Journal of Marine Science and Technology

Volume 25 | Issue 2 Article 14

A HEURISTIC ALGORITHM OF VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND LESS-THAN-TRUCKLOAD CARRIER SELECTION

Chao-Sheng Wu
Department of Marketing and Logistics, Taipei Chengshih University of Science and Technology, Taiwan, R.O.C., cwchu@ntou.edu.tw

Ching-Wu Chu
Department of Shipping and Transportation Management, National Taiwan Ocean University, Taiwan, R.O.C.

Hsiu-Li Hsu
Department of Air & Sea Logistics and Marketing, Taipei College of Maritime Technology, Taiwan, R.O.C.

Recommended Citation
Wu, Chao-Sheng; Chu, Ching-Wu; and Hsu, Hsiu-Li (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: Iss. 2, Article 14.
DOI: 10.6119/JMST-014-1223-1
Available at: https://jmstt.ntou.edu.tw/journal/vol25/iss2/14

A HEURISTIC ALGORITHM OF VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND LESS-THAN-TRUCKLOAD CARRIER SELECTION

Acknowledgements

This research was partially supported by the National Science Council of Taiwan under grant NSC100-2410-H-019-002. The authors would like to thank the anonymous referees for their helpful comments.

Journal of Marine Science and Technology, Vol. 25, No. 2, pp. 129-141 (2017)
DOI: 10.6119/JMST-014-1223-1

A HEURISTIC ALGORITHM OF VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND LESS-THAN-TRUCKLOAD CARRIER SELECTION

Chao-Sheng Wu[1], Ching-Wu Chu[2], and Hsiu-Li Hsu[3]

Key words: vehicle routing problem with time windows, less-than-truckload, Heuristics, logistics, 0-1 integer programming.

ABSTRACT

The Vehicle Routing Problem with time windows is an important and practical problem for logistics managers. In reality, when facing fluctuations of demand, logistics managers may consider using an outside carrier to satisfy partial customer demand during the peak season. That is, the logistics managers must make a selection between a truckload (a private truck) and a less-than-truckload carrier (an outside carrier). Selecting the right mode to transport a shipment may bring significant cost savings to the company.

In this paper, we address the problem of routing a fixed number of trucks from a central warehouse to customers with known demand and time windows. A heuristic algorithm is developed for routing the private trucks with time windows and for selecting of less-than-truckload carriers by minimizing the total cost function. Computational results are encouraging and some suggestions for future research are presented.

I. INTRODUCTION

Vehicle routing with time windows (VRPTW) is an important and practical problem for logistics managers. In many sectors of the 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.

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. Our motivation for this study stems from observations on a local logistics company. This company owns different types of 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 five hours, so the delivery time window is a major concern. Furthermore, the company is facing fluctuations of 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. Since 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 carrier selection, a logistics manager can make a choice between a truckload (a private truck) 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. 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 trucks with limited capacity from a central warehouse to customers with known demand and time windows. The objective 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.

The literature of VRP with time windows can be classified into two categories, the exact method and heuristic algorithm. Although there are some exact methods (Laporte, 1992; Laporte

Paper submitted 06/25/14; revised 10/31/14; accepted 12/23/14. Author for correspondence: Ching-Wu Chu (e-mail:cwchu@ntou.edu.tw).
[1] Department of Marketing and Logistics, Taipei Chengshih University of Science and Technology, Taiwan, R.O.C.
[2] Department of Shipping and Transportation Management, National Taiwan Ocean University, Taiwan, R.O.C.
[3] Department of Air & Sea Logistics and Marketing, Taipei College of Maritime Technology, Taiwan, R.O.C.

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: (1) classical heuristics, and (2) metaheuristics.

Classical heuristics include construction and improvement approaches. Construction heuristics build a feasible route by iteratively inserting a customer into 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 in order 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) has reported good results with genetic algorithms.

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. Recently, 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. The latest work is a carrier collaboration problem for less-than-truckload carriers of Her-nández 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.

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 distance to cost and also incorporates the fixed cost of different types of trucks into the model. In addition, we simultaneously consider the vehicle routing problem with time windows and the selection of less-than-truckload carriers. A mathematical model is also proposed to represent and solve the problem. 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.

