Received 12 April 2019
Revised 28 April 2019
23 May 2019
Accepted 4 June 2019

A heuristic algorithm for multiple trip vehicle routing problems with time window constraint and outside carrier selection

Ching-Wu Chu
Department of Shipping and Transportation Management, National Taiwan Ocean University, Keelung, Taiwan, and

Hsiu-Li Hsu
Department of Air and Sea Logistics and Marketing, Taipei College of Marine Technology, Taipei, Taiwan

Abstract

Purpose – In this paper, the authors introduced a real world new problem, the multi-trip vehicle routing problem with time windows and the possible use of a less-than-truckload carrier to satisfy customer demands. The purpose of this paper is to develop a heuristic algorithm to route the private trucks with time windows and to make a selection between truckload and less-than-truckload carriers by minimizing a total cost function.

Design/methodology/approach – Both mathematical model and heuristic algorithm are developed for routing the private trucks with time windows and for selecting of less-than-truckload carriers by minimizing the total cost function.

Findings – In all, 40 test problems were examined with the heuristics. Computational results show that the algorithm obtains the optimal or near-optimal solutions efficiently in terms of time and accuracy.

Originality/value – The research described in this paper differs from the previous one on fleet planning or vehicle routing, in that it modifies the Clarke and Wright method by shifting the performance measure from a distance to cost and also incorporates the fixed cost of different types of trucks into the model. In addition, the authors simultaneously consider the multiple trip vehicle routing problems with time windows and the selection of less-than-truckload carriers that is an integrated scenario of real-world application. To the best of the authors' knowledge, this scenario has not been considered in the literature.

Keywords Logistics, Heuristics, 0-1 Integer programming, Less-than- truckload, Multi-trip vehicle routing problems with time windows

Paper type Research paper

1. Introduction

Vehicle routing problem is to find optimal routes for multiple vehicles visiting a set of locations by minimizing the total travel length. A variety of vehicle routing problems has been studied in the literature to address different practical situations. Typically, different vehicle routing problems address different practical situations. Vehicle routing problem with time windows (VRPTW) is an important and practical problem for logistics managers. In many sectors of the

This research was partially supported by the National Science Council of Taiwan under grant 105-2410-H-019-005. The authors would like to thank the anonymous referees for their helpful comments.

economy, transportation costs amount for a fifth or even a quarter (lumber, wood, petroleum, stone, clay and glass products) of the average sales amount (Schneider, 1985). Thus, appropriately identifying and modeling the problems and developing algorithms to solve them have been the continuing research effort in the last several decades.

Our motivation for this study stems from observations of a local logistics company. This company owns different types of private trucks, and its main business is delivering food and beverages to wholesalers. The logistics company promises the customer that a shipment received during business hours will be delivered to the destination within 5 h, so the delivery time window is a major concern. Furthermore, the company is facing fluctuations in demand from its customers. When the customer demands are greater than the total capacity of owned trucks during the peak season, the company has three strategies to use: using overtime, outsourcing vehicles and using outside carriers. As the overtime cost and the rents of outsourcing vehicles are much higher than that of using an outside carrier, sometimes using an outside carrier is a more attractive option.

Regarding the outside carrier selection, a logistics manager can make a choice between a truckload (a private truck with multiple trips) and a less-than-truckload carrier (an outside carrier). A private truck allows a company to consolidate several shipments, going to different destinations, and in a single truck with multiple trips. A less-than-truckload carrier usually assumes the responsibility for routing each shipment from the origin to the destination. The freight charged by a less-than-truckload carrier is typically much higher than the cost of a private truck. Choosing the right customers to be served by outside carriers may yield significant cost savings to the company.

In this paper, we address the problem of routing a fixed number of private trucks with limited capacity from a central warehouse to customers with known demand and time windows. Furthermore, each vehicle can perform several trips during the working day. The objective of this paper is to develop a heuristic algorithm to route the private trucks with time windows and to make a selection of less-than-truckload carriers by minimizing a total cost function.

The literature of VRPTW can be classified into two categories, the exact method and heuristic algorithm. Although there are some exact methods (Laporte, 1992; Laporte and Nobert, 1998; Azi *et al.*, 2007), their application is limited because the solution time is exponentially increasing with the number of customers. Clearly, a heuristic algorithm remains a viable alternative for larger instances. Heuristic algorithms can be broadly classified into two categories: classical heuristics and metaheuristics.

Classical heuristics include construction and improvement approaches. Construction heuristics build a feasible route by iteratively inserting a customer into a current route based on maximum savings or minimum additional distance. Some examples of construction heuristics are Solomon (1987), Potvin and Rousseau (1993), Bramel and Simchi-Levi (1996) and Ioannou *et al.* (2001). Improvement heuristics modify the current solution iteratively by performing local searches for better solutions. Some examples of improvement heuristics are Potvin and Rousseau (1995), Russell (1995), Cordone and Wolfler Calvo (2001) and Bräysy *et al.* (2004).

Metaheuristics are general solution procedures exploring the solution space to identify good solutions and incorporating some classical heuristics. In contract to classical heuristics, metaheuristics allow infeasible and deteriorating solutions during the search process to escape from local optimum. So far, the Tabu Search and Genetic Algorithm have shown the best performance for vehicle routing problems (Mester and Bräysy, 2005). Tabu Search is a local search metaheuristic introduced by Glover (1986). Details about Tabu Search can also be found (Glover, 1989; Glover, 1990; Glover and Laguna, 1997). Rochat and Taillard (1995), Taillard *et al.* (1997), Chiang and Russell (1997), Schulze and Fahle (1999) and Cordeau *et al.* (2001) have successfully applied Tabu Search to the VRPTW. The ideas involved in Genetic Algorithm were originally developed by Holland (1975). A Genetic Algorithm begins with a pool of the population of chromosomes that these chromosomes undergo crossovers and mutation to generate some children. Although these children are different from parents, they inherit some characteristics from their parents. This process continues until no further improvement in the solution appears possible. Gendreau and Tarantilis (2010) reported good results with genetic algorithms. This year, Demir *et al.* (2019) published a book, and in Chapter 8, discussed the basic principles of vehicle routing to provide readers with a complete introductory resource.

