Issues for the Linehaul-Feeder Vehicle Routing Problem with Virtual Depots and Time Windows

Huey-Kuo CHEN

Graduate School of Civil Engineering, National Central University, Taoyuan, 32001, Taiwan
E-mail: ncutone@ncu.edu.tw

Abstract: The linehaul-feeder vehicle routing problem with virtual depots and time windows (LFVRPTW) is a new version of city logistics. Four important issues are explored and fifteen sample examples used for demonstration. First, the LFVRPTW always yields better results than the vehicle routing problem with time windows (VRPTW). Second, LFVRPTW is advantageous compared with VRPTW. Third, more virtual depot (VD) candidates would yield a bit better results than fewer VD candidates. Fourth, less restricted time window constraints can yield significant benefit to the LFVRPTW. We also test the combined effect of VD candidates and time window constraints. We conclude that: (1) The LFVRPTW problem generally performs better than the VRPTW; (2) The LFVRPTW algorithm embedding the modified sequential insertion heuristic I1 is more effective than that containing the minimum cost insertion method; (3) For LFVRPTW, a looser time window constraint is more critical than the addition of more VD candidates.

Keywords: Linehaul-Feeder; Vehicle Routing Problem; Virtual Depot; Modified Sequential Insertion Heuristic I1

1. INTRODUCTION

The linehaul-feeder vehicle routing problem with virtual depots and time windows (LFVRPTW) is an important new version of city logistics, which has recently been introduced and studied by Chen et al. (2011a) and Chen and Wang (2011). In the LFVRPTW, small and large vehicles deliver services to two types of customers (type-I and type-II) within time constraints. Type-II customers are always serviced by small vehicles due to difficulties in parking and/or road accessibility. A type-I customer is serviced by a large vehicle if it acts as a virtual depot (VD) for reloading services; otherwise, it can be serviced by either a small vehicle or a large vehicle depending on the insertion cost. In addition, small vehicles en route may reload commodities from either the physical depot (PD) or from a larger vehicle at a VD before continuing onward. The LFVRPTW may be classified as a vehicle routing problem with additional complexity generated by variable VDs and time window constraints. This type of problem is especially useful in a situation where most customers cannot be accessed and serviced by a large vehicle and/or the PD is distant, and hence costly, for the dispatched small vehicles to get back for reloading. With the two-stage solution heuristic involving Tabu search proposed by Glover (1989, 1990) and Glover and Laguna (1997), Chen et al. (2011a) and Chen and Wang (2011) demonstrate with numerical examples that the LFVRPTW performs better than the vehicle routing problem with time windows (VRPTW) in terms of both the objective value and the number of small vehicles dispatched. However, there are some interesting issues yet to be explored. As a follow-up study for the LFVRPTW problem, here we name four and discuss them in order.
The first issue concerns how the solution quality of the LFVRPTW would be affected by different solution algorithms. To this end, two solution algorithms, called “A” and “B” algorithms, will be compared. The entire solution procedure of these two solution algorithms is about the same, i.e., a two-stage solution heuristic. However, their methods of inserting customers into vehicle routes are different, i.e., the minimum cost insertion method versus the modified sequential insertion heuristic II. The former heuristic was first proposed by Chen et al. (2011a) and modified later by Chen and Wang (2011) by allowing more than one large vehicle and imposing capacity constraint for large vehicles. The latter heuristic is essentially a variation of the former heuristic. This variation replaces the minimum cost insertion method with the modified sequential insertion heuristic II, which is believed to be more effective for situations where each route can service relatively few customers, as indicated by Dullaert and Bräysy (2003). To highlight the advantage of the LFVRPTW, a solution algorithm called “VRPTW-B” for solving the VRPTW is also included and compared.

These solution algorithms are then examined further for different scenarios of interest. The second issue is, thus, to determine how advantageous the LFVRPTW is when compared with the VRPTW under different levels of customer demands; these different demand levels can be easily generated by varying the scaling factors of customer demands. The third issue concerns how the final solution would be affected by changes in the numbers of VD candidates available. The fourth issue is to explore how the LFVRPTW would benefit from looser time windows for customers. All four issues will be extensively elaborated using 15 test examples.

The remainder of this paper is organized as follows. Section 2 presents the two solution algorithms for the LFVRPTW using 15 test problems. Section 3 compares the results derived from the LFVRPTW to those from the VRPTW. Section 4 conducts a sensitivity analysis for different scaling factors of customer demands, different number of VD candidates available, as well as different time window constraints. Finally, Section 5 concludes with a few remarks.