II. MATHEMATICAL MODEL

To simplify our analysis, we formulate our mathematical model based on the following assumptions:

1. A single warehouse system is considered; all trucks start at the warehouse and return back to the warehouse.
2. We restrict ourselves to delivery only.
3. The requirements of all the customers are known and each customer's requirement cannot exceed the truck capacity.
4. The time window of each customer is known.
5. 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.
6. 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 integer programming model and the relevant notations are given below:

$$\min \ z = \sum_{k=1}^{m} FC_k Y_{1k} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} C_{ijk} X_{ijk} + \sum_{i=2}^{n} CL_i L_i$$

Subject to

$$\sum_{k=1}^{m} Y_{1k} \le m, \tag{1}$$

$$\sum_{k=1}^{m} Y_{ik} + L_i = 1 \quad \left(i = 2, 3, \ldots, n \right), \tag{2}$$

$$\sum_{j=1}^{n} X_{ijk} = Y_{ik} \quad \left(i = 1, 2, \ldots, n; k = 1, 2, \ldots, m \right), \tag{3}$$

$$\sum_{j=1}^{n} X_{jik} = Y_{ik} \left(i = 1, 2, \ldots, n; k = 1, 2, \ldots, m \right), \quad (4)$$

$$\sum_{i=2}^{n} q_i Y_{ik} \le Q_k \quad \left(k = 1, 2, \ldots, m \right), \quad (5)$$

$$T_i + s_i + t_{ij} - M(1 - X_{ijk}) \le T_j$$
$$\left(i = 1, 2, \ldots, n;\ j = 2, 3, \ldots, n;\ k = 1, 2, \ldots, m \right), \quad (6)$$

$$e_i \le T_i \le l_i \quad \left(i = 2, 3, \ldots, n \right), \quad (7)$$

$$T_i + s_i + t_{i1} \le T_{max} \quad \left(i = 1, 2, \ldots, n \right), \quad (8)$$

$$\forall\, X_{ijk}, L_i, Y_{ik} \in \{ 0, 1 \}$$

i: $\{i = 1, 2, \ldots, n\}$, the index set of customers (let 1 denote the warehouse).

j: $\{j = 1, 2, \ldots, n\}$, the index set of customers.

k: $\{k = 1, 2, \ldots, m\}$, the index set of trucks.

n: the number of customers.

m: the number of trucks.

FC_k: 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*.

T_i: the start service time of customer *i*.

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*.

T_{max}: the maximum route time allowed for a vehicle.

$$X_{ijk} = \begin{cases} 1, & \text{if truck } k \text{ travels from customer } i \text{ to customer } j, \\ 0, & \text{otherwise,} \end{cases}$$

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

$$Y_{ik} = \begin{cases} 1, & \text{if the service of customer } i \text{ is satisfied by} \\ & \text{the private truck } k, \\ 0, & \text{otherwise} \end{cases}$$

The objective of this model is to route the private trucks and to make a selection of less-than-truckload carriers by minimizing a total cost function.

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

Constraint (2) defines that each customer is served either by 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 total load transported by a truck cannot exceed the truck capacity.

Constraints (6)-(8) ensure feasibility of the time schedule.

III. THE HEURISTIC ALGORITHM

In this section, we describe our algorithm, called VRPTWLTL, for solving the vehicle routing problem 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 its main steps separately.

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.

In order 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. It is obvious that we should arrange the customers in ascending order based on the freight charged by the less-than-truckload carrier and choose the customers with the lowest cost.

The detail for selecting the customers is described as follows:

(1) Calculate the total demand from all customers.
(2) Calculate the whole capacity of owned trucks.
(3) If the total demand from all customers is greater than the capacity of owned trucks, go to step (4), otherwise skip this procedure.
(4) Subtract the capacity of own trucks from the total demand, which is the unsatisfied truck capacity.
(5) Arrange the customers in ascending order based on the freight charged by the less-than-truckload carrier. Starting at the top of the list, do the following.
(6) Sum up the demand of each customer until the total demand is greater than the unsatisfied truck capacity. The corresponding customers will be the first group of candidates served by the less-than-truckload carrier.
(7) Calculate the total cost charged by the less-than-truckload carrier based on the first group of customers in step (6).
(8) Using the data in step (5), sort the customers in descending order based on the demand. Sum up the demand of customers until the total demand is greater than the unsatisfied truck capacity. The corresponding customers will be the