Little research has examined the problem of choosing between a less-than-truckload and truckload carrier. Ball *et al.* (1985) considered a fleet planning problem for long-haul deliveries with fixed delivery locations and an option to use an outside carrier. Agarwal (1985) studied the static problem with a fixed fleet size and an option to use an outside carrier. Klincewicz *et al.* (1990) developed a methodology to address the fleet size planning and to route limited trucks from a central warehouse to customers with random daily demands. Chu (2005) introduced a heuristic to simultaneously select customers to be served by external transportation providers and to route a limited number of owned heterogeneous trucks. A carrier collaboration problem for less-than-truckload carriers was studied by Hernández and Peeta (2014). They considered a single-carrier collaboration problem in which a less-than-truckload carrier of interest seeks to collaborate with other carriers by acquiring the capacity to service excess demand. Recently, Wu *et al.* (2017) extended Chu's work (2005) by incorporating time windows constraint into a vehicle routing problem while less-than-truckload carriers are available.

In general, our research described here differs from the previous one on fleet planning or vehicle routing in that it modifies the Clarke and Wright method by shifting the performance measure from a distance to cost and also incorporates the fixed cost of different types of private trucks into the model. In addition, we simultaneously consider the multiple trip vehicle routing problems with time windows and the selection of less-than-truckload carriers that is an integrated scenario of real-world application. Furthermore, a mathematical model is also proposed to represent and solve the problem. To sum up, our problem is a generalization of the VRPTW, the multi-trip vehicle routing problem with time windows (MTVRPTW) and the heterogeneous vehicle routing problem (HVRP) with external carriers (following the abbreviation in literature, Koc, *et al.* (2016)). To the best of our knowledge, this scenario has not been considered in the literature. The rest of the paper is organized as follows. The next section formulates the mathematical model for our problem. Section 3 presents the heuristic algorithm. Computational results are reported in Section 4. Finally, concluding remarks and suggestions for future research are provided in Section 5.

2. Mathematical model

We formulate our mathematical model based on the following assumptions:

- A single warehouse system is considered; all trucks start at the warehouse and return to the warehouse.
- We restrict ourselves to delivery only.
- The requirements of all the customers are known, and each customer's requirement cannot exceed the truck capacity.
- The time window of each customer is known.
- Multiple trips are allowed on a truck.
- A maximum driving time is imposed on each route.

- Each customer is served by one truck (either by the private truck or the less-than-truckload carrier), and all customers' requirements must be met.
- The cost of operating the truck fleet consists of a fixed cost and a variable cost. The principal items in the fixed cost include personnel, insurance and truck depreciation. The main component of the variable cost is fuel, which is usually proportional to the traveled distance.

The relevant notations and integer programming model are as follows:

- *1*: the warehouse (initial point).
- *D*: the warehouse (terminal points).
- *N*: the set of demand nodes.
- *TN*: the set of all nodes° $TN \in N \cup \{1, D\}$.
- *R*: the set of all trips.
- *K*: the set of all trucks.
- *i*: $\{i = 1,2,\ldots,n\}$, the index set of customers including the warehouse.
- *j*: $\{j = 1,2,\ldots,n\}$, the index set of customers including the warehouse.
- *k*: $\{k = 1,2,\ldots,m\}$, the index set of trucks.
- *r*: $\{r=1,2,\ldots,R\}$, the index set of trips.
- *n*: the number of customers.
- *m*: the number of trucks.
- *R*: the maximum number of allowed trips for a vehicle.
- FC_k: the fixed cost of private truck k.
- C_{ijk}: the cost of truck k traveling from customer i to customer j.
- CL_i: the cost charged by the less-than-truckload carrier for serving customer i.
- q_i: the delivery of customer i.
- Q_k: the capacity of private truck k.
- s_i: the required service time of customer i.
- t_{ij}: the travel time from customer i to customer j.
- e_i: the earliest time to start service at customer i.
- l_i: the latest time to start service at customer i.
- *M*: a big number.
- T_{max}: the maximum route time allowed for a vehicle.

Decision variables:

$$X_{ijkr} = \begin{cases} 1, & \text{if the k-th truck at its r-th trip traveling} \\ & \text{from customer i to customer j,} \\ 0, & \text{otherwise,} \end{cases}$$

$$Y_{ikr} = \begin{cases} 1, & \text{if the customer i is served by the k-th truck} \\ & \text{at its r-th trip,} \\ 0, & \text{otherwise,} \end{cases}$$

$$L_i = \begin{cases} 1, & \text{if the service of customer i is satisfied by the less-than-truckload carrier,} \\ 0, & \text{otherwise,} \end{cases}$$

T_{ikr} = the start service time at customer i by the k-th truck at its r-th trip.
LQ_{ij} = the number of loaded shipments on the car while traveling from customer i to customer j.

$$\min z = \sum_{k\in K} FC_k * Y_{1k1} + \sum_{r\in R}\sum_{i\in TN}\sum_{j\in TN}\sum_{k\in K} C_{ijk} * X_{ijkr} + \sum_{i\in N} CL_i * L_i$$

Subject to

$$\sum_{k\in K} Y_{1k1} \le m \tag{1}$$

$$\sum_{k\in K}\sum_{r\in R} Y_{ikr} + L_i = 1 \qquad \forall\ i \in N \tag{2}$$

$$\sum_{j\in TN} X_{ijkr} = Y_{ikr} \qquad \forall\ i \in TN,\ k \in K,\ r \in R \tag{3}$$

$$\sum_{j\in TN} X_{jikr} = Y_{ikr} \qquad \forall\ i \in TN,\ k \in K,\ r \in R \tag{4}$$

$$\sum_{i\in N} q_i * Y_{ikr} \le Q_k \qquad \forall\ k \in K,\ r \in R \tag{5}$$

$$T_{ikr} + S_i + t_{ij} - M(1 - X_{ijkr}) \le T_{jkr} \qquad \forall\ i,j \in TN, k \in K,\ r \in R \tag{6}$$

$$e_j * \sum_{i\in TN} X_{ijkr} \le T_{jkr} \le l_j * \sum_{i\in TN} X_{ijkr} \qquad \forall j \in TN, k \in K,\ r \in R \tag{7}$$

$$T_{ikr} + S_i + t_{iD} \le T_{max} \qquad \forall i \in N, k \in K,\ r \in R \tag{8}$$

$$\sum_{i\in N} X_{1ikr} - \sum_{i\in N} X_{iDkr} = 1 \qquad \forall\ k \in K,\ r \in R \tag{9}$$

