_i \) \((i = 1 \text{ to } n) \). Based on the dual feasibility, if the least reduced cost in the sub-problem is positive among all remaining columns, the optimal solution for the primal problem is found. Otherwise, the remaining column (variable) that reduces the objective value of sub-problem the most is added to the primal problem. The design of the sub-problem can be problem specific. For example, for the master problem of (4) to (6), Agawal et al. (1989) designed the sub-problem as follows.

\[ \text{Min} \quad f(y) - \sum_{ic} \pi_i \cdot y_i \]  

647
In the objective function (7), \( f(y) \) is the optimal cost of the TSP (Traveling Salesman Problem) for the route \( y \). Constraint (8) is the vehicle capacity limitation. Constraint (9) shows that \( y_i \) is a binary decision variable. However, in general, \( f(y) \) is difficult to find. Agawal \textit{et al.} (1989) replaced it by a linear approximation \( f(y) = g \cdot y \), where \( g \) is the vector formed by \( g_i \) (\( i=1 \) to \( n \)), the linear estimate corresponding to the cost of serving customer \( i \). The original sub-problem can then be transformed as a knapsack problem, and it is solved by the branch and bound method. The solution of the primal problem can be found repeatedly by this process until the sub-problem cannot generate a new variable. If the final solution of the relaxed problem happens to be an integer solution, it is also the optimal to the original VRP. Otherwise, a branch-and-bound procedure is needed to obtain the integer solutions.

Although CG overcomes the difficulty of enumerating all routes and is capable of producing the optimal solution, the solution procedure is relatively complicated. The sub-problem (7) to (9) is especially arduous to solve. In addition, the optimal solution of the primal problem with the integer constraint relaxed is often fractional and thus infeasible. The extra branch-and-bound is needed to find the optimal integer solution. Therefore, we try to establish and update a solution space, a partial set of routes, with a concept similar to CG. However, instead of relaxing the binary constraint, we develop a solution procedure based on the Lagrangian relaxation, which is a method proven to be effective for the SCP and relaxes the coverage constraint instead in the SCP formulation.

2.3.3 The Heuristics Combining CG and LR

The column generation and the Lagrangian relaxation have been integrated into one solution algorithm in several prior research works. Desaulniers \textit{et al.} (2005) in their book presented and discussed the algorithm structure of combining CG and LR. In general, CG is used as the backbone in the solving process, and the Simplex dual variables are approximated by the Lagrangian Multipliers in the sub-problems to raise the efficiency of the solution procedure. One example with this approach is Jans and Degraeve (2004), which developed a heuristics to cope with the Capacitated Lot-Sizing Problem (CLSP).

There are very few research works applying the heuristics combining CG and LR to the VRP, especially for the case that LR is the main method in terms of the algorithm structure. Huang and Wu (2007) developed a heuristics based on LR to the CVRP. In their algorithm, only a partial set of the feasible routes is initially generated and, according to a concept of CG, adjusted through the iterative LR solution procedure to find the approximate solution for the SCP formulation.

Following the idea in Huang and Wu (2007), we transformed the VRPPC into a SCP model and applied the Lagrangian relaxation to solve this problem. In particular, we modified the LR
algorithm developed by Caprara et al. (1999) to obtain the feasible solution for the VRPPC. In addition, in order to avoid dealing with the sub-problem for route generation, we used the concept of the insertion heuristic developed by Qureshi et al. (2010), who developed a CG-based exact algorithm for the VRPTW (Vehicle Routing Problem with Time Window).

3. MATHEMATICAL MODEL AND SOLUTION ALGORITHM

The mathematical formulation of the VRPPC is presented in the first section. The Lagrangian relaxed model and the structure of the whole solution algorithm is provided in the second section, and the building blocks of the solution algorithm, including the adjustment of the Lagrangian multipliers, the determination of the VRPPC feasible solution, and the update of the solution space, are explained in the subsequent sections.

3.1 Formulations of VRPPC

This paper follows the mathematical model of the VRPPC defined by Bolduc et al. (2008), but only the case of homogeneous vehicles is considered. The primary premises of the model are listed as follow.