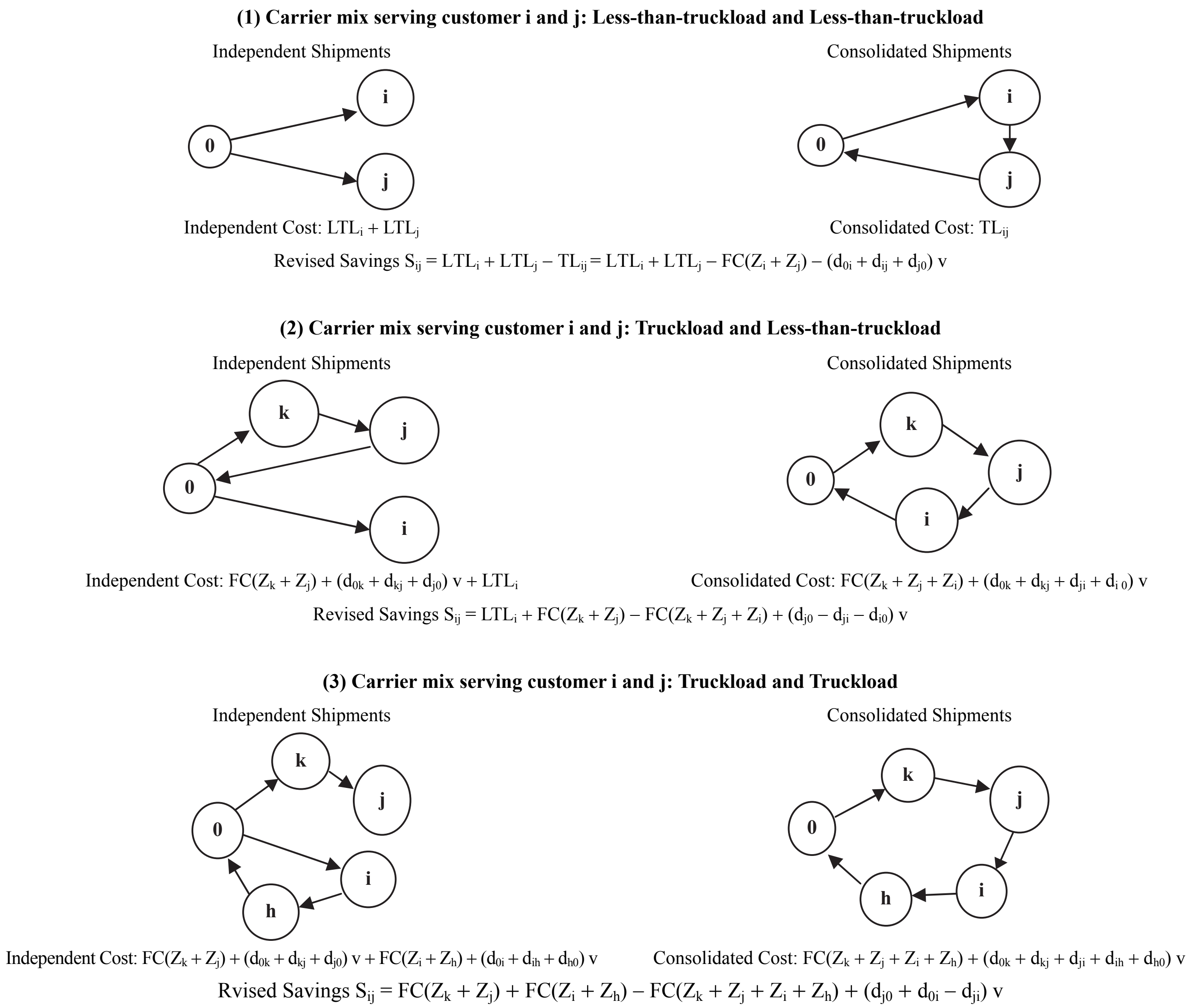

Fig. 1. Savings calculation from consolidating two customers.

second group of candidates served by the less-than-truckload carrier.

(9) Calculate the total cost charged by the less-than-truckload carrier based on the second group of customers in step (8).

(10) Choose a group of candidates with a lower total cost based on steps (7) and (9). The corresponding customers will be served by the less-than-truckload carrier, and the remaining customers in the list will be served by private trucks and will be used to construct an initial solution.

2. Initial Solution Construction

The initial solution construction step is composed of four procedures: construct, reverse, move, and time check.

The construct procedure is designed to generate the initial routes without taking time window constraints into consideration. The Clarke and Wright's savings algorithm is used to solve this problem by making two modifications. The first modification is a shift in criterion from the distance to cost. The second modification is a change in the savings calculation.

The mathematical relationship of the savings linking two customers is a function of the mix of a less-than-truckload carrier and a private truck. There are three possible mixes serving a pair of customers: (1) two less-than-truckload carriers, (2) a private truck and a less-than-truckload carrier, and (3) two private trucks.

Before explaining the revised savings calculation, we list the relevant notations as follows:

S_{ij}: savings from consolidating shipments to customer i and j into the same truck.

LTL_i: the total cost charged by the less-than-truckload carrier for serving customer i.

TL_{ij}: the total cost of a private truck that travels from warehouse to customer i, then from customer i to customer j and finally returns back to warehouse.
FC(Z): the fixed cost of the smallest truck that can serve a demand of Z.
d_{ij}: the distance from customer i to customer j.
v: the cost of traveling a mile for a private truck ($/per mile).

Fig. 1 illustrates the revised savings calculation from linking two customers under each of the three possible mixes.