2. SOLUTION ALGORITHMS

The LFVRPTW is a new variation of VRPTW problems, which can generally be solved by a two-stage solution algorithm (Bräysy and Gendreau, 2005a, 2005b). In the first stage, an initial solution is constructed, whereas in the second stage the initial solution is improved by using a local search, a metaheuristic, or both. The quality of the final solution is heavily dependent on whether the initial feasible solution constructed in the first stage can be efficiently improved by the heuristics adopted in the second stage. It is intuitive that, for a loosely constrained vehicle routing problem (VRP), the quality of an initial solution is not a serious concern because any initial solution can be efficiently improved to reach a “good” feasible final solution using any well-known heuristic, such as Tabu search. However, for a highly constrained VRP problem like the LFVRPTW, this advantage may not exist at all. The reason is simply because any initial solution for highly constrained problems cannot be easily improved to attain a “good” feasible final solution due to complex constraints. Moreover, past research indicates that high quality initial solutions allow metaheuristics to achieve higher quality solutions more quickly, as described by Liu and Shen (1999), Van Breedam (2001), and Bräysy and Dullaert (2002). It means that the quality of the initial solution is critical to the quality of final solutions for the LFVRPTW, and indeed should be emphasized as essential.

To solve the LFVRPTW problem, Chen et al. (2011a) first proposed a two-stage solution algorithm. In the first stage, the minimum cost insertion method (which will be
described in Section 2.2) is used to insert customers into vehicle routes whereas in the second stage Tabu search is adopted to improve the initial solution. As discussed in the previous section, the quality of the initial solution is critical to the final solution of the LFVRPTW; we will hereinafter put our focus on the generation of initial solutions. In other words, given the same improvement procedure in the second stage, we explore how an initial solution generated in the first stage affects the quality of its final solution. The steps of the solution algorithm in the second stage will not be discussed here; the interested reader may refer to Chen et al. (2011a) for details.

2.1. Conceptual Framework for Generating Initial Solutions

The first initial solution algorithm proposed by Chen et al. (2011a) treats the large and small vehicles in a sequential order. Small vehicles are first routed to service all type-II customers subject to three constraints: customer time window, vehicle capacity, and driver work time limitation. When its loading is empty or near-empty, a small vehicle may choose either the PD or a VD for reloading if the corresponding saving is warranted. Consider that the VDs that currently perform worse may have the chance to improve in subsequent routes; the second best VD is also taken into account, but the maximum number of route branches for all vehicles is kept under 20.

When small vehicle routes are constructed on a temporary or provisional basis, the large vehicle departs from the PD, services all VDs, and returns to the PD. Since the small vehicle reloading operation at a VD can be executed only when both the matched small and large vehicles are present, late arrival of the large vehicle may affect the subsequent route plan of the matched small vehicle after reloading. If the subsequent route plan becomes infeasible, the unsatisfied customers remaining for the matched small vehicles after reloading need to be rescheduled.

Chen et al.’s (2011a) algorithm works very well and has demonstrated its efficiency with numerical examples. However, their algorithm is not without room for further improvement. First, the algorithm assumes that only a single large vehicle of unspecified capacity is used. This assumption is not realistic. In practice, vehicles, whether small or large, must have a capacity constraint. In addition, the number of large vehicles available should not be fixed (at one), because more large vehicles can be rented from car rental companies for a short period of time if this decision is warranted. A similar arrangement can be equally employed for the small vehicles. Second, since the large vehicle and small vehicle are interdependent with each other, they cannot be constructed separately. For example, when a small vehicle needs to reload, we should know the exact position of large vehicles at that time for choosing the optimal VD. Otherwise, the incurred costs of reloading will be miscalculated due to incomplete information about the locations of the large vehicles.

To remedy the drawbacks associated with Chen et al.’s (2011a) algorithm, Chen and Wang (2011) have proposed a new routes construction heuristic that allows more than one large vehicle to be used and imposes a capacity constraint for them. Moreover, this new algorithm dispatches small vehicles and large vehicles (if necessary) simultaneously. The general steps for generating initial routes by Chen and Wang (2011) are excerpted here for easy reference:

Step 0: Initialization.
Sort all unserviced type-II customers in order of time priority. If the time window constraint is the same, then the distance will be used to determine the order. The ordered customers are stored in the sequence list.

Step 1: Construct small vehicle routes (for the type-II customers in the sequence list)
and large vehicle routes (for the VD customers) simultaneously using the minimum cost insertion method.

Step 2: Select the best initial solution among all feasible solutions and stop.

In Step 1, each type-II customer in the sequence list is inserted into one small vehicle route by the minimum cost insertion method subject to the three constraints: customer time window, vehicle capacity, and driver work time limitation. If reloading is needed for a small vehicle, a type-I customer is chosen as a VD, which must be serviced by a large vehicle. In other words, this VD customer must be inserted into a large vehicle route and after the reloading operation, the loads of both the small vehicle and the large vehicle must be updated. Subsequently, the schedule of all routes will be modified by taking into account the costs and delay caused by the insertion of the VD. If a Type-I customer is not chosen as a VD at any time during the solution procedure, it will be treated as a non-VD Type-I customer and inserted into a small or large vehicle route.

