REAL-TIME VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND SIMULTANEOUS DELIVERY/PICKUP DEMANDS

Mei-Shiang CHANG
Associate Professor
Department of Business Administration
Chung Hua University
No.30, Tung Shiang, Hsin Chu, Taiwan 30067
Fax:+886-3-5186570
E-mail: mschang@chu.edu.tw

Shyang-Ruey CHEN
Master
Department of Business Administration
Chung Hua University
No.30, Tung Shiang, Hsin Chu, Taiwan 30067
Fax:+886-3-5186570

Che-Fu HSUEH
Ph.D. Student
Department of Civil Engineering
National Central University
No. 300, Jung-Da Road, Wu-Chaun Li, Chung-Li, Taiwan 32054
E-mail: e01209@mail.trtc.com.tw

Abstract: The real-time vehicle routing problem with time windows and simultaneous delivery/pickup demands (RT-VRPTWDP) is formulated as a mixed integer programming model which is repeatedly solved in the rolling time horizon. The real-time delivery/pickup demands are served by capacitated vehicles with limited initial loads. Moreover, pickup services aren't necessarily done after delivery services in each route. A heuristic comprising of route construction, route improvement and tabu search is proposed. The route improvement procedure follows the general guidelines of anytime algorithm. Numerical examples made up by Gélinas were taken with modification for validation. Based on Taguchi orthogonal arrays approach, the optimal parameter setting for tabu search is set through experimentations on the RT-VRPTWDP. The results show that the proposed algorithm can efficiently decrease the total route cost.

Key Words: vehicle routing problem with time window, real time demands, anytime algorithm, Taguchi orthogonal arrays approach

1. INTRODUCTION

The vehicle routing problem with time windows (VRPTW) is a well-known NP-hard problem. Almost all VRPTW methods proposed are devoted to a static problem where all data are known before the route is constructed and do not change thereafter. The advancement of communication and information technology makes entrepreneurs more aware of the importance of just-in-time managerial strategies. In the past decade, express transshipment activities and e-commerce business have experienced a rapid growth. These developments have led to a gradual growth of a new class of problems, known as real-time routing and scheduling problems, where problem size and parameters change after the vehicle routes are constructed. Though strategies to deal with real time demands have been widely discussed, relevant models and algorithms for the VRPTW are scarce in the literature.

In our research, we describe a real-time vehicle routing problem with time windows and simultaneous delivery/pickup demands (RT-VRPTWDP), an extension to traditional VRPTW. Some requests are made after the routes are constructed. Each of the new requests must be assigned to an appropriate vehicle in real time. The uncertainty comes from the occurrence of the new service requests. There is no knowledge of further incoming requests. The problem size of RT-VRPTWDP changes therefore in real time. Furthermore, mixed vehicle routes with both delivery and pickup services are constructed in our study.

This paper is organized as follows. After introduction, a brief literature review is presented in Section 2. The RT-VRPTWDP is formulated as a mixed integer programming model in Section 3. Section 4 elaborates a proposed heuristic, which includes methods of route construction, route improvement and tabu search. In Section 5, 15 testing problems made up by Gélinas are taken with minor modifications for demonstration and validation. Finally, concluding remarks are given in the end.

2. LITERATURE REVIEW

The vehicle routing problem has been and is still an enriched research topic for researchers and practitioners. A large fraction of this work is concerned with static problems, that is, all order for all customers are known a priori. For a description of static vehicle routing problem, please see the recent survey of the routing problem (Ball et al., 1995; Fisher, 1995; Desrosiers et al., 1995).

During the past decade the number of published papers dealing with dynamic vehicle routing problem has been growing. Psaraftis (1995) examines the main issues in this area and provides a survey of the results found for various dynamic vehicle routing problems. Relevant models and algorithms for the real-time VRPTW are scarce in the literature. One of the earliest work on dynamic vehicle routing problem was from Bertsimas and van Ryzin (1991, 1993) that is essentially a generic mathematical model with waiting time as an objective function. A recent survey of dynamic vehicle routing is given by Psaraftis (1995), including the delivery of petroleum products or industrial gases, courier services, intermodal services, tramp ship operations, pickup and delivery services, management of container terminals.

Powell et al. (1995) provide an excellent survey on various dynamic vehicle routing problems such as the dynamic traffic assignment problem which consists in finding the optima routing of some goods from origin to the destination through a network of links which could have time-dependent capacities. Bertsimas and Simchi-Levi (1996) provide a survey of deterministic and static as well as dynamic and stochastic vehicle routing problems for which they examine the worst and average-case behaviors of the known algorithms for dynamic routing problems. Gendreau and Potvin (1998) is the most recent survey on the dynamic vehicle routing problem. They point out that it is relevant to consider several sources of uncertainty like cancellation of requests and service delays rather just to focus on uncertainty in the time-space occurrence of service requests. Gendreau et al. (1999) employed tabu search with parallel processing technique to solve a problem with real time demands and soft time window. The tabu search heuristic used in this work was originally designed for the static version of this problem and was therefore modified in order to deal with dynamic version. Shieh and May (1998) treated a VRPTW problem with real time demands and demonstrated with numerical examples. Liu (2000) tackled a traveling salesman problem with both time dependent travel times and real time demands. The time-dependency accounts for variations in travel speed caused by congestion. Larson (2001) provides the literature review dealing with the dynamic vehicle routing problem and related problems. The dynamic traveling repairman problem is extended to embrace advance request customers as well as immediate request customer. The capacitated vehicle routing problem with time windows is examined under varying levels of dynamism.