The detail about the construct procedure is described as follows:

(1) Calculate the savings for all pairs customers based on revised savings scenario 1 in Fig. 1.
(2) Arrange the savings in descending order. Starting at the top of the list, do the following.
(3) Find the feasible link in the list which can be used to extend one of the two ends of the currently constructed route.
(4) If the route cannot be expanded further, terminate the route. Otherwise, choose the first feasible link in the list to start a new route.
(5) Repeat Steps (3) and (4) until no more links can be chosen.
(6) Output all the temporary single-customer routes (served by the less-than-truckload carriers) and multi-customer routes.
(7) Calculate the savings for single-customer routes based on revised savings scenario 2 in Fig. 1.
(8) Sort the savings in descending order. Starting at the top of the list, do the following.
(9) Find the feasible link in the current multi-customer routes which can be used to extend the route.
(10) If the route cannot be expanded further, terminate the route.
(11) Repeat Steps (9) and (10) until no more links can be chosen.
(12) Output all the routes.

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 *time check* is a procedure used to examine the time window feasibility of routes generated from the *construct*, *reverse,* or *move* procedures.

The *move* procedure is designed to achieve the time window feasibility of routes. Within the move procedure, the *time check* is used to examine the violation of the time window. If there is a violation of the time window constraint, the *move* procedure will be executed. If there are no violations of the time window, the program will skip the *move* procedure and go to the *refining* procedure directly. Let a_j denote the arrival time of the vehicle at the customer j. If $a_j < e_j$, the vehicle incurs a waiting time $w_j = e_j - a_j$. If $a_j > l_j$, the time window at customer j is violated. A do loop is applied to the current route to find any customer k with a time window violation and any customer j who satisfies two conditions (1) $a_j < e_j$, (2) $w_k \geq a_j - l_j$. This search do loop keeps moving the customer k after the customer j until all customers with time window have been handled.

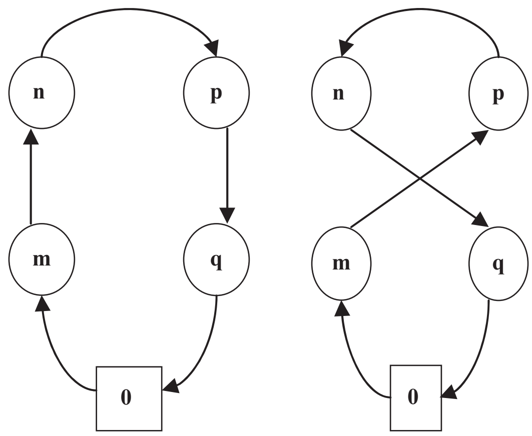

Fig. 2. An example of an intra-route swap exchange.

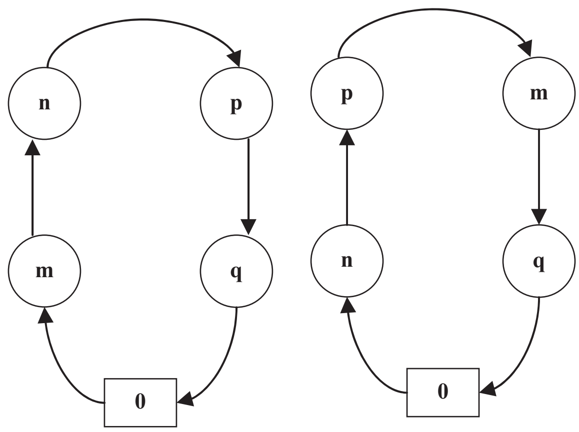

Fig. 3. An example of an intra-route Insert_1.

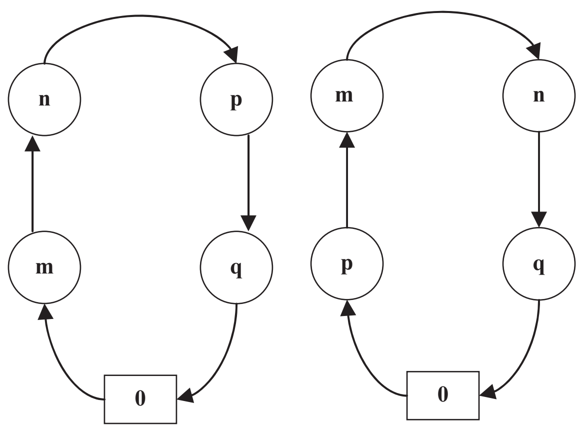

Fig. 4. An example of an intra-route Insert_2.

3. Refining Procedure

A refining procedure is applied to the solution obtained through the initial solution step. This procedure is composed

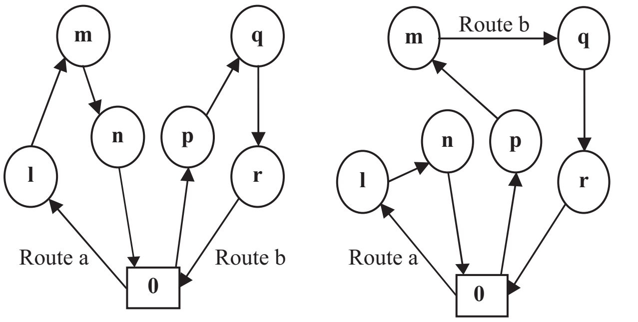

Fig. 5. An example of inter-route one-exchange.