- Each customer is served exactly once. Its demand is satisfied by one vehicle, and no split shipping is allowed.
- Only one depot is considered.
- Only the delivery activity (or the pick-up, but not a mix of both) is considered.
- The total demand of each route is limited by the capacity of the vehicle.
- The number of customers ($n$) and the number of private vehicles ($m$) are given.
- There is no capacity limitation for the common carrier and the outsourcing cost is given.
- When the fleet vehicle is used, a fixed cost is incurred.

Based on the above problem assumptions and the following notations, the VRPPC can be formulated as the following MIP (Mixed Integer Programming) problem.

$$
\min \sum_{k=1}^{m} f_k Y_{0k} + \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{m} C_{ijk} X_{ijk} + \sum_{i=1}^{n} p_i z_i
$$

$$
\sum_{j=0}^{n} \sum_{k=1}^{m} X_{0jk} = \sum_{i=1}^{n} \sum_{k=1}^{m} X_{i0k} \leq m
$$

$$
\sum_{j=0}^{n} X_{hjk} = \sum_{i=0}^{n} X_{ihk} \leq Y_{hk} \quad \text{for } h = 0 \cdots n, k = 1 \cdots m
$$

$$
\sum_{k=1}^{m} Y_{ik} + z_i = 1 \quad \text{for } i = 1 \cdots n
$$

$$
\sum_{i=1}^{n} d_{i} Y_{ik} \leq Q \quad \text{for } k = 1 \cdots m
$$

$$
\sum_{i,j \in S} X_{ijk} \leq |S| - 1 \quad \text{for all } S, k = 1 \cdots m
$$

$$
X_{ijk}, Y_{ik}, Z_i : \text{binary, for } i, j = 0 \cdots n, k = 1 \cdots m
$$
\(i\): index for customers \((i = 1 \text{ to } n, 0 \text{ represents the depot.})\)
\(d_i\): demand of customer \(i\)
\(p_i\): the cost of assigning customer \(i\) to the common carrier
\(k\): index for vehicles \((k = 1 \text{ to } m)\)
\(f_k\): fixed cost of vehicle \(k\)
\(Q\): capacity limitation of the vehicle
\(C_{ij}\): cost for moving from customer \(i\) to customer \(j\)
\(X_{ijk}\): binary variable representing vehicle \(k\) directly moving from customer \(i\) to \(j\).
\(Y_{ik}\): binary variable representing customer \(i\) is served by vehicle \(k\).
\(Y_{0k}\): binary variable representing vehicle \(k\) is used.
\(Z_i\): binary variable representing customer \(i\) is served by the common carrier.

The objective function is to minimize the total costs, including the fixed cost and the routing cost of the used private vehicles and the outsourcing cost associated with the common carriers. Constraint (11) ensures that the origin and destination of the private vehicle routes is the depot, and it is not necessary to assign all vehicles. Constraint (12) states that if the customer is served by the private fleet, it can be only served once with the same vehicle. Constraint (13) shows that the customer is either served by the private fleet or the common carrier. Constraint (14) states that the overall load of the vehicle can not exceed its capacity. Constraint (15) is to prevent sub-tour occurrence, where \(S\) is the subset of customers forming the sub-tour. This mathematical model is mainly used to illustrate the problem features and cannot be solved directly due to the difficulty of eliminating the sub-tours. Thus, the VRPPC is transformed into the following SCP-based formulation, which is very similar to the model of (1) to (3) but has the extra notations \(p_i\) and \(z_i\) representing the linear cost and the binary decision for outsourcing customer \(i\).

\[
\text{Min} \sum_{r \in R} c_r x_r + \sum_{i=1}^{n} p_i z_i \\
\sum_{r \in R} a_{ir} x_r + z_i \geq 1 \quad \forall i \in I \setminus \{0\} \\
\sum_{r \in R} x_r \leq m \quad (19) \\
x_r, z_i : \text{Binary} \quad \forall r \in R, \ i = 1 \cdots n
\]

The objective function (17) is to minimize the routing costs of the private fleet plus the expense charged by the common carriers. Constraint (18) ensures that a customer is served by the private fleet or assigned to the common carriers. Constraint (19) specifies that the assignments can not exceed the total number of the private vehicles. Finally, in Constraint (20), the decisions \(x_r\) and \(z_i\) are set to be binary variables.