2.2. Criteria for Constructing Vehicle Routes

Note that the criterion used to construct vehicle routes plays an important role in determining the quality of the generated initial solution. In addition to the minimum cost insertion method adopted by Chen and Wang (2011), there have been many other criteria available in the literature of vehicle routing. For the purpose of comparison, we select one other criterion which is called the modified sequential insertion heuristic I1 (Dullaert and Bräysy, 2003). The modified sequential insertion heuristic I1 is essentially modified from Solomon’s (1987) sequential insertion heuristic I1 for routing problems in which the number of customers per route is relatively small. By using a new measure for the additional time to insert a customer, routing costs in the initial solution can be reduced significantly. Since the LFVRPTW is generally characterized by a high customer demand-to-capacity ratio as well as tight time window constraints, Dullaert and Bräysy’s (2003) algorithm, which is suitable for cases where the number of customers per route is relatively few, may be more appropriate for use. The general steps for generating initial routes using the modified sequential insertion heuristic I1 can be stated as follows:

- Step 1: Construct small vehicle routes (for all type-II customers) and large vehicle routes (for the VD customers) simultaneously using the modified sequential insertion heuristic I1.
- Step 2: Select the best initial solution among all feasible solutions and stop.

Unlike the minimum cost insertion method, the modified sequential insertion heuristic I1 does not need to construct a sequence list for all unserviced customers in terms of time priority. Rather, it inserts unserviced customers directly into routes depending on the cheapest insertion position calculated by the modified sequential insertion heuristic I1.
3. COMPARISONS OF SOLUTION ALGORITHMS FOR THE LFVRPTW

We test our LFVRPTW problem using numerical examples. The fifteen test problems for the LFVRPTW were adopted from the VRP web site (http://neo.lcc.uma.es/radi-aeb/WebVRP/) (2011) with necessary modifications (see Table 1). All tests were run on a personal computer equipped with Intel i5 460M 2.53GHz CPU and 2GB memory under Microsoft Windows 7 computer environment.

3.1. Test Problems

Table 1. Fifteen test examples

<table>
<thead>
<tr>
<th>Problem No</th>
<th>Number of nodes</th>
<th>Small vehicle capacity</th>
<th>Large vehicle capacity</th>
<th>Scaling factor of customer demands</th>
<th>Virtual depot candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>R101(50)</td>
<td>50</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>21,33,37,48</td>
</tr>
<tr>
<td>R102(50)</td>
<td>50</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>21,33,37,48</td>
</tr>
<tr>
<td>R103(50)</td>
<td>50</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>21,33,37,48</td>
</tr>
<tr>
<td>RC101(50)</td>
<td>50</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>24,30,39,46</td>
</tr>
<tr>
<td>RC102(50)</td>
<td>50</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>24,30,39,46</td>
</tr>
<tr>
<td>RC103(50)</td>
<td>50</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>24,30,39,46</td>
</tr>
<tr>
<td>R101(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>7,33,72,98</td>
</tr>
<tr>
<td>R102(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>7,33,72,98</td>
</tr>
<tr>
<td>R103(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>7,33,72,98</td>
</tr>
<tr>
<td>RC101(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>32,41,86,88</td>
</tr>
<tr>
<td>RC102(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>32,41,86,88</td>
</tr>
<tr>
<td>RC103(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>32,41,86,88</td>
</tr>
<tr>
<td>RC104(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>32,41,86,88</td>
</tr>
<tr>
<td>RC105(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>32,41,86,88</td>
</tr>
<tr>
<td>RC106(100)</td>
<td>100</td>
<td>200</td>
<td>2000</td>
<td>1.0</td>
<td>32,41,86,88</td>
</tr>
</tbody>
</table>

Note: Test examples are modified from http://neo.lcc.uma.es/radi-aeb/WebVRP/.

a Numbers in the parenthesis denote the number of customers.
b Scaling factor represents a multiplier to the original demand.
c Number of virtual depot candidates is set to four, one in each quadrant.