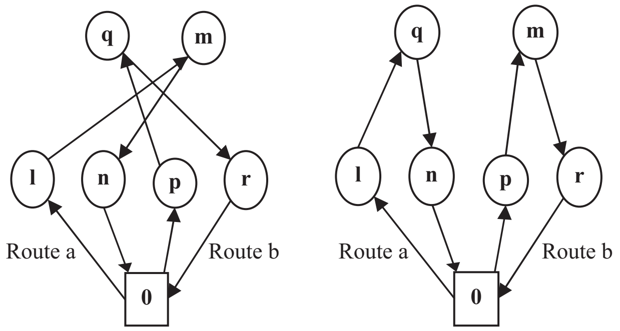

Fig. 6. An example of inter-route two-exchanges.

of a succession of intra-route and inter-route arc exchanges which are well known in the literature.

1) Intra-Route Improvement

In this step, each route is improved by using further local search procedures. These procedures include the swap exchange, Insert_1 and Insert_2, illustrated in Figs. 2-4, 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 Fig. 2.

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 Fig. 3.

For two consecutive nodes m and n, the Insert_2 corresponding to its insertion after node p, is obtained by removing arcs (0, m), (n, p) and (p, q), and replacing them with arcs (0, p), (p, m), and (n, q), as illustrated in Fig. 4.

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 and two consecutive vertices exchanges, illustrated in Figs. 5-7, 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 Fig. 5.

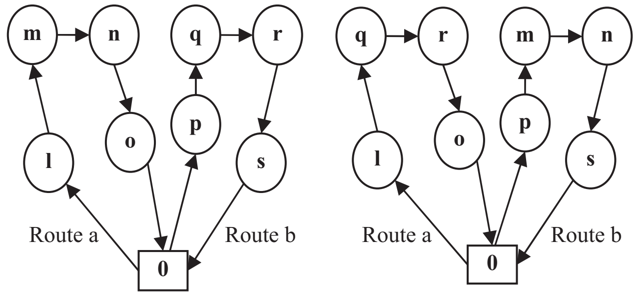

Fig. 7. An example of inter-route 2 consecutive vertices exchanges.

For each node m (on route a), the two-exchanges corresponding to its exchange with node q (on route b), are obtained by removing 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 Fig. 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 by removing arcs (l, m), (m, n), (n, o), (p, q) (q, r) and (r, s), and replacing them with arcs (l, q), (q, r), (r, o), (p, m), (m, n) and (n, s), as illustrated in Fig. 7.

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 solution.

The improvement procedures mentioned above include intra-route swap exchange, Insert_1, Insert_2, inter-route one-exchanges, two exchanges and two consecutive vertices exchanges. The possible permutations of six different improvement procedures are 720. 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 *time check* procedure mentioned before is also applied during the search process to avoid the route infeasibility. The purpose of this loop procedure is in a sense similar to that of the tabu search method to escape from a local 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.

4. Post-Optimization

Post-optimization 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 satisfying demand[j] ≤ demand[k] + unused truck capacity of the route s, and $LTL_k < LTL_j$.

Table 1. 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

Table 2. Vehicle capacities and relevant costs for test problems with ten customers.

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

TL: Truckload (a private truck); LTL: less-than-truckload (an outside carrier).

Table 3. Summary results for five customers.

Problem	Optimal Solution		Heuristics		% Deviation
	Total Costs	CPU Time	Total Costs	CPU Time	
1-1	346.8	1	346.8	0.03	0
1-2	346.7	1	346.8	0.03	0.0288
1-3	289.32	1	289.32	0.03	0
1-4	289.32	1	289.32	0.03	0
1-5	260.71	1	260.71	0.03	0
1-6	241.09	1	241.09	0.03	0
1-7	183.71	1	183.71	0.03	0
1-8	183.71	1	183.71	0.03	0
1-9	155.1	1	155.1	0.03	0
1-10	155.1	1	155.1	0.03	0
2-1	346.8	1	346.8	0.03	0
2-2	356.7	1	356.8	0.03	0.028
2-3	309.32	1	309.32	0.03	0
2-4	319.32	1	319.32	0.03	0
2-5	300.71	1	300.71	0.03	0
2-6	291.09	1	291.09	0.03	0
2-7	243.71	1	243.71	0.03	0
2-8	253.71	1	253.71	0.03	0
2-9	235.1	1	235.1	0.03	0
2-10	245.1	1	245.1	0.03	0

IV. COMPUTATIONAL RESULTS

Since there are no standard instances available for our problem, we generate forty test problems to evaluate the efficiency and accuracy of our algorithm. The coordinates and demands of all test problems are adopted from vehicle routing test banks. The vehicle capacities and relevant costs for forty test problems are shown in Tables 1 and 2, 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 LINGO version 10.0. Both of them were implemented on a PC with a 2000 MHz processor. Computational results on forty test problems are reported in Tables 3-6, respectively.