The major limitation of the SCP model is that extremely heavy computational load is needed to find all possible routes. In this study, we establish and update a solution space, a partial set of routes, and apply the concept of CG to design a procedure for adding promising routes. Besides, as SCP is still an NP-hard problem, we relax the customer coverage constraint to develop a heuristic algorithm based on the Lagrangian relaxation.
3.2 Lagrangian Relaxation Model and Algorithm Structure

Constraint (18) is relaxed to form the LR model (21) to (24), in which \( \lambda_i \) is the Lagrangian multiplier for customer \( i \) (\( \lambda \) is the vector formed by \( \lambda_i \)), and its value must be non-negative. In particular, \( \lambda_i \) can be interpreted as the cost of serving customer \( i \) from the viewpoint of the dual price. The Lagrangian cost of a route is denoted as \( c_r(\lambda) \) and can be computed as (22), in which \( I_r \) represents the set of customers included in route \( r \), that is, \( I_r = \{i \in I : a_{ir} = 1\} \).

\[
L(\lambda) = \min \sum_{r \in R} c_r(\lambda)x_r + \sum_{i \in I}(p_i - \lambda_i)z_i + \sum_{i \in I} \lambda_i
\tag{21}
\]

\[
c_r(\lambda) = c_r - \sum_{i \in I_r} \lambda_i \quad \forall r \in R
\tag{22}
\]

\[
\sum_{r \in R} x_r \leq m
\tag{23}
\]

\[
x_r, z_i: \text{Binary} \quad \forall r \in R, \ i = 1 \cdots n
\tag{24}
\]

The integer programming model (21) to (24) is fairly easy to solve. For a given set of the Lagrangian multipliers, all the routes with negative \( c_r(\lambda) \) are selected, i.e., \( x_r = 1 \) for \( c_r(\lambda) \leq 0 \). In addition, customer \( i \) is assigned to the common carriers if the corresponding \( (p_i - \lambda_i) \) is negative. However, as the solution space does not contain all routes, the value of \( L(\lambda) \) is not the lower bound to the original problem (10) to (16).

Based on the LR model, the key issues are how to determine the Lagrangian multipliers, how to derive the feasible solution from the relaxed solution, and how to update the solution space so as to generate a good approximate solution. The flowchart of this LR-based iterative solution algorithm is shown in Figure 1, and the procedures related to these three key issues are described in the subsequent sections.

3.3 Updating the Lagrangian Multipliers

For each of the iteration in the LR-based solution algorithm, the Lagrangian multipliers are updated based on a sub-gradient method, which is similar to the one shown in (25) and (26) proposed by Held and Karp (1970).

\[
\lambda_i^{t+1} = \max \left\{ \lambda_i^t + \alpha \frac{UB - L(\lambda^t)}{s(\lambda^t)} s_i(\lambda^t), 0 \right\} \quad \forall i \in I
\tag{25}
\]

\[
s_i(\lambda^t) = 1 - \sum_{r \in R_i} x_r(\lambda^t) \quad \forall i \in I
\tag{26}
\]

\( R_i \) represents the set of routes containing customer \( i \), that is, \( R_i = \{r \in R : a_{ir} = 1\} \).