$$\sum_{i\in TN} LQ_{ij} - \sum_{i\in TN} LQ_{ji} = q_j \qquad \forall\ j \in N \tag{10}$$

$$\sum_{j\in N} LQ_{1j} = \sum_{j\in N} q_j \quad (11)$$

$$0 \le LQ_{ij} \le \sum_{k\in K}\sum_{r\in R} Q_k * X_{ij\ kr} \qquad \forall\ i,j \in TN \quad (12)$$

$$\sum_{j\in N} X_{1j\ k\ r} \ge \sum_{j\in N} X_{1j\ k\ r+1} \qquad \forall\ k\in K,\ r\in\{1,2,\ldots,|R|-1\} \quad (13)$$

$$T_{D\ k\ r} = T_{1\ k\ r+1} \qquad \forall\ k\in K,\ r\in\{1,2,\ldots,|R|-1\} \quad (14)$$

$$\forall\ X_{ijkr},\quad Y_{ikr},\quad L_i \quad \in \{0,\ 1\};\ \ T_{ikr},\quad LQ_{ij} \quad \ge 0$$

The objective function is to route the private trucks and to make a selection of less-than-truckload carriers by minimizing a total cost function. The first and second terms are the fixed cost and variable costs of the private trucks; the last term is the cost from the less-than-truckload carrier.

Constraint (1) ensures that at most m trucks are used.

Constraint (2) defines that each customer is served either by a trip of a private truck or a less-than-truckload carrier.

Constraints (3) and (4) guarantee that a truck arrives at a customer and also leaves that location.

Constraint (5) ensures that the loaded shipments cannot exceed the loading capacity of each truck.

Constraints (6)-(8) assure the feasibility of the schedule.

Constraint (9) guarantees that a truck leaves the initial point and arrives at the terminal point for each trip. The difference between the number of leaving the initial point and the number of arriving at the terminal point is equal to one.

Constraint (10) ensures that the loaded shipments on the truck should equal LQ_{ij} minus q_j after the truck leaving customer j.

Constraint (11) assures that the total loaded shipments at the warehouse is equal to the sum of the required shipment from all demand nodes.

Constraint (12) guarantees that the loaded shipments on the truck should be equal to or greater than zero; the loaded shipments on the truck cannot exceed the truck capacity.

Constraint (13) ensures that there exits the r-th trip of the k-th truck, then there is a possibility of $(r+1)$-th trip.

Constraint (14) assures that the arriving time at the warehouse of the k-th truck at its r-th trip is equal to the leaving time at the warehouse of the k-th truck at its $(r+1)$-th trip.

3. The heuristic algorithm

In this section, we describe our algorithm, called MTVRPTW-LTL, for solving the multiple trip vehicle routing problems with time windows, and the selection of less-than-truckload carriers. The heuristic algorithm can be decomposed into four main steps. In the following, we describe this algorithm by examining the main steps separately.

3.1 Selection

The first step requires the selection of a group of customers, who will be served by the less-than-truckload carriers. In this step, we check if the demand is greater than the total capacity of owned trucks. If it is not, we skip this step and implement the next step directly. To minimize the total cost, we have to design a procedure that can achieve this goal. In reality, the freight charged by the less-than-truckload carrier is usually higher than the cost handled by a private truck. The detail for selecting the customers is described as follows:

- Divide customers into four groups based on the quadrants.
- Calculate the total demand of all customers.
- Calculate the whole capacity of owned trucks.
- If the total demand from all customers is greater than or equal to the capacity of owned trucks, go to Step (5); otherwise, skip this procedure.
- Subtract the capacity of owned trucks from the total demand, which is the unsatisfied truck capacity.
- Sort the customers based on demand in descending order.
- Sum up the demand of each customer until the total demand is greater than or equal to the unsatisfied truck capacity. The corresponding customers will be the candidates served by the less-than-truckload carrier.
- Remove the corresponding customers in Step (7) from the quadrants.
- The remaining customers in the quadrants will be served by private trucks and will be used to construct an initial solution.
- Calculate the required number of trips (RNT), satisfying the truck capacity $\times$ RNT $\leq$ the total demand from all customers in Step (2).

3.2 Initial solution construction

The initial solution construction step is composed of five procedures: construct, truck reduction, route merge, reverse and check PF procedures.

The construct procedure is designed to construct the initial route within each quadrant. Two criteria, the latest time to start service at the customer and the distance between the depot and the customers, are used to build the route within each quadrant, customers are ordered on ascending order based the latest time to start service. If there exits the same the latest time to start service between two customers, the customer with a smaller distance between the depot and the customer will be chosen.

As we divide customers into four groups based on the quadrants, it is possible that a route contains only one customer while there are few customers. The truck reduction procedure is used to reduce the route with only one customer. It is easy to insert the only customer into another route. A loop is written to handle this scenario.

After truck reduction procedure is executed, the number of routes may be reduced. The route merge procedure is designed to combine different routes into a single route with multiple trips.

The reverse procedure is simply a service routine designed to reverse the sequence of any route. It is possible for the construct procedure to generate an infeasible route because the construct procedure does not take time window constraints into consideration. If this happens, the reverse procedure can significantly reduce the number of violations.

The PF procedure is used to check time feasibility when inserting a customer. All four procedures mentioned above adopt the PF procedure to check time feasibility. The same procedure was used in Solomon's (1987) study. The necessary and sufficient conditions for time feasibility when inserting a customer, say u, between i_{p-1} and i_p, $1 \leq p \leq m$, on a partially constructed feasible route $(i_0, i_1, i_2,\ldots, i_m)$, $i_0 = i_m = 0$ are $b_u \leq l_u$ and $b_i + PF_{ir} \leq l_{ir}$, $p \leq r \leq m$.

3.3 Refining procedure

A refining procedure is applied to the solution obtained through the initial solution step. This procedure is composed of a series of intra-route and inter-route arc exchanges which are well known in the literature.

3.3.1 Intra-route improvement. In this step, each route is improved by using further local search procedures. These procedures include the two-exchanges (swap exchange), and three-exchanges (insert_1 exchange), illustrated in Figures 1-2, respectively. Given a route, a swap exchange is obtained by replacing arcs (m, n) and (p, q) with arcs (m, p) and (n, q), as illustrated in Figure 1. For each node m, the Insert_1 corresponding to its insertion after node p, is obtained by removing arcs (0, m), (m, n) and (p, q), and replacing them with arcs (0, n), (p, m) and (m, q), as illustrated in Figure 2.