General information associated with these problems is as follows:
(1) triangular inequality relations of customers’ distance are held in the network;
(2) all pairs of customers are connected;
(3) link travel distances of small vehicles are represented by the Euclidian distance between the two customers;
(4) link travel costs of small vehicles are represented by the corresponding link travel distances divided by 40 km/hr;
(5) costs for a large vehicle (1200cc van) include rent and insurance; stated in New Taiwan Dollars (NTD), costs are:
  (NTD 1450/day), fuel cost (NTD 3.3/km), and driver’s wage (NTD 270/hr);
(6) costs for a small vehicle (125cc motorcycle) consist of rent and insurance (NTD 600/day), fuel cost (NTD 0.77/km), and driver’s wage (NTD 120/hr).
(7) The objective value refers to the total travel cost, which includes the fixed cost (including rent and issuance) and variable cost (including the fuel cost and driver’s wage).
3.2. Test Results

Three algorithms are used for comparison. The first algorithm, named “VRPTW-B”, is essentially a simplified version of the third algorithm, i.e., this “B” algorithm, by deactivating reloading function of the LFVRPTW, yields a solution to a VRPTW. The second algorithm, called “A” algorithm, adopts the proposed two-stage solution procedure that embeds the minimum cost insertion method (cf. Section 2.2). The third algorithm, called “B” algorithm, adopts the proposed two-stage solution procedure that embeds the modified sequential insertion heuristic II (cf. Section 2.2). All fifteen instances test results for the LFVRPTW as well as the VRPTW are obtained, but only averaged results are summarized in Table 2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Scaling factor of customer demands</th>
<th>Objective value</th>
<th>Deviation percentage (%)</th>
<th>No. of small vehicles (large vehs)</th>
<th>CPU time (Sec)</th>
<th>No. of Depot</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRPTW-B(^a)</td>
<td>1.0(^d)</td>
<td>11392.4</td>
<td>--</td>
<td>12.5</td>
<td>0.3</td>
<td>--</td>
</tr>
<tr>
<td>A(^b)</td>
<td></td>
<td>12152.0</td>
<td>6.25</td>
<td>13.3</td>
<td>0.4</td>
<td>--</td>
</tr>
<tr>
<td>B(^c)</td>
<td></td>
<td>11392.4</td>
<td>0.0</td>
<td>12.5</td>
<td>0.5</td>
<td>--</td>
</tr>
</tbody>
</table>

Note:
\(^a\) The “VRPTW-B” algorithm adopts the modified I1 method to insert customers into routes.
\(^b\) The “A” algorithm adopts the minimum cost insertion method to insert customers into routes.
\(^c\) The “B” algorithm adopts the modified sequential insertion heuristic I1 to insert customers into routes.
\(^d\) Scaling factor represents a multiplier to the original demand.
\(^e\) Deviation percentage is defined as the deviation from the VRPTW in percentage (%)

The computational results can be summarized as follows:

1. In terms of number of virtual depots: All fifteen examples show that only small vehicles are required. There is no need for large vehicles and hence for virtual depots, implying the LFVRPTW is essentially reduced to the VRPTW. In this situation, the result obtained from “VRPTW-B” algorithm is identical to that from “B” algorithm. Therefore, in the following three points, we will neglect the results from “VRPTW-B” algorithm and only compare “A” algorithm with “B” algorithm.

2. In terms of objective values: All fifteen examples show that the objective values obtained from “B” algorithm are better than that from “A” algorithm for the LFVRPTW, with an average improvement rate of 6.25%.

3. In terms of number of small vehicles needed: All fifteen examples show that the number of small vehicles required from “B” algorithm (with the average number of small vehicle routes =12.5) for the LFVRPTW is fewer than that from “A” algorithm (with the average number of small vehicle routes =13.3).

4. In terms of computational times: It is observed that the average CPU times for the LFVRPTW solutions are 0.4 sec on average from “A” algorithm and 0.5 sec from “B” algorithm on average.

With the customer demands given in Table 2, the LFVRPTW problem is essentially reduced to the pure VRPTW problem. After looking into the solution of each test example, we found the customer demands are too low for small vehicles to use the VDs for reloading. To justify the usage of the LFVRPTW, we conduct sensitivity analysis by scaling up the
coefficients of customer demands, and also analyze effects of changes in other features, in the following sections.

4. SENSITIVITY ANALYSES

4.1. Scaling Factors of Customer Demands

Chen et al. (2011b) have shown that the higher the customer demands, the greater the benefit of using the linehaul-feeder vehicle routing problem with virtual depots (LFVRP-VD) rather than the vehicle routing problem (VRP). This benefit occurs because, whenever customer demands are high, the function of reloading service can be fully utilized. We conjecture that this advantage can be equally applied to the LFVRPTW which is known to be more complicated than the LFVRP-VD problem by imposing the additional constraints of time windows. The averaged test results for the LFVRPTW as well as the VRPTW are summarized in Table 3.

The computational results can be summarized as follows:

1. In terms of objective values: All fifteen examples show that the objective values obtained from algorithms “A” and “B” for the LFVRPTW are better than that from algorithm “VRPTW-B” for the VRPTW, with an average improvement rate of 3.25% and 6.34%, respectively. Moreover, algorithm “B” is superior to algorithm “A” because it outperforms in 13 of 15 test problems.