Table 3 summarizes the results for five customers. Except for problems 1-2 and 2-2, our heuristic algorithm obtains the optimal solutions. As shown in Table 3, both the mathematical model and the heuristic algorithm yield the same total cost in 18 instances. As to problems 1-2 and 2-2, our heuristic algorithm also obtains the near-optimal solutions since the percentage of deviation from the optimal solution is less than 0.03%.

Combining the total cost in Table 3 and the truck capacity in Table 1, we plot Figs. 8 and 9 for problems 1-1-1-10 and 2-1-2-10, respectively. We can find that there is a negative relationship between total costs and the truck capacities. It means that the higher truck capacity, the lower 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.

Table 4. Summary results for ten customers.

Problem	Optimal Solution		Heuristics		% Deviation
	Total Costs	CPU Time	Total Costs	CPU Time	
3-1	549.22	131	597.31	0.04	8.75
3-2	512.86	294	512.86	0.06	0
3-3	512.86	280	512.86	0.06	0
3-4	512.86	292	512.86	0.06	0
3-5	512.86	351	512.86	0.06	0
3-6	512.86	227	512.86	0.06	0
3-7	512.86	284	512.86	0.06	0
3-8	512.86	303	512.86	0.06	0
3-9	512.86	293	512.86	0.06	0
3-10	512.86	301	512.86	0.06	0
4-1	704.2	788	704.2	0.06	0
4-2	696.28	1352	704.2	0.06	1.13
4-3	659.03	1128	686.96	0.06	4.23
4-4	659.03	1733	686.96	0.06	4.23
4-5	655.46	3031	686.96	0.06	4.8
4-6	607.36	3929	607.36	0.06	0
4-7	607.36	4360	621.91	0.06	2.39
4-8	568.81	3483	568.81	0.06	0
4-9	568.81	5520	568.81	0.06	0
4-10	568.81	8783	568.81	0.06	0

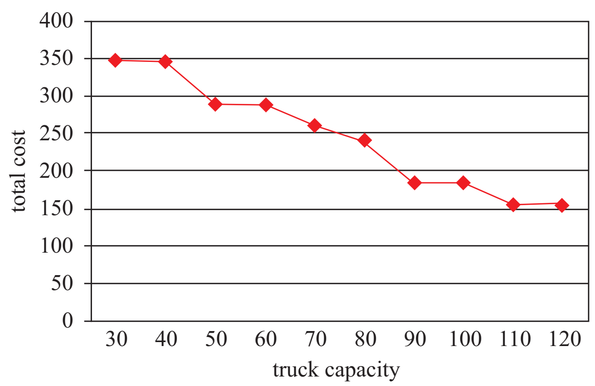

Fig. 8. The relationship between truck capacity and total cost for problems 1-1-1-10.

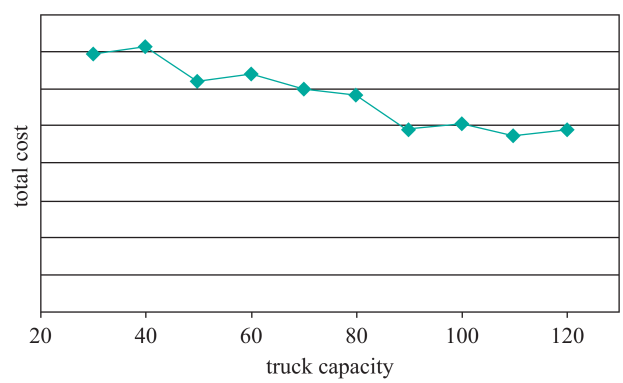

Fig. 9. The relationship between truck capacity and total cost for problems 2-1-2-10.

Table 4 summarizes the results for ten customers. For problems 3-1-3-10, our heuristic algorithm obtains the optimal solutions in 9 instances. As shown in Table 6, both the mathematical model and the heuristic algorithm yield the same total cost and the same route sequence in 9 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

Table 5. Detailed results of test problems with five customers.