3.3.2 Inter-route improvement. In this step, a set of routes is obtained by using further local search procedures. These procedures are based on the so-called inter-route one-exchange, two-exchanges, three exchanges, and two consecutive vertices exchanges, illustrated in Figures 3-7, respectively.

For each node m (belonging to route a), the one-exchange corresponding to its insertion after node p (belonging to route b), is obtained by removing arcs (l, m), (m, n) and (p, q), and replacing them with arcs (l, n), (p, m) and (m, q), as illustrated in Figure 3.
For each node m (on route a), the two-exchanges, (1, 1) corresponding to its exchange with node q (on route b), are obtained by re- moving arcs (l, m), (m, n), (p, q) and (q, r), and replacing them with arcs (l, q), (q, n), (p, m) and (m, r), as illustrated in Figure 4.

For nodes q and r (belonging to route b), the two exchanges (2, 0) corresponding to its insertion between nodes m and n (belonging to route a), is obtained by removing arcs (m, n), (p, q) and (r, s), and replacing them with arcs (m, q), (r, n) and (p, s), as illustrated in Figure 5.

For nodes q, r and s (belonging to route b), the three exchanges (3, 0) corresponding to its insertion between nodes m and n (belonging to route a), is obtained by removing arcs (m, n), (p, q) and (s, t), and replacing them with arcs (m, q), (s, n) and (p, t), as illustrated in Figure 6.

For two consecutive nodes m and n (on route a), the two consecutive vertices exchanges corresponding to its exchange with two consecutive nodes q and r (on route b), are obtained

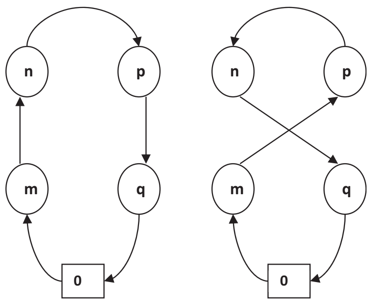

Figure 1.
Example of intra-route two-exchanges

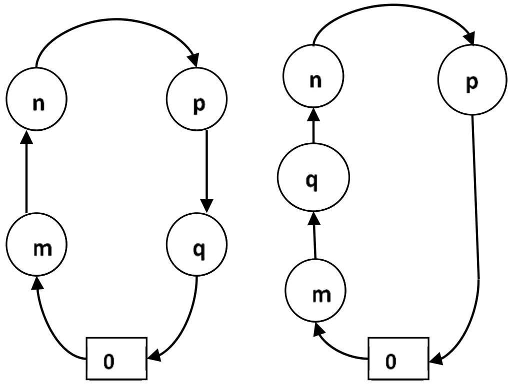

Figure 2.
Example of intra-route three-exchanges

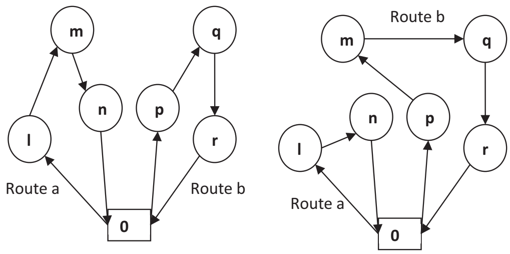

Figure 3.
Example of inter-route one exchange, (1, 0)

by re- moving arcs (l, m), (m, n), (n, o), (p, q) (q, r) and (r, s), and re- placing them with arcs (l, q), (q, r), (r, o), (p, m), (m, n) and (n, s), as illustrated in Figure 7.

3.3.3 Search procedure. A search procedure is designed to search for a better solution. From the results of extensive experiments which are not shown here, we are aware that the implementation sequence of intra- route and inter-route improvement procedure might have impacts on the quality of a solution. The improvement procedures mentioned above include intra- route swap exchange, Insert_1, inter-route one-exchanges, two exchanges and two consecutive vertices exchanges. The possible permutations of different improvement procedures are 240. Therefore, a loop procedure consisting of arranging the possible sequences of intra-route and inter-route improvement is applied to the solution obtained in the initial solution construction phase and the PF procedure mentioned before is also applied during the search process to avoid the route in- feasibility. The purpose of this loop procedure is in a sense similar to that of the Tabu search method to escape from a local

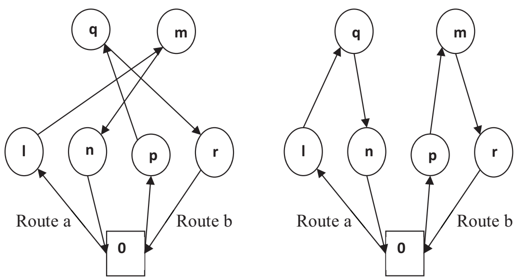

Figure 4.
Example of inter-route two exchanges, (1, 1)

minimum. Once a better solution is found after completing the improvement phase, the best solution record is updated. We repeat the above improvement processes until all possible permutations of different improvement procedures have been implemented.

3.4 Post-Optimization

Two procedures, reinsertion and cost-comparison, are used in Post-optimization. Re-insertion is used to decrease the cost of less-than-truckload carriers. It tries to reinsert any customers served by the Less-than-truckload carriers in the current routes. If this solution improves upon the current one, it is accepted. Let L be the ordered list of customers served by the Less-than-truckload carriers. Starting from the top of L, the insertion is achieved by exchanging customer j served by a Less-than-truckload carrier with customer k in a route s.