2. In terms of number of small vehicles needed: All fifteen examples show that the number of small vehicles required for the LFVRPTW is fewer than that from “VRPTW-B” for the VRPTW. The average number of small vehicle routes required for LFVRPTW delivery operation is 24.9 (from “A” algorithm) and 23.2 (from “B” algorithm) whilst the VRPTW needs 29.7 small vehicles.

3. In terms of computational times: It is observed that the average CPU times for the LFVRPTW solutions are 10.2 sec (from “A” algorithm) and 10.4 sec (from “B” algorithm), which are much higher than that of 0.4 sec from “VRPTW-B” for VRPTW. In these solutions, most of the computational time is spent in performing the reloading operation, with the result being an average improvement rate of about 3.25% (from “A” algorithm) and 6.34% (from “B” algorithm), respectively, in terms of average objective value. Since the average computational times for solving the LFVRPTW are less than 11 seconds for all fifteen test problems, the computational efficiency should not be a big concern.

All 15 test problems indicate the advantage of the LFVRPTW over the VRPTW with respect to the objective value. This result confirms our conjecture that the LFVRPTW is advantageous over the VRPTW.
Table 3. Averaged Results for the LFVRPTW and the VRPTW
(with customer demand factors equal to 5.0 and 4.0)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Scaling factor of customer demands</th>
<th>Objective value</th>
<th>Deviation percentage (%)</th>
<th>No. of small vehicles (large vehs)</th>
<th>CPU time (Sec)</th>
<th>No. of Depot</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRPTW-B(^a)</td>
<td>4.4(^d)</td>
<td>24544.9</td>
<td>--</td>
<td>29.7</td>
<td>0.4</td>
<td>--</td>
</tr>
<tr>
<td>A(^b)</td>
<td>23771.7</td>
<td>-3.25</td>
<td>24.9</td>
<td>10.2</td>
<td>1.73</td>
<td></td>
</tr>
<tr>
<td>B(^c)</td>
<td>23080.5</td>
<td>-6.34</td>
<td>23.2</td>
<td>10.4</td>
<td>2.06</td>
<td></td>
</tr>
</tbody>
</table>

Note: \(^{a, b, c, e}\) refer to the “Note” in Table 2 for the annotation.

\(^d\) Scaling factors of customer demands are set as 5.0 for 50-customer problems and 4.0 for 100-customer problems.

To show how the advantage of the LFVRPTW over the VRPTW is related to the scaling factors, we use 9 scaling factors from 1.0 to 5.0 with increments of 0.5 and rerun the computer codes. The experiments show that the LFVRPTW problem outperforms the VRPTW for all test examples.

We use problem R103(50) for illustration. As shown in Figure 1, when the scaling factor of customer demands is set to 3 or higher, the objective value of the VRPTW is higher than that of the LFVRPTW. This positive effect from higher scaling factors can be further verified by the number of small vehicles needed. As shown in Figure 2, when the scaling factor of customer demands is set to 3, the number of small vehicles required for the LFVRPTW is equal to 11, which is lower than that for the VRPTW problem (=12). This advantage becomes more pronounced when the scaling factor is set to 5.

The effect of scaling factors for customer demands can also be explained by a new index, called demand-to-capacity (D/C) ratio, which is defined as the percentage (%) of the average customer demand to the capacity of a small vehicle. The reverse of the D/C ratio of the test problem R103(50) is equivalent to the average number of customers to be serviced by each small vehicle. For instance, when the D/C ratio is 20%, it implies that, due to capacity constraint, each small vehicle services approximately five customers. It can be seen from Figure 1 that, when the D/C ratio is below 20%, there is no apparent advantage of the LFVRPTW over the VRPTW. However, starting from a D/C ratio equal to 20%, the advantage of the LFVRPTW over the VRPTW becomes evident, which implies that reloading service at VDs is advantageous for a routing problem with relatively few customers per small vehicle route. Similar phenomenon can also be observed in Figure 2.

In fact, all test examples show a pattern similar to the one for problem R103(50). Therefore, we can draw a conclusion: the LFVRPTW problem generally performs better than the VRPTW when the scaling factor of customer demands is set at a threshold, and the advantage would become greater for higher customer demands.
Figure 1. Objective values versus scaling factor of customer demands (problem R103(50))

Figure 2. Number of dispatched vehicles versus scaling factor of customer demands (problem R103(50))

4.2. Number of Virtual Depot Candidates

Chen et al. (2011b) have shown that eight type-I customers perform better than four type-I customers in terms of average objective value for the LFVRP-VD. This is because more type-I customer nodes generally imply more flexibility for reloading, suggesting better solutions. Again, we conjecture that this advantage can be equally applied to the LFVRPTW, which is known to be more restricted than the LFVRP-VD problem by imposing time window constraints. To justify this, we compare the solutions of the eight type-I customer problems to the four type-I customer problems. The additional type-I customer in each quadrant is set as the customer with the maximum total distance to other customers in the same quadrant.