In (25), \( UB \) and \( L(\lambda^t) \) are the upper and lower bounds found so far, and the Lagrangian multipliers are updated with the step size adjusted according to the difference between the two bounds. In addition, \( \alpha \) is a pre-determined step-size parameter, which can be tuned dynamically as well. The direction \( s_i(\lambda^t) \) is determined by (26), in which the number of times for customer \( i \) to be covered is computed based on the routes with \( c_i(\lambda) < 0 \). If customer \( i \) is covered exactly once, \( s_i(\lambda^t) = 0 \), and no adjustment for the corresponding \( \lambda_i \) is needed. We
change the original sub-gradient method (25) to (27) and (28) so as to cope with the different SCP model encountered in the study.

\[
\lambda_{i}^{t+1} = \max \left\{ \lambda_{i}^{t} + \alpha^{t} s_{i}(\lambda^{t}), 0 \right\} \quad \forall i \in I
\]

\[
\alpha^{t+1} = \begin{cases} 
0.5\alpha^{t} & \text{if } \sum_{i} |s_{i}(\lambda^{t})| - \sum_{i} |s_{i}(\lambda^{t+1})| \leq 0 \\
\alpha^{t} & \text{otherwise}
\end{cases}
\]

As the solution space only contains a limited number of routes, the lower bound \(L(\lambda^{t})\) is good for the current solution space, not for the original VRPPC, the step size is changed to depend only on the parameter \(\alpha\) as shown in (27). In addition, based on \(\sum |s_{i}(\lambda^{t})|\), which is the sum of the absolute error for coverage, we modify the value of \(\alpha\) as (28). If \(\sum |s_{i}(\lambda^{t})|\) is smaller than that of the previous iteration, we half the step size with a aim to further reduce this absolute error sum.

In addition, the determination of the direction \(s_{i}(\lambda^{t})\) in (26) is modified by following idea proposed by Caprara et al. (1999). The concern is that with the increasing number of routes in the solution space, the number of routes with \(c_{i}(\lambda^{t})<0\) may increase dramatically. That can cause the step size adjustment unstable. Caprara et al. (1999) developed the following

---

**Figure 1 Flowchart of the solution algorithm**

1. Generate the initial solution space
2. Set the initial Lagrangian Multipliers

**Update Lagrangian Multipliers (Sec. 3.3)**
1. Solve the Lagrangian relaxed problem
2. Update \(\lambda\) by executing the sub-gradient method for a fixed number of times

**Derive VRPPC feasible solution (Sec. 3.4)**
1. Determine the routes of the private fleet
2. Deal with the decisions related to the usage of the common carriers

**Update solution space (Sec. 3.5)**
Apply the insertion heuristics to add the routes of the private fleet to the solution space

Meeting stopping criterion

Stop

Yes

No
procedure to find the critical routes, which play a more important role for affecting the Lagrangian multipliers.

- Sort the routes with \( c_r(\lambda) < 0 \) by the value of \( c_r(\lambda) \) in a decreasing manner to form a subset \( R' \) of the solution space.
- The customers covered by the routes in \( R' \) are denoted by the set \( I(R') \).
- Based on the order in \( R' \), delete the route if \( I(R') \) remains the same after doing so.
- Replace \( R \) with \( R' \) in (26) for calculating the direction.

The adjustment procedure of the modified (26) for \( s_i(\lambda_t) \) and the (27) and (28) for \( \alpha \) are executed for a fixed number (denoted by \( T \)) of times before the values of the Lagrangian multipliers are used as the input for the next stage for finding the VRPPC feasible solution.

### 3.4 Solving the Feasible VRPPC Solution

The feasible solution always exists for the VRPPC because the non-covered customers, due to the relaxation of the coverage constraint, can be consigned to the common carrier. In addition, for cost minimization, the customers served more than once by the private vehicles should be adjusted for being served exactly once. However, the design of the correction procedure can not be too complicated so as not increase the computation load too much. The correction procedure includes two parts. First, we deal with the routes of the private fleet. Second, we determine whether a customer should be assigned to the common carrier or not.

#### 3.4.1 The Heuristics for the Routes of the Private Fleet

We modified the LR solution algorithm for the SCP in Caprara et al. (1999). For solving of the Lagrangian relaxed model in (21) to (24), the routes with \( c_r(\lambda) < 0 \) should be selected. In fact, \( c_r(\lambda) \) of a route can be thought as one kind of score, which is determined as (22) by subtracting the sum of Lagrangian multipliers from the corresponding routing cost. A route is more promising and likely to be selected if its score is more negative.