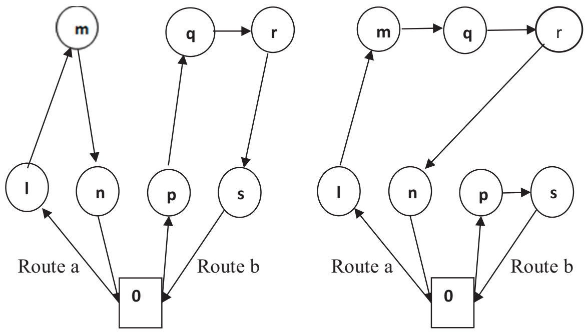

Figure 5. Example of inter-route two exchanges, (2, 0)

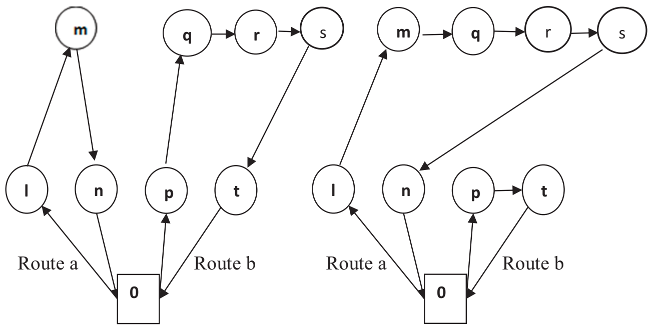

Figure 6. Example of inter-route three exchanges, (3, 0)

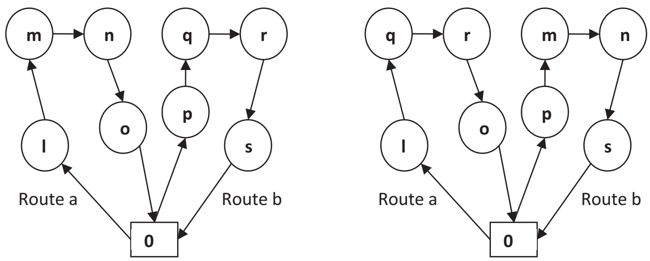

Figure 7. Example of inter-route 2 consecutive vertices exchanges, (2, 2)

Cost-comparison is used to calculate and compare the cost between the current multiple trip and used another car. Different scenarios will be considered. For example, the current optimal solution is one car with three trips, and some customers are served by less-than-truckload carriers. Another scenario is one car with two trips, another car with one trip or some customers are served by less-than-truckload carriers. A scenario with a lower cost will be used as the best solution.

4. Computational results

As our scenario has not been explored in the literature, there are no standard instances available for our problem. As mentioned in the introduction, Wu *et al.* (2017) consider a vehicle routing problem with time constraints while less-than-truckload carriers are available. However, only a single trip is considered in the work of Wu *et al.*

In this study, we simultaneously consider the multiple trip vehicle routing problems with time windows while less-than-truckload carriers are available. Since each vehicle can perform several trips during the working day in our study, the problem studied by Wu *et al.* is a special case of our problem. Our results are supposed to be better than those of Wu's work. Hence, we use the test problems proposed by Wu *et al.* (2017) to evaluate the efficiency and accuracy of our algorithm.

The vehicle capacities and relevant costs for test problems are shown in Tables I and II, and the detailed coordinates, demands and time windows of customers are given in the Appendix. The solutions produced by the heuristic algorithm are compared to the optimal results from the mathematical model mentioned in Section 2. The heuristic algorithm was written in FORTRAN language, and the mathematical model was solved using the software

Table I. Vehicle capacities and relevant costs for test problems with five customers

Problem	Vehicle capacities (cwt)	Fixed cost ($)	Variable costs ($)
1-1	30	50	TL $1/per mile, LTL $5/per mile
1-2	40	50	TL $1/per mile, LTL $5/per mile
1-3	50	50	TL $1/per mile, LTL $5/per mile
1-4	60	50	TL $1/per mile, LTL $5/per mile
1-5	70	50	TL $1/per mile, LTL $5/per mile
1-6	80	50	TL $1/per mile, LTL $5/per mile
1-7	90	50	TL $1/per mile, LTL $5/per mile
1-8	100	50	TL $1/per mile, LTL $5/per mile
1-9	110	50	TL $1/per mile, LTL $5/per mile
1-10	120	50	TL $1/per mile, LTL $5/per mile
2-1	30	50	TL $1/per mile, LTL $5/per mile
2-2	40	60	TL $1/per mile, LTL $5/per mile
2-3	50	70	TL $1/per mile, LTL $5/per mile
2-4	60	80	TL $1/per mile, LTL $5/per mile
2-5	70	90	TL $1/per mile, LTL $5/per mile
2-6	80	100	TL $1/per mile, LTL $5/per mile
2-7	90	110	TL $1/per mile, LTL $5/per mile
2-8	100	120	TL $1/per mile, LTL $5/per mile
2-9	110	130	TL $1/per mile, LTL $5/per mile
2-10	120	140	TL $1/per mile, LTL $5/per mile

Notes: TL: Truckload (a private truck); LTL: less-than-truckload (an outside carrier); Cwt: A hundredweight. In North America, a hundredweight is equal to 100 pounds and is also known as a short hundredweight. In the UK, a hundredweight is 112 pounds and is also known as a long hundredweight

Problem	Vehicle capacities (cwt)	Fixed cost ($)	Variable costs ($)
3-1	250	250	TL $1/per mile, LTL $5/per mile
3-2	260	250	TL $1/per mile, LTL $5/per mile
3-3	270	250	TL $1/per mile, LTL $5/per mile
3-4	280	250	TL $1/per mile, LTL $5/per mile
3-5	290	250	TL $1/per mile, LTL $5/per mile
3-6	300	250	TL $1/per mile, LTL $5/per mile
3-7	310	250	TL $1/per mile, LTL $5/per mile
3-8	320	250	TL $1/per mile, LTL $5/per mile
3-9	330	250	TL $1/per mile, LTL $5/per mile
3-10	340	250	TL $1/per mile, LTL $5/per mile
4-1	100, 100	150, 150	TL $1/per mile, LTL $5/per mile
4-2	105, 105	150, 150	TL $1/per mile, LTL $5/per mile
4-3	110, 110	150, 150	TL $1/per mile, LTL $5/per mile
4-4	115, 115	150, 150	TL $1/per mile, LTL $5/per mile
4-5	120, 120	150, 150	TL $1/per mile, LTL $5/per mile
4-6	125, 125	150, 150	TL $1/per mile, LTL $5/per mile
4-7	130, 130	150, 150	TL $1/per mile, LTL $5/per mile
4-8	135, 135	150, 150	TL $1/per mile, LTL $5/per mile
4-9	140, 140	150, 150	TL $1/per mile, LTL $5/per mile
4-10	145, 145	150, 150	TL $1/per mile, LTL $5/per mile

Notes: TL: Truckload (a private truck); LTL: less-than-truckload (an outside carrier); Cwt: A hundredweight. In North America, a hundredweight is equal to 100 pounds and is also known as a short hundredweight. In the UK, a hundredweight is 112 pounds and is also known as a long hundredweight

Table II. Vehicle capacities and relevant costs for test problems with ten customers

LINGO version 16.0. Both of them were implemented on a PC with an Intel Core i5 2.7 GHz processor. Computational results on test problems are reported in Tables III-VI, respectively.