As can be seen in Table 4, both “A” and “B” algorithms for solving the LFVRPTW perform better, with similar trend, than the “VRPTW-B” algorithm, for solving the VRPTW. Moreover, the example with eight type-I customers performs better than the one with four type-I customers in terms of average objective value, with an improvement of 1.72% and
2.3%, respectively, for “A” and “B” algorithms. This improvement is due mainly to being able to choose from among a greater number of virtual depot candidates. The improvement rates, however, are rather small, which is probably due to tight time window constraints. To test whether the reloading services can be fully utilized by having looser time window constraints, we proceed with an experiment in the next section.

<table>
<thead>
<tr>
<th>Problem No.</th>
<th>Objective value</th>
<th>Depot number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“A” algorithm</td>
<td>“B” algorithm</td>
</tr>
<tr>
<td></td>
<td>4 Type-I customers</td>
<td>8 Type-I customers</td>
</tr>
<tr>
<td>R101 (50)</td>
<td>15955.9</td>
<td>15935.0 (0.1)</td>
</tr>
<tr>
<td>R102 (50)</td>
<td>15750</td>
<td>15371.9 (2.5)</td>
</tr>
<tr>
<td>R103 (50)</td>
<td>13997</td>
<td>13997 (0)</td>
</tr>
<tr>
<td>R1010 (50)</td>
<td>21835.3</td>
<td>21543.6 (-1.4)</td>
</tr>
<tr>
<td>R102 (50)</td>
<td>20458.6</td>
<td>18988.2 (-3.7)</td>
</tr>
<tr>
<td>R103 (50)</td>
<td>19684.2</td>
<td>19311.8 (-1.9)</td>
</tr>
<tr>
<td>R101 (100)</td>
<td>24179.8</td>
<td>24005.2 (-0.7)</td>
</tr>
<tr>
<td>R102 (100)</td>
<td>24519</td>
<td>24374 (-0.6)</td>
</tr>
<tr>
<td>R103 (100)</td>
<td>24100</td>
<td>23567.7 (-2.3)</td>
</tr>
<tr>
<td>R1010 (100)</td>
<td>30482</td>
<td>29946.5 (-1.8)</td>
</tr>
<tr>
<td>R102 (100)</td>
<td>29750.9</td>
<td>29590.3 (-0.5)</td>
</tr>
<tr>
<td>R103 (100)</td>
<td>29351.4</td>
<td>29364.8 (-0.1)</td>
</tr>
<tr>
<td>R104 (100)</td>
<td>27131</td>
<td>26790.4 (-1.3)</td>
</tr>
<tr>
<td>R105 (100)</td>
<td>29733</td>
<td>28774 (-3.3)</td>
</tr>
<tr>
<td>R106 (100)</td>
<td>29648</td>
<td>29004.2 (-2.2)</td>
</tr>
<tr>
<td>Average</td>
<td>23771.7</td>
<td>23370.9 (-1.72)</td>
</tr>
</tbody>
</table>

4.3. Limits of Time Windows

To test how the LFVRPTW can benefit from looser time window constraints, we relax the lower time limit of each customer by 50 units while the upper time limit is kept the same, i.e., \{e_i, li+50\} for each customer i. The averaged results (Table 5) obtained from rerunning the computer codes can be summarized as follows:

1. In terms of objective values: The average improvement of the “B” algorithm is 19.07% which is higher than the “A” algorithm (12.5%).
2. In terms of computational time: The average computational times for solving all fifteen test problems is less than 12 seconds; again the computational efficiency should not be a big concern.
### 4.4. Number of Virtual Depot Candidates and Limits of Time Windows

Next, it would be interesting to know the combined effect of eight VD candidates and looser time windows (by relaxing the lower time limit by 50 time units) on the result of the LFVRPTW. The averaged results (Table 6) obtained from rerunning the computer codes can be summarized as follows:

1. **In terms of objective values:** The average improvement of the “B” algorithm is 21.53% which is higher than the “A” algorithm (13.71%).

2. **In terms of computational time:** The average computational times for solving all fifteen test problems are less than 12 seconds.