This concept works well for the SCP derived from the scheduling problem from the rail industry in Caprara et al. (1999). However, for the VRP faced in this study, a good solution cannot be found without considering the geographical location of the customers, as the routes selected at the beginning of the procedure have a strong influence on the subsequent route selection. For example, suppose customers are uniformly distributed around the logistics center, and the routes of four vehicles are about to be determined. If, based on the corresponding route scores of \( c_r(\lambda) \), the first three selected routes happen to have a radial of the angle of 0°, 120°, and 240° respectively, it is then almost impossible to find the fourth route that can cover all of the customers and form a good overall solution. Thus, we used the sweep concept and tried to assign the routes to the customers in an orderly fashion. In particular, the design of the procedure also needs to accommodate the case when customers are not uniformly distributed.

First, the polar coordinates of the customers with respect to the depot are denoted by \( \phi \), and they are sorted in an increasing order so as to form a function denoted by \( f(u) \), which links the relation between the angle and the accumulated customers. Figure 2 is the illustrative diagram for \( f(u) \). As shown in Figure 2, the angle is from 0 to \( 2\pi \), and the accumulated number of customers is naturally \( n \).
Suppose $\theta_1$ represents the exact angle (based on the center of mass for the served customers) of the first selected route. The expected number of customers to be served for the second route should be $n/m$, which can be used to determine $\varphi_2$, the expected angle of the second vehicle, as shown in Figure 2. The same concept can be extended to the rest of the vehicles, and the details are summarized as the following procedure.

[Step 1] For the first vehicle, set $k = 1$ and randomly generate the angle $\varphi_1$.

[Step 2] Compute the modified Lagrangian multipliers ($\tau_i$) and the biased route score ($\rho_r$) based on (29) and (30)
\[
\tau_i = \lambda_i \left( 1 - \frac{\varphi_i - \varphi_k}{2\pi} \right) \quad \forall i \in I \tag{29}
\]
\[
\rho_r = c_r - \sum_{i \in I} \tau_i \quad \forall r \in R \tag{30}
\]

[Step 3] Select the $k$th route based on the route with the most negative biased route score $\rho_r$, and compute the corresponding route angle $\theta_k$. In addition, set the Lagrangian multipliers ($\lambda_i$) of the customers in this route to be 0 to make them not attractive in the selection of the rest of the routes.

[Step 4] If $k = m$, terminate the procedure as the routes of all vehicles are determined. Otherwise, set $k = k+1$, and compute the expected angle for the next vehicle $\varphi_k$ based on (31) and go to [Step 2].
\[
\varphi_k = f^{-1} \left( f(\theta_{k-1}) + \frac{n - f(\theta_{k-1})}{m - k + 1} \right) \tag{31}
\]

3.4.2 The Adjusting Procedure for the Common Carriers

The customers not covered in the above-mentioned heuristic in general can be assigned to the common carriers. However, in order to minimize the total cost, the following adjustment procedure is conducted, which also examines the possibility of assigning the outsourced customers to the route of a private vehicle.

[Step 1] Calculate and sort the cost improvements of inserting the outsourced customers to the nearby existing routes of the private fleet without considering the vehicle capacity limitation.

[Step 2] Focus on the most promising outsourced customer and try to insert in into the private fleet routes. If the total demand of the route exceeds the vehicle capacity,

---

**Figure 2 Illustrative diagram for the expected angles of the routes**
one existing customer of this route is deleted if it has the least deletion cost, which is computed by adding outsourcing cost and subtracting the corresponding routing cost. Keep the deletion process until the capacity constraint is met. If the next effect of this customer insertion is positive, perform the insertion and go to [Step 3]. Otherwise, terminate the procedure.

[Step 3] If one or more customers become un-covered due to the deletion, go to [Step 1]. Otherwise, go to [Step 2] to perform the next insertion of an un-covered customer.

3.5 Solution Space Initialization and Update

The concept of sweeping is used to group the customers to create the initial routes. The polar coordinates of the customers with respect to the deport ($\phi$) are first sorted in an increasing order. Starting from the first node, every $g$ (a fixed parameter) customers are grouped, and the associated optimal TSP route is determined by enumerating all possible sequences of customers. The same process is repeated for ($g$-1) times with the starting point changing from the second node to the ($g$-1)$th node. Thus, a number of $n$ routes with the length of $g$ can be created.