Table III summarizes the results for five customers. Our heuristic algorithm obtains optimal solutions. As shown in Table III, both the mathematical model and the heuristic algorithm yield the same total cost in 20 instances. By comparing our results with the results from a single trip, we can find that our heuristic algorithm outperforms the single trip in 16 instances except for 1-9, 1-10 and 2-9, 2-10. The average percentage of savings from our algorithm is about 27.41 per cent. Combining the total cost in Table III and the vehicle capacity in Table I, we plot Figures 8 and 9 for problems 1-1-1-10 and 2-1-2-10, respectively. From Figure 8, we can find that there is a negative relationship between total costs and vehicle capacities. It means that the higher the vehicle capacity, the lower the total cost. It makes sense since the freight charged by a less-than-truckload carrier is usually much higher than the cost of a private truck.

From Figure 9, we can find that there is a negative relationship between total costs and the vehicle capacities from the range 30 to 60. With similar reasoning in Figure 8, it makes sense since the freight charged by a less-than-truckload carrier is usually much higher than the cost of a private truck. There is a positive relationship between total costs and vehicle capacities from the range 70 to 120. The reason for this is that in this range, the increase in fixed cost is much higher than that of the cost savings from increasing capacity.

Table IV summarizes the results for ten customers. For problems 3-1∼3-10, our heuristic algorithm obtains the optimal solutions in 10 instances. As shown in Table VI, both the mathematical model and the heuristic algorithm yield the same total cost and the same route

Table III.
Summary results for five customers

Problem	Optimal solution (single trip)		Optimal solution (multi-trip)		Heuristics (multi-trip)		
	Total Costs($)	CPU Time (seconds)	Total Costs($)	CPU Time (seconds)	Total Costs($)	CPU Time (seconds)	(%) Deviation
1-1	346.8	1	276.51	less than 1 sec	276.51	0.5	0
1-2	346.7	1	204.15	less than 1 sec	204.15	0.5	0
1-3	289.32	1	191.13	1	191.13	0.5	0
1-4	289.32	1	164.13	2	164.13	0.5	0
1-5	260.71	1	164.13	2	164.13	0.5	0
1-6	241.09	1	164.13	2	164.13	0.5	0
1-7	183.71	1	161.88	2	161.88	0.5	0
1-8	183.71	1	161.88	2	161.88	0.5	0
1-9	155.1	1	155.11	2	155.11	0.5	0
1-10	155.1	1	155.11	2	155.11	0.5	0
2-1	346.8	1	276.51	less than 1 sec	276.51	0.5	0
2-2	356.7	1	214.15	less than 1 sec	214.15	0.5	0
2-3	309.32	1	211.13	1	211.13	0.5	0
2-4	319.32	1	194.13	2	194.13	0.5	0
2-5	300.71	1	204.13	2	204.13	0.5	0
2-6	291.09	1	214.13	2	214.13	0.5	0
2-7	243.71	1	221.88	2	221.88	0.5	0
2-8	253.71	1	231.88	3	231.88	0.5	0
2-9	235.1	1	235.11	3	235.11	0.5	0
2-10	245.1	1	245.11	3	245.11	0.5	0

Table IV.
Summary results for ten customers

Problem	Optimal solution (single trip)		Optimal solution (multi-trip)		Heuristics (multi-trip)		
	Total Costs($)	CPU Time (seconds)	Total Costs($)	CPU Time (seconds)	Total Costs($)	CPU Time (seconds)	(%) Deviation
3-1	549.22	131	549.23	2498	549.23	1.3	0
3-2	512.86	294	512.87	2321	512.87	1.4	0
3-3	512.86	280	512.87	1950	512.87	1.5	0
3-4	512.86	292	512.87	2189	512.87	1.5	0
3-5	512.86	351	512.87	1797	512.87	1.6	0
3-6	512.86	227	512.87	1896	512.87	1.5	0
3-7	512.86	284	512.87	2123	512.87	1.5	0
3-8	512.86	303	512.87	1690	512.87	1.6	0
3-9	512.86	293	512.87	1741	512.87	1.6	0
3-10	512.86	301	512.87	1939	512.87	1.7	0
4-1	704.2	788	498.53	17203	498.53	7.2	0
4-2	696.28	1352	448.91	16411	448.91	7.6	0
4-3	659.03	1128	448.91	18085	448.91	7.8	0
4-4	659.03	1733	448.91	23252	448.91	7.6	0
4-5	655.46	3031	448.91	7404	448.91	7.7	0
4-6	607.36	3929	446.29	11882	448.91	8.4	0.587
4-7	607.36	4360	446.29	7543	448.91	8.3	0.587
4-8	568.81	3483	446.29	13739	448.91	8.5	0.587
4-9	568.81	5520	442.27	15173	448.91	9.2	1.5
4-10	568.81	8783	442.27	13357	448.91	11.4	1.5

Problem	Optimal Solution	Heuristic solution
1-1	1-3-7-6-7-4-2-7 L5	1-3-7-6-7-4-2-7 L5
1-2	1-5-7-6-4-7-3-2-7	1-5-7-6-4-7-2-3-7
1-3	1-3-7-5-7-2-4-6-7	1-3-7-5-7-2-4-6-7
1-4	1-3-2-4-7-5-6-7	1-3-2-4-7-5-6-7
1-5	1-4-2-3-7-6-5-7	1-4-2-3-7-6-5-7
1-6	1-4-2-3-7-6-5-7	1-4-2-3-7-6-5-7
1-7	1-3-7-2-4-6-5-7	1-3-7-2-4-6-5-7
1-8	1-2-4-6-5-7-3-7	1-2-4-6-5-7-3-7
1-9	1-5-6-4-2-3-7	1-5-6-4-2-3-7
1-10	1-5-6-4-2-3-7	1-5-6-4-2-3-7
2-1	1-3-7-6-7-4-2-7 L5	1-3-7-6-7-4-2-7 L5
2-2	1-5-7-6-4-7-3-2-7	1-5-7-6-4-7-3-2-7
2-3	1-3-7-5-7-2-4-6-7	1-3-7-5-7-2-4-6-7
2-4	1-3-2-4-7-5-6-7	1-3-2-4-7-5-6-7
2-5	1-4-2-3-7-6-5-7	1-4-2-3-7-6-5-7
2-6	1-4-2-3-7-6-5-7	1-4-2-3-7-6-5-7
2-7	1-3-7-2-4-6-5-7	1-3-7-2-4-6-5-7
2-8	1-2-4-6-5-7-3-7	1-2-4-6-5-7-3-7
2-9	1-5-6-4-2-3-7	1-5-6-4-2-3-7
2-10	1-5-6-4-2-3-7	1-5-6-4-2-3-7