### Table 6. Averaged Results for 8 Type-I customers and looser time windows (lt+50)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Scaling factor of customer demands</th>
<th>Objective value</th>
<th>Deviation percentage (%)</th>
<th>No. of small vehicles (large vehs)</th>
<th>CPU time (Sec)</th>
<th>No. of Depot</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRPTW-B</td>
<td></td>
<td>24218.8</td>
<td>--</td>
<td>29.5</td>
<td>0.8</td>
<td>--</td>
</tr>
<tr>
<td>A</td>
<td>4.4(^d)</td>
<td>21528.5</td>
<td>-12.50</td>
<td>19.8</td>
<td>10.7</td>
<td>2.07</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>20339.8</td>
<td>-19.07</td>
<td>18.5</td>
<td>11.3</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Note: \(a, b, c, d, e\) refer to the “Note” in Table 3 for the annotation.

### 4.5. Summary

From the sensitivity analysis, we get the following two conclusions:

1. **The LFVRPTW problem generally performs better than the VRPTW when the scaling factor of customer demands is set to a certain value and up.** This advantage would become more significant as the value of the scaling factor increases.

2. **“B” algorithm outperforms “A” algorithm in all different scenarios.** This finding confirms our conjecture that the modified sequential insertion heuristic I1 is more effective than the minimum cost insertion method for solving the LFVRPTW problem.

In fact, we can perform still further analysis for each algorithm across the different factors, i.e., 8 VD candidates, looser time windows, as well as a combined effect. From Table 7 we can see that, as compared with the case of a higher scaling factor (from Table 3), the performance of each algorithm for different scenarios can be summarized as follows:

1. **“VRPTW-B” algorithm:**
   
   1. **In terms of objective value:** For the case with 8 VD candidates the average objective value (=24544.9) is exactly the same as that with 4 VD candidates. More
VDs do not have any effect at all because no reloading is required. For the case with looser time window constraints, the average objective value (≈24218.8) is about the same as that with original time window constraints, i.e., 24544.9. The “VRPTW-B” algorithm can benefit only from a looser time window constraint (with improvement rate of 1.35%).

(2) In terms of number of small vehicles: For the case with 8 VD candidates the number of small vehicles required (≈29.7) is exactly the same as that with 4 VD candidates. More VDs do not have any effect at all because no reloading is required. For the case with looser time window constraints, the number of small vehicles required (≈29.5) is about the same as that with original time window constraints, i.e., 29.7.

(3) This performance implies that, with its lack of reloading services, the VRPTW cannot be improved with more VD candidates, and neither can it be largely benefited from looser time window constraints.

2. “A” algorithm:

(1) In terms of objective value: For the case with more VD candidates the average objective value (≈23370.9) is a bit better than that with original number of VD candidates, i.e., 23771.7 (with improvement rate of 1.72%). More VDs do have small positive effect. For the case with looser time window constraints, the average objective value (≈21528.5) is significantly better than that with the original time window constraints, i.e., 23771.7 (with improvement rate of 10.42%). Their combined effect is 11.6% which is about the same magnitude as the looser time windows alone. This performance, therefore, implies that a looser time window constraint is more critical than more VD candidates for the LFVRPTW.

(2) In terms of number of small vehicles: For the case with 8 VD candidates the number of small vehicles required (≈24.0) is just a bit fewer than that with 4 VD candidates (≈24.9). More VDs have only a small effect because reloading services are not fully utilized. For the case with looser time window constraints, the number of small vehicles required (≈19.8) is much fewer than that with original time window constraints, i.e., 24.9 (with improvement rate of 25.75%), implying a large improvement. Their combined effect is 27.69%, which is about the same magnitude as the looser time windows alone.

3. “B” algorithm:

(1) In terms of objective value: For the case with more VD candidates, the average objective value (≈22561.32) is a bit better than that with original number of VD candidates, i.e., 23080.5 (with improvement rate of 2.3%). More VDs do have small positive effect. For the case with looser time window constraints, the average objective value (≈20339.8) is significantly better than that with original time window constraints, i.e., 23080.5 (with improvement rate of 13.47%). Their combined effect is 15.82% which is about the same magnitude as the looser time windows alone. This performance, therefore, implies that a looser time window constraint is more critical than more VD candidates for the LFVRPTW.

(2) In terms of number of small vehicles: For the case with 8 VD candidates, the number of small vehicles required (≈23.1) is just a bit fewer than that with 4 VD candidates (≈23.2). More VDs have only a small effect because reloading services are not fully utilized. For the case with looser time window constraints, the number of small vehicles required (≈18.5) is much fewer than that with original time window constraints, i.e., 23.2 (with improvement rate of 25.4%), implying a large improvement. Their combined effect is 27.47%, which is about the same magnitude
as the looser time windows alone.

4. Average value:

1) In terms of objective value: For the case with more VD candidates, the average objective value (23492.4) is a bit better than that with original number of VD candidates, i.e., 23799 (with improvement rate of 1.34%). More VDs do have a small positive effect. For the case with looser time window constraints, the average objective value (22029) is significantly better than that with original time window constraints, i.e., 23799 (with improvement rate of 8.41%). Their combined effect is 9.59% which is about the same magnitude as the looser time windows alone. This performance, therefore, implies that looser time window constraint is more critical than more VD candidates for the LFVRPTW.