Regarding the process of add the new routes to the solution space, an insertion procedure is used. The PFIH (Push Forward Insertion Heuristic) proposed by Solomon (1987) is the first insertion heuristic designed for the VRP algorithm. Qureshi et al. (2010) modified that heuristics to generate the new route for a VRP with soft time windows to avoid solving the CG sub-problems. This study further modified the procedure in Qureshi et al. (2010) to fit the context of the VRPPC. The routes in the solution space are sorted based on the Lagrangian score $c_r(\lambda)$ in an increasing order. One route out of the first $n$ (number of customers) is randomly selected to perform the following insertion process.

For each customer not included in the selected route, the steps explained below are performed to find the best node to be inserted and its corresponding inserting position.

[Step 1] Suppose a node not included in the selected route is denoted by $\delta$. For each possible inserting position (the preceding and following nodes are denoted by $a$ and $b$ respectively), the increase of the Lagrangian cost is computed as $c(a, \delta, b) = (d_{a,\delta} + d_{\delta,b} - d_{a,b})$, where $d_{i,j}$ is $(C_{i,j} - \lambda_j)$, that is the distance between two nodes minus the Lagrangian multiplier of the leaving node. The inserting position with the least increase is the best position for that node.

[Step 2] By comparing the increase of the Lagrangian cost of all nodes not included in the selected route, the best node for insertion is determined.

[Step 3] Repeat the previous two steps and generate the route with one more customer if the capacity limit is not violated.

The route generation steps are performed for $m/2$ times so as to generate about $m$ (number of private vehicles) new routes. However, while executing the subsequent insertion, the Lagrangian multipliers of the nodes that have been inserted are set to be zero so as to increase the chance for selecting the routes not containing the inserted nodes, based on (22) for $c_r(\lambda)$ determination.
4. NUMERICAL EXPERIMENT

Bolduc et al. (2008) have designed the VRPPC benching marking problems, some of which were used in the numerical experiment in this study. In general, the cost for the private vehicle between two nodes ($C_{ij}$) is basically the distance between them. The capacity of the whole private fleet is only about 80% of the total demand, so it is required to use the common carrier in order to serve all customers. The outsourcing costs of the customers are set at three levels based on their distances to the depot. The related information of the test problems is shown in Table 2.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Number of customers</th>
<th>Number of vehicles</th>
<th>Vehicle capacity</th>
<th>Fixed cost of each vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE-01</td>
<td>50</td>
<td>4</td>
<td>160</td>
<td>120</td>
</tr>
<tr>
<td>CE-02</td>
<td>75</td>
<td>9</td>
<td>140</td>
<td>100</td>
</tr>
<tr>
<td>CE-03</td>
<td>100</td>
<td>6</td>
<td>200</td>
<td>140</td>
</tr>
<tr>
<td>CE-04</td>
<td>150</td>
<td>9</td>
<td>200</td>
<td>120</td>
</tr>
<tr>
<td>CE-05</td>
<td>199</td>
<td>13</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>CE-06</td>
<td>50</td>
<td>4</td>
<td>160</td>
<td>140</td>
</tr>
<tr>
<td>CE-07</td>
<td>75</td>
<td>9</td>
<td>140</td>
<td>120</td>
</tr>
<tr>
<td>CE-08</td>
<td>100</td>
<td>6</td>
<td>200</td>
<td>160</td>
</tr>
<tr>
<td>CE-09</td>
<td>150</td>
<td>10</td>
<td>200</td>
<td>120</td>
</tr>
<tr>
<td>CE-10</td>
<td>199</td>
<td>13</td>
<td>200</td>
<td>120</td>
</tr>
<tr>
<td>CE-11</td>
<td>120</td>
<td>6</td>
<td>200</td>
<td>180</td>
</tr>
<tr>
<td>CE-12</td>
<td>100</td>
<td>8</td>
<td>200</td>
<td>120</td>
</tr>
<tr>
<td>CE-13</td>
<td>120</td>
<td>6</td>
<td>200</td>
<td>260</td>
</tr>
<tr>
<td>CE-14</td>
<td>100</td>
<td>7</td>
<td>200</td>
<td>140</td>
</tr>
</tbody>
</table>