Note: *1 and 7 stand for the warehouse

Table V.
Detailed results of test problems with five customers

Problem	Optimal Solution	Heuristic solution
3-1	1-10-2-4-9-6-5-7-11-12 L3, L8	1-10-2-4-9-6-5-7-11-12 L3, L8
3-2	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-3	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-4	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-5	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-6	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-7	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-8	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-9	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
3-10	1-3-10-2-4-9-6-5-7-11-12 L8	1-3-10-2-4-9-6-5-7-11-12 L8
4-1	1-10-2-4-9-12-6-5-12-7-11-12 L3, L8	1-10-2-4-9-12-6-5-12-7-11-12 L3, L8
4-2	1-10-2-4-9-12-6-5-7-12-3-11-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-3	1-10-2-4-9-12-6-5-7-12-3-11-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-4	1-10-2-4-9-12-6-5-7-12-3-11-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-5	1-10-2-4-9-12-6-5-7-12-3-11-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-6	1-11-10-2-4-12-6-5-7-3-12-9-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-7	1-11-10-2-4-12-6-5-7-3-12-9-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-8	1-11-10-2-4-12-6-5-7-3-12-9-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-9	1-10-2-4-9-12-6-5-7-11-12-3-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8
4-10	1-10-2-4-9-12-6-5-7-11-12-3-12 L8	1-10-2-4-9-12-6-5-7-12-3-11-12 L8

Note: *1 and 12 stand for the warehouse

Table VI.
Detailed results of test problems with ten customers

sequence in 10 instances. As to problems 4-1 and 4-10, our heuristic algorithm also obtains the optimal solutions in five instances. From the computational experiments, we found that the selection of customers served by the LTL carriers, and the initial solution have a great impact on whether an optimal solution can be reached.

Table IV shows that the solution time for the mathematical model increased dramatically with the size of the problem. In general, the solution time for problem 4-1∼4-10 is range from 2 to 6 h. It takes more than 6 h to solve the problem 4-4. Notice that the execution time reported here doesn't include the time for sub- tour breaking. Computationally, exact algorithms for the VRP are restricted to solving problems of only up to about 50 customers (Desaulniers, 2010). Even though the Branch-and-price-and-cut is used for solving the problem, it is still difficult to find the optimal solution in reasonable computing time. On the other side, our heuristic algorithm requires little time to solve the problem. The solving time of a problem is ranged from 0.5 to 11.4 s. The CPU time of test problems is not very sensitive to the problem size. From Tables III and IV, we find that the heuristic algorithm obtains the optimal or near-optimal solutions. The worst case of deviation from the optimum is only 1.5 per cent, and the average percentage deviation from the optimum for the forty test problems is 0.119 per cent and the execution time for all test problems is less than 12 s. It is an encouraging result in terms of both time and accuracy.

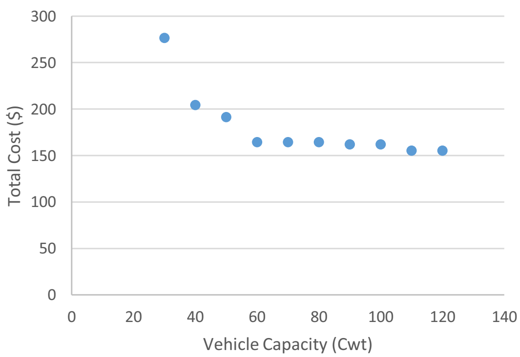

Figure 8.
The relationship between vehicle capacity and the total cost for problems 1-1-1-10

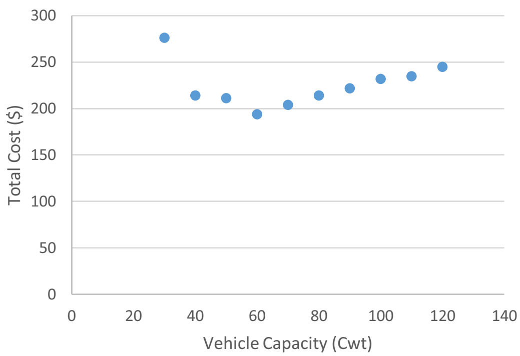

Figure 9.
The relationship between vehicle capacity and the total cost for problems 2-1∼2-10

5. Conclusions

Vehicle routing plays a central role in logistics management. In this paper, we introduced a real world new problem, the MTVRPTW and the possible use of a less-than-truckload carrier to satisfy customer demands. Our new problem is a generalization of the VRPTW and the MTVRPTW and the HVRP with external carriers. To the best of our knowledge, the scenario has never been explored in the literature. We developed both the mathematical model and the heuristic algorithm. In all, 40 test problems developed by Wu *et al.* were examined with our heuristics. The results obtained from our mathematical model and heuristic algorithm outperform the results from the exact solution method by Wu *et al.* The results are encouraging as our algorithm obtains the optimal or near-optimal solutions in an efficient way in terms of time and accuracy.

As for future research, a wide range of test instances should be built and tested. Furthermore, it would be interesting to see if other intelligent optimization techniques, such as Tabu Search, Genetic Algorithms, Ants Colony, Simulated Annealing and Neural Networks, can be used to solve this problem and even provide better results.

References

Agarwal, Y.K. (1985), *Vehicle Routing with Limited Fleet and Common Carrier Option*, TIMS/ORSA Joint National Meeting, Boston.

Azi, N., Gendreau, M. and Potvin, J.Y. (2007), "An exact algorithm for a single vehicle routing problem with time windows and multiple routes", *European Journal of Operational Research*, Vol. 178 No. 3, pp. 755-766.

Ball, M.O., Golden, A., Assad, A. and Bodin, L.D. (1985), "Planning for truck fleet size in the presence of a common-carrier option", *Decision Sciences*, Vol. 14 No. 1, pp. 103-120.

Bramel, J. and Simchi-Levi, D. (1996), "Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows", *Operations Research*, Vol. 44 No. 3, pp. 501-509.