Problem	Optimal Solution	Heuristic solution
1-1	Route 1: 1-2-4-1 LTL: 3, 5 and 6	Route 1: 1-2-4-1 LTL: 3, 5 and 6
1-2	Route 1: 1-6-4-1 LTL: 2, 3 and 5	Route 1: 1-2-4-1 LTL: 3, 5 and 6
1-3	Route 1: 1-2-4-6-1 LTL: 3 and 5	Route 1: 1-2-4-6-1 LTL: 3 and 5
1-4	Route 1: 1-2-4-6-1 LTL: 3 and 5	Route 1: 1-2-4-6-1 LTL: 3 and 5
1-5	Route 1: 1-3-2-4-6-1 LTL: 5	Route 1: 1-3-2-4-6-1 LTL: 5
1-6	Route 1: 1-4-6-5-1 LTL: 2 and 3	Route 1: 1-4-6-5-1 LTL: 2 and 3
1-7	Route 1: 1-2-4-6-5-1 LTL: 3	Route 1: 1-5-6-4-2-1 LTL: 3
1-8	Route 1: 1-2-4-6 5-1 LTL: 3	Route 1: 1-5-6-4-2-1 LTL: 3
1-9	Route 1: 1-3-2-4-6-5-1	Route 1: 1-5-6-4-2-3-1
1-10	Route 1: 1-3-2-4-6-5-1	Route 1: 1-5-6-4-2-3-1
2-1	Route 1: 1-2-4-1 LTL: 3, 5 and 6	Route 1: 1-2-4-1 LTL: 3, 5 and 6
2-2	Route 1: 1-6-4-1 LTL: 2, 3 and 5	Route 1: 1-2-4-1 LTL: 3, 5 and 6
2-3	Route 1: 1-2-4-6-1 LTL: 3 and 5	Route 1: 1-2-4-6-1 LTL: 3 and 5
2-4	Route 1: 1-2-4-6-1 LTL: 3 and 5	Route 1: 1-2-4-6-1 LTL: 3 and 5
2-5	Route 1: 1-3-2-4-6-1 LTL: 5	Route 1: 1-3-2-4-6-1 LTL: 5
2-6	Route 1: 1-4-6-5-1 LTL: 2 and 3	Route 1: 1-4-6-5-1 LTL: 2 and 3
2-7	Route 1: 1-2-4-6-5-1 LTL: 3	Route 1: 1-5-6-4-2-1 LTL: 3
2-8	Route 1: 1-2-4-6-5-1 LTL: 3	Route 1: 1-5-6-4-2-1 LTL: 3
2-9	Route 1: 1-3-2-4-6-5-1	Route 1: 1-5-6-4-2-3-1
2-10	Route 1: 1-3-2-4-6-5-1	Route 1: 1-5-6-4-2-3-1

the initial solution have a great impact on whether an optimal solution can be reached.

Table 4 shows that the solution time for the mathematical model increased dramatically with the size of the problem. It takes more than 2 hours to solve the problem 4-10. Notice that the execution time reported here doesn't include the time for subtour breaking. Computationally, exact algorithms for the VRP are restricted to solving problems of only up to about 25 customers. Even though the Lagrangean relaxation is used for solving the problem, it is still difficult to find the optimal solution in a reasonable computing time. On the other side, our heuristic algorithm requires little time to solve the problem. Every problem takes only less than a second. The CPU time of test problems is not very sensitive to the problem size.

From Tables 3 and 4, we find that the heuristic algorithm obtains the optimal or near-optimal solutions. The average percentage deviation from the optimum for the forty test problems is 0.639% and the execution time for all test problems is less than a second. It is an encouraging result in terms of both time and accuracy.

V. CONCLUSIONS

Vehicle routing plays a central role in logistics management. In this paper, we considered a vehicle routing problem with time windows and the possible use of a less-than-truckload carrier

Table 6. Detailed results of test problems with ten customers.

Problem	Optimal Solution	Heuristic solution
3-1	Route 1: 1-10-2-4-9-6-5-7-11-1 LTL: 3 and 8	Route 1: 1-2-4-9-6-5-11-10-3-1 LTL: 3 and 8
3-2	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-3	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-4	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-5	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-6	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-7	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-8	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-9	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
3-10	Route 1: 1-3-10-2-4-9-6-5- 7-11-1 LTL: 8	Route 1: 1-3-10-2-4-9-6-5-7-11-1 LTL: 8
4-1	Route 1: 1-4-9-6-5-1 Route 2: 1-11-10-2-1 LTL: 3, 7 and 8	Route 1: 1-5-6-9-4-1 Route 2: 1-11-10-2-1 LTL: 3, 7 and 8
4-2	Route 1: 1-10-2-4-9-1 Route 2: 1-6-5-7-1 LTL: 3, 8 and 11	Route 1: 1-5-6-9-4-1 Route 2: 1-11-10-2-1 LTL: 3, 7 and 8
4-3	Route 1: 1-5-6-9-4-2-1 Route 2: 1-3-10-11-1 LTL: 7 and 8	Route 1: 1-11-10-2-4-1 Route 2: 1-9-6-5-3-1 LTL: 7 and 8
4-4	Route 1: 1-5-6-9-4-2-1 Route 2: 1-11-10-3-1 LTL: 7 and 8	Route 1: 1-11-10-2-4-1 Route 2: 1-9-6-5-3-1 LTL: 7 and 8
4-5	Route 1: 1-3-10-2-4-9-1 Route 2: 1-5-7-11-1 LTL: 6 and 8	Route 1: 1-11-10-2-4-1 Route 2: 1-9-6-5-3-1 LTL: 7 and 8
4-6	Route 1: 1-2-4-9-6-5-1 Route 2: 1-7-11-10-1 LTL: 3 and 8	Route 1: 1-10-11-7-1 Route 2: 1-5-6-9-4-2-1 LTL: 3 and 8
4-7	Route 1: 1-5-6-9-4-2-1 Route 2: 1-7-11-10-1 LTL: 3 and 8	Route 1: 1-11-10-4-9-1 Route 2: 1-6-5-7-2-1 LTL: 3 and 8
4-8	Route 1: 1-6-5-7-3-1 Route 2: 1-9-4-2-10-11- 1 LTL: 8	Route 1: 1-6-5-7-3-1 Route 2: 1-9-4-2-10-11-1 LTL: 8
4-9	Route 1: 1-6-5-7-3-1 Route 2: 1-9-4-2-10-11-1 LTL: 8	Route 1: 1-6-5-7-3-1 Route 2: 1-9-4-2-10-11-1 LTL: 8
4-10	Route 1: 1-6-5-7-3-1 Route 2: 1-9-4-2-10-11-1 LTL: 8	Route 1: 1-6-5-7-3-1 Route 2: 1-9-4-2-10-11-1 LTL: 8