Euchi and Chabchoub (2010) provided the computational results for the available VRPCC algorithms, which are the SRI (Selection, Routing and Improvement) heuristics in Bolduc et al. (2005), the RIP (Randomized construction Improvement Perturbation) heuristics in Bolduc et al. (2008), and the TS (Tabu Search) heuristic developed by themselves. Those results were used to compare the performance of the developed LR algorithm. The averages of the objective and the computation time are listed in Table 3. In addition, the best result out of 10 runs for the algorithms are shown in Table 4, in which the values in boldface are the best solution ever found.

Based on the results in the two tables, the developed LR algorithm should be applicable to the real operational environment as its solution quality is not too bad, and the computation time is acceptable. However, there is a significant gap between its performance and that of two meta-heuristics, the RIP heuristics from Bolduc et al. (2008) and the TS heuristic from Euchi and Chabchoub (2010). In terms of solution quality, the average gap is 2.72% for the 14 test problems, but the solution quality is degraded when the problem size is increased. As for the computation time, the developed LR heuristic is slightly faster than the two latest meta-heuristics. As the CPU used by Euchi and Chabchoub (2010) is Intel Xeon 3.6GHz, which is faster than Intel E8400 3.0GHz used in this study. In addition, the computer program of this study was coded by MATLAB, whose efficiency is usually lower than that of the C++ language used in Euchi and Chabchoub (2010). Thus, the computation time advantage of the developed LR heuristics should be larger than the one currently observed in Table 3.
### Table 3 Average cost and average computation time of various algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CE-01</td>
<td>1199.99</td>
<td>0</td>
<td>1132.94</td>
<td>25</td>
</tr>
<tr>
<td>CE-02</td>
<td>1890.33</td>
<td>0</td>
<td>1835.76</td>
<td>73</td>
</tr>
<tr>
<td>CE-03</td>
<td>2050.33</td>
<td>1</td>
<td>1959.65</td>
<td>107</td>
</tr>
<tr>
<td>CE-04</td>
<td>2694.72</td>
<td>1</td>
<td>2545.72</td>
<td>250</td>
</tr>
<tr>
<td>CE-05</td>
<td>3228.67</td>
<td>3</td>
<td>3172.22</td>
<td>474</td>
</tr>
<tr>
<td>CE-06</td>
<td>1282.94</td>
<td>0</td>
<td>1208.33</td>
<td>25</td>
</tr>
<tr>
<td>CE-07</td>
<td>2092.32</td>
<td>0</td>
<td>2006.52</td>
<td>71</td>
</tr>
<tr>
<td>CE-08</td>
<td>2163.32</td>
<td>1</td>
<td>2082.75</td>
<td>110</td>
</tr>
<tr>
<td>CE-09</td>
<td>2526.82</td>
<td>1</td>
<td>2443.94</td>
<td>260</td>
</tr>
<tr>
<td>CE-10</td>
<td>3511.02</td>
<td>3</td>
<td>3464.90</td>
<td>478</td>
</tr>
<tr>
<td>CE-11</td>
<td>2375.71</td>
<td>1</td>
<td>2333.03</td>
<td>195</td>
</tr>
<tr>
<td>CE-12</td>
<td>2037.54</td>
<td>0</td>
<td>1953.55</td>
<td>128</td>
</tr>
<tr>
<td>CE-13</td>
<td>2916.21</td>
<td>1</td>
<td>2864.21</td>
<td>188</td>
</tr>
<tr>
<td>CE-14</td>
<td>2220.77</td>
<td>1</td>
<td>2224.63</td>
<td>110</td>
</tr>
</tbody>
</table>