2) In terms of number of small vehicles: For the case with 8 VD candidates, the number of small vehicles required (25.6) is exactly a bit fewer than that with 4 VD candidates (25.93). More VDs have only small effect because reloading services is not fully utilized. For the case with looser time window constraints, the number of small vehicles required (22.6) is much fewer than that with original time window constraints, i.e., 25.93 (with improvement rate of 14.73%), implying a large improvement. Their combined effect is 15.75%, which is about the same magnitude as the looser time windows alone.

From the above discussion, we can get once again conclude that a looser time window constraint is more critical than more VD candidates for the LFVRPTW.

Table 7. Summary of the averaged results from different scenarios

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Original problem (from Table 2)</th>
<th>Higher scaling factors (Baseline) (from Table 3)</th>
<th>8 VD candidate (from Table 4)</th>
<th>Looser time window (from Table 5)</th>
<th>Combined effect (from Table 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRPTW-Ba</td>
<td>11392.4 (-)9</td>
<td>12.5 (-)</td>
<td>24544.9 (-)</td>
<td>24544.9 (0)</td>
<td>24218.8 (-1.35)</td>
</tr>
<tr>
<td>A b</td>
<td>12152 (-)</td>
<td>13.3 (-)</td>
<td>23771.7 (-)</td>
<td>23370.9 (-1.72)</td>
<td>21528.5 (-10.42)</td>
</tr>
<tr>
<td>Bc</td>
<td>11392.4 (-)</td>
<td>12.5 (-)</td>
<td>23080.5 (-)</td>
<td>22561.3 (-2.3)</td>
<td>20339.8 (-13.47)</td>
</tr>
<tr>
<td>Average of averages</td>
<td>11645.6 (-)</td>
<td>12.77 (-)</td>
<td>23799 (-)</td>
<td>23492.4 (-1.34)</td>
<td>22029 (-8.41)</td>
</tr>
</tbody>
</table>

Note: a, b, c, d refer to the “Note” in Table 3 for the annotation.

Numbers in parenthesis denote the improvement rate which is defined as the deviation from the higher scaling factors in percentage (%)

5. CONCLUDING REMARKS

The objective of this research is to explore four important issues related to the LFVRPTW, i.e., line-haul feeder operations, reloading services, number of VD candidates, and time window restrictions. Fifteen sample examples are used for demonstration. The first issue concerns the
solution quality obtained by different solution algorithms. This first issue can be answered by comparing three algorithms, i.e., “VRPTW-B”, “A” and “B”. We confirm that the LFVRPTW always yields better results than the VRPTW, which is indeed analogous to the relation of the LFVRP-VD versus the vehicle routing problem (VRP) (Chen et al., 2011b). From the numerical examples, we did find that the LFVRPTW is advantageous over the VRPTW when the scaling factor of customer demands is set to 3 or higher.

The second issue concerns whether the introduced reloading services en route are advantageous in the situation where customer demands are high. The third issue is related to the number of VD candidates available in the test problems. We verify that more VD candidates available (in less congested and hence less restricted areas) would yield better results than for fewer VD candidates but the benefit is marginal. The fourth issue studies the extent to which the time window constraints would affect the result of the LFVRPTW. The experimental result indicates that less restrictive time window constraints can result in significant benefit for the LFVRPTW. In addition, we also test the combined effect of both more VD candidates and looser time window constraints on the solution quality of the LFVRPTW and found the obtained results are better than those from adopting only a single factor, i.e., using either more VD candidates or less restricted time window constraints individually.

The current paper attempts to shed some light on most important issues associated with the LFVRPTW. The major contribution of this research is to indicate the issues clearly and, based on discussion of these issues, to conclude that: (1) The LFVRPTW problem generally performs better than the VRPTW, and the advantage would increase especially when the scaling factor of customer demands is set to higher values; (2) The LFVRPTW algorithm embedding the modified sequential insertion heuristic I1 is more effective than that containing the minimum cost insertion method; and (3) A looser time window constraint is more critical than more VD candidates for the LFVRPTW. By incorporating these findings, the application of the LFVRPTW would be more efficient and worthwhile. Besides, systematic reporting of the four important issues would be a significant aid to improvement of the LFVRPTW practice in city logistics. It is also noted that real data from enterprises were difficult to obtain and hence not available to us, which prevents us from further validating our model and algorithms. Nevertheless, this practical research should be conducted in the immediate future.

Acknowledgments

The authors would like to acknowledge the Ministry of Science and Technology, Taiwan, for providing partial financial support for this work.

REFERENCES