to satisfy customer demands. We developed both the mathematical model and the heuristic algorithm. A variety of test problems were examined with our heuristics. 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, it would be interesting to see if other intelligent optimization techniques, such as Genetic Algorithms, Ants Colony, Tabu Search and Neural Networks, can be used to solve this problem and even provide better results. Furthermore, a multi-trip vehicle routing problem with time windows and selecting less-than-truckload carriers is worthwhile to explore in the future.

ACKNOWLEDGEMENTS

This research was partially supported by the National Science Council of Taiwan under grant NSC100-2410-H-019-002. The authors would like to thank the anonymous referees for their helpful comments.

APPENDIX

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

REFERENCES

Agarwal, Y. K. (1985). Vehicle routing with limited fleet and common carrier option. TIMS/ORSA Joint National Meeting, Boston.

Azi, N., M. Gendreau and J. Y. Potvin (2007). An exact algorithm for a single vehicle routing problem with time windows and multiple routes. European Journal of Operational Research 178, 755-766.

Ball, M. O., A. Golden, A. Assad and L. D. Bodin (1985). Planning for truck fleet size in the presence of a common-carrier option. Decision Sciences 14, 103-120.

Bramel, J. and D. Simchi-Levi (1996). Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows. Operations Research 44, 501-509.

Bräysy, G., G. Hasle and W. Dullaert (2004). A multi-start local search algorithm for the vehicle routing problem with time windows. European Journal of Operational Research 159, 586-605.

Chiang, W. C. and R. A. Russell (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows. INFORMS Journal on Computing 9, 417-430.

Chu, C. W. (1995). A heuristic algorithm for the truckload and less-than-truckload problem. European Journal of Operational Research 165, 657-667.

Cordeau, J. F., G. Laporte and A. Mercie (2001). A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society 52, 928-936.

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

Fisher, M. L. (1995). Vehicle routing. In: Handbooks in Operations Research and Management Science 8: Network Routing, edited by Ball, M., Elsevier Science Publishers: Amsterdam, 1-33.

Gendreau, M. and C. D. Tarantilis (2010). Solving large-scale vehicle routing problems with time windows: The state-of-the-art. CIRRELT.

Glover, F. (1986). Future path for integer programming and links to artificial intelligence. Computers and Operations Research 13, 533-549.

Glover, F. (1989). Tabu Search Part I. Journal on Computing 1, 190-206.

Glover, F. (1990). Tabu Search Part II. Journal on Computing 2, 4-32.

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

Hernández, S. and S. Peeta (2014). A carrier collaboration problem for less-than-truckload carriers: characteristics and carrier collaboration model. Transportmetrica A: Transport Science 10, 327-349.

Holland, J. H. (1975). Adaptation in natural and artificial systems, Ann Arbor, MI: University of Michigan Press.

Ioannou, G., M. Kritikos and G. Prastacos (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. Journal of the Operational Research Society 52, 523-537.

Klincewicz, J. G., H. Luss and M. G. Pilcher (1990). Fleet size planning when outside carrier service are available. Transportation Science 24, 169-182.

Laporte, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 59, 345-358.

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

Mester, D. and O. Bräysy (2005). Active guided evolution strategies for large-scale vehicle routing problems with time window. Computers & Operations Research 32, 1593-16149.

Potvin, J. Y. and J. M. Rousseau (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research 66, 331-340.

Potvin, J. Y. and J. M. Rousseau (1995). An exchange heuristic for routing problems with time windows. Journal of the Operational Research Society 46, 1433-1446.

Rochat, Y. and E. Taillard (1995). Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1, 147-167.

Russell, R. A. (1995). Hybrid heuristics for the vehicle routing problem with time windows. Transportation Science 29, 156-166.

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

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

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35, 254-265.

Taillard, E., P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 31, 170-186.