### Table 4 Comparison of the best solutions of various algorithms

<table>
<thead>
<tr>
<th>Instance</th>
<th>TS Euchi and Chabchoub (2010)</th>
<th>RIP Bolduc et al. (2008)</th>
<th>LR-based Heuristic</th>
<th>GAP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE-01</td>
<td>1119.47</td>
<td>1119.47</td>
<td>1121.33</td>
<td>0.17%</td>
</tr>
<tr>
<td>CE-02</td>
<td>1814.52</td>
<td>1814.52</td>
<td>1846.98</td>
<td>1.79%</td>
</tr>
<tr>
<td>CE-03</td>
<td>1920.90</td>
<td>1937.23</td>
<td>1988.55</td>
<td>3.52%</td>
</tr>
<tr>
<td>CE-04</td>
<td>2512.64</td>
<td>2528.36</td>
<td>2610.37</td>
<td>3.89%</td>
</tr>
<tr>
<td>CE-05</td>
<td>3097.67</td>
<td>3107.04</td>
<td>3263.29</td>
<td>5.35%</td>
</tr>
<tr>
<td>CE-06</td>
<td>1207.47</td>
<td>1207.47</td>
<td>1208.33</td>
<td>0.07%</td>
</tr>
<tr>
<td>CE-07</td>
<td>2004.53</td>
<td>2006.52</td>
<td>2038.99</td>
<td>1.72%</td>
</tr>
<tr>
<td>CE-08</td>
<td>2052.05</td>
<td>2052.05</td>
<td>2126.57</td>
<td>3.63%</td>
</tr>
<tr>
<td>CE-09</td>
<td>2425.26</td>
<td>2436.02</td>
<td>2518.06</td>
<td>3.83%</td>
</tr>
<tr>
<td>CE-10</td>
<td>3386.08</td>
<td>3407.13</td>
<td>3561.28</td>
<td>5.17%</td>
</tr>
<tr>
<td>CE-11</td>
<td>2330.94</td>
<td>2332.21</td>
<td>2403.41</td>
<td>3.11%</td>
</tr>
<tr>
<td>CE-12</td>
<td>1952.86</td>
<td>1953.55</td>
<td>1976.68</td>
<td>1.22%</td>
</tr>
<tr>
<td>CE-13</td>
<td>2859.12</td>
<td>2858.94</td>
<td>2947.29</td>
<td>3.09%</td>
</tr>
<tr>
<td>CE-14</td>
<td>2213.02</td>
<td>2216.68</td>
<td>2247.54</td>
<td>1.56%</td>
</tr>
</tbody>
</table>

### 5. CONCLUSIONS

The meta-heuristic has drawn significant attention in solving various kinds of vehicle routing problems. However, its inherited complexity, as well as the heavy computational load for some occasions, limits its applicability to some real world situations. We made use of several mathematical programming techniques, including the set covering model, the Lagrangian relaxation, and the column generation, to develop a VRPPC heuristic, which could serve as an alternative.
The VRPPC is first formulated in the form of the set covering problem, and the Lagrangian relaxation is used as the backbone in designing the iterative algorithm. In addition, a concept similar to column generation is used to update the solution space, a partial set of routes. Based on the results in the numerical experiment, the developed LR algorithm should be applicable to the real operational environment as its solution quality and the computation time is acceptable. However, there is a significant room for improvement, and the following are the future research directions.

The efficiency of the developed algorithm can be improved if the values of parameters are carefully tuned, and some steps (for example, the exchange of customers between the private fleet and the common carrier) are re-designed. Besides, optimizing the computer codes or using a faster programming language may be helpful for reducing the computation time.

The adjustment of the solution space is made by adding new routes through the insertion procedure, but no route is deleted. However, when the number of iterations increases, it may cast a serious impact on the computational load, as the number of routes in the solution can be huge. It is considered that a dynamic adjustment mechanism can be implemented to preserve only the promising routes in the solution space.

The proposed solution framework can be extended to solve the VRPPC with the time window, the pick-up and delivery, and/or the heterogeneous vehicles. In addition, as the SCP approach owns the advantage of flexibility in terms of the operational rules and the route cost determination, the real world problems with many practical considerations can be a suitable extension for this study.

REFERENCES