Bräysy, G., Hasle, G. and Dullaert, W. (2004), "A multi-start local search algorithm for the vehicle routing problem with time windows", *European Journal of Operational Research*, Vol. 159 No. 3, pp. 586-605.

Chiang, W.C. and Russell, R.A. (1997), "A reactive tabu search metaheuristic for the vehicle routing problem with time windows", *INFORMS Journal on Computing*, Vol. 9 No. 4, pp. 417-430.

Chu, C.W. (2005), "A heuristic algorithm for the truckload and less-than-truckload problem", *European Journal of Operational Research*, Vol. 165 No. 3, pp. 657-667.

Cordeau, J.F., Laporte, G. and Mercie, A. (2001), "A unified tabu search heuristic for vehicle routing problems with time windows", *Journal of the Operational Research Society*, Vol. 52 No. 8, pp. 928-936.

Cordone, R. and Wolfler Calvo, R. (2001), "A heuristic for the vehicle routing problem with time windows", *Journal of Heuristics*, Vol. 7, pp. 107-129.

Demir, E., Huckle, K., Syntetos, A., Lahy, A., and Wilson, M. (2019), "Vehicle routing problem: past and future", *Contemporary Operations and Logistics*, Cham, Palgrave Macmillan, Vol. 7, pp. 97-117.

Desaulniers, G. (2010), "Branch-and-price-and cut for the split-delivery vehicle routing problem with time windows", *Operations Research*, Vol. 58 No. 1, pp. 179-192.

Gendreau, M. and Tarantilis, C.D. (2010), *Solving Large-Scale Vehicle Routing Problems with Time Windows: The State-of-the-Art*, CIRRELT, Canada.

Glover, F. (1986), "Future path for integer programming and links to artificial intelligence", *Computers and Operations Research*, Vol. 13 No. 5, pp. 533-549.

Glover, F. (1989), "Tabu search part I", *Journal on Computing*, Vol. 1, pp. 190-206.

Glover, F. (1990), "Tabu search part II", *Journal on Computing*, Vol. 2, pp. 4-32.

Glover, F. and Laguna, M. (1997), *Tabu Search*, Kluwer Academic Publishers, Netherlands.

Hernández, S. and Peeta, S. (2014), "A carrier collaboration problem for less-than- truckload carriers: characteristics and carrier collaboration model", *Transportmetrica A: Transport Science*, Vol. 10 No. 4, pp. 327-349.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of MI Press.

Ioannou, G., Kritikos, M. and Prastacos, G. (2001), "A greedy look-ahead heuristic for the vehicle routing problem with time windows", *Journal of the Operational Research Society*, Vol. 52 No. 5, pp. 523-537.

Klincewicz, J.G., Luss, H. and Pilcher, M.G. (1990), "Fleet size planning when outside carrier service are available", *Transportation Science*, Vol. 24 No. 3, pp. 169-182.

Koc, C., Bektas, T., Jabali, O. and Laporte, G. (2016), "Thirty years of heterogeneous vehicle routing", *European Journal of Operational Research*, Vol. 249, pp. 1-21.

Laporte, G. (1992), "The vehicle routing problem: an overview of exact and approximate algorithms", *European Journal of Operational Research*, Vol. 59 No. 3, pp. 345-358.

Laporte, G. and Nobert, Y. (1998), "Exact algorithm for vehicle routing problem", In: *Surveys in Combinatorial Optimization*, in Martello, S., Laporte, G., Minoux, M. and Riberio, C. (Eds), Amsterdam, North Holland, pp. 147-184.

Mester, D. and Bräysy, O. (2005), "Active guided evolution strategies for large-scale vehicle routing problems with time window", *Computers and Operations Research*, Vol. 32, pp. 1593-1614.

Potvin, J.Y. and Rousseau, J.M. (1993), "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows", *European Journal of Operational Research*, Vol. 66 No. 3, pp. 331-340.

Potvin, J.Y. and Rousseau, J.M. (1995), "An exchange heuristic for routing problems with time windows", *Journal of the Operational Research Society*, Vol. 46 No. 12, pp. 1433-1446.

Rochat, Y. and Taillard, E. (1995), "Probabilistic diversification and intensification in local search for vehicle routing", *Journal of Heuristics*, Vol. 1 No. 1, pp. 147-167.

Russell, R.A. (1995), "Hybrid heuristics for the vehicle routing problem with time windows", *Transportation Science*, Vol. 29 No. 2, pp. 156-166.

Schneider, L.M. (1985), "New era in transportation strategy", *Transportation Strategy*, pp. 118-126.

Schulze, J. and Fahle, T. (1999), "A parallel algorithm for the vehicle routing problem with time window constraints", *Annals of Operations Research*, Vol. 86, pp. 585-607.

Solomon, M. (1987), "Algorithms for the vehicle routing and scheduling problems with time window constraints", *Operations Research*, Vol. 35 No. 2, pp. 254-265.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J.Y. (1997), "A tabu search heuristic for the vehicle routing problem with soft time windows", *Transportation Science*, Vol. 31 No. 2, pp. 170-186.

Wu, C.S., Chu, C.W. and Hsu, H.L. (2017), "A heuristic algorithm of vehicle routing problem with time windows and less-than-truckload carrier selection", *Journal of Marine Science and Technology*, Vol. 25 No. 2, pp. 129-141.

Further reading

Fisher, M.L. (1995), "Vehicle routing", In: *Handbooks in Operations Research and Management Science 8: Network Routing*, Ball, M.E (Ed.), Science Publishers, Amsterdam, pp. 1-33.

Pang, K.W. (2011), "An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints", *Expert Systems with Applications*, Vol. 38 No. 9, pp. 11939-11946.

Appendix

Table AI.

No.	(X, Y)		q_i	e_i	l_i	L_i
1	0	0	0	0	480	0
2	11	6	11	0	480	62.65
3	−2	7	22	0	480	36.4
4	23	−5	16	100	200	117.69
5	−18	−18	37	50	250	127.28
6	6	−15	19	100	250	80.78
7	−22	−5	46	300	350	112.81
8	6	−18	63	400	450	94.87
9	12	−12	27	0	480	84.85
10	−9	23	43	0	480	123.49
11	13	16	36	0	480	103.08

Corresponding author
Ching-Wu Chu can be contacted at: cwchu@ntou.edu.tw