3. MODEL FORMULATION

Whenever the real time demands are generated, the routing schedule must change in response to new or altered requests. In the model we consider, there is an initial routing schedule that incorporates all works currently known. This routing schedule is adjusted as new work arrives, and can be improved providing this does not interfere with decisions that have already been committed to. To clarify the scope of the research, necessary assumptions are stated as follows:

1. There is a communication and transmission systems between the dispatcher and drivers. Through the communication system, the dispatcher informs the drivers which demand to serve next only when committing to that decision. Through the transmission system, the dispatcher faxes delivery/pickup notes to the drivers. Once a driver is en route to the next destination, however, he must necessarily serve this node. No diversion is allowed.
2. Real time demands are generated in the rolling time horizon. The real-time routing schedule needs to be solved with estimated computation time $\triangle$ units for initial solution. Real time demand means that the planning time span to transfer demand information from dispatcher to driver is short.

3. There is a time lag in the on-line dispatch system. Suppose τ is the time at which the deployment of vehicle routing begins and equal to the current wall-clock time plus $\triangle$ units.
4. Delivery problems (delivering goods from a depot to the customer) and pickup problems (picking up goods at the customer and bring back to a depot) are considered simultaneously. Moreover, pickup services aren’t necessarily done after delivery services in each route.
5. Delivery demand and pickup demand are irrelevant. The pickup goods cannot be directly delivered to the customers.
6. Uncertainty comes from a single source, namely the occurrence of new requests. There is no uncertainty associated with the customer locations and travel times.
7. In a least commitment strategy, the drivers are asked to wait at their current location if some waiting time is expected at the next customer. The latest possible time allows last changes to the planned routing schedule.
8. Demand forecast is not tackled by this research. Expected quantities and occurrence times of orders can improve the solution quality of the real-time routing problem, but increase the complexity of the real-time routing problem.

3.1 Notation

1. Parameters and constants

α : weight associated with link travel time in the objective function
β : weight associated with waiting time in the objective function
$\triangle$: estimated computation time for initial solution
τ : time to implement the results computed from the RT-VRPTWDP, which is usually set as the time of occurrence of new demands
e_i : lower end of the time window at node *i*
G_k : initial loads of vehicle k at depot
$\overline{G}_{ik}(\tau)$: remaining loads to deliver of vehicle *k* on arrival at critical node *i* at time τ
l_i : upper end of the time window at node *i*
M : a very big number
q_i : delivery demands at node *i*
$q_i^{'}$: pickup demands at node *i*
Q_k : capacity of vehicle *k*
$\overline{Q}_{ik}(\tau)$: free capacity of vehicle *k* on arrival node *i* at time τ
s_i : service time at node *i*
c_{ij} : travel time between nodes *i* and *j*

2. Set of Nodes

$\{0\}$: depot
$N_c(\tau)$: set of critical nodes at time τ; critical node is defined as the last node being served or scheduled to be served by each vehicle
$N_u(\tau)$: set of unassigned nodes at time τ
$N_{cu}(\tau)$: set of critical and unassigned nodes at time τ
$N_{u0}(\tau)$: set of depot and unassigned nodes at time τ
$N_{cu0}(\tau)$: set of depot, critical and unassigned nodes at time τ
$P_k(h)$: set of nodes which are served after node *h* by vehicle *k*

3. Set of vehicles

K : set of all vehicles
$K_0(\tau)$: set of vehicles in the depot at time τ
$\overline{K}_0(\tau)$: set of dispatched vehicles from the depot at time τ

4. Superscripts and subscripts

i,j,h : node designation
k : vehicle designation
k_i : vehicle heading to or currently at critical node *i*

5. Variable

a_i : time arriving at node *i*
a_{0k} : time for vehicle k to return the depot
$c(i,h,j)$: increased travel time after the insertion of node h between nodes *i* and *j*
C_{ij} : generalized cost accrued from nodes *i* and *j*
$C(0,h,0)$: total cost for serving node h from the depot
$C(i,h,j)$: total increased cost after the insertion of node h between nodes *i* and *j*
$\Delta C(h)$: total increased cost for serving node *h*
d_i : time to depart from node *i*
d_{0k} : time for vehicle *k* to depart from the depot
G_{ik} : remaining loads to deliver of vehicle *k* on arrival at node *i*
Q_{ik} : free capacity of vehicle *k* on arrival at node *i*
x_{ijk} : 1 if vehicle *k* departs node *i* toward node *j*; 0, otherwise
w_i : waiting time before departure at node *i*
$w(i,h,j)$: increased waiting time after the insertion of node *h* between nodes *i* and *j*

3.2 Mathematical Model

At time τ, the real-time vehicle routing problem with time windows and delivery/pickup demands can be formulated as a mixed integer problem as follows:

$$\min\ Z(\tau) = \alpha \sum_{i \in N_{cu0}(\tau)} \sum_{j \in N_{u0}(\tau)} \sum_{k} c_{ij} x_{ijk} + \beta \sum_{i \in N_{cu}(\tau)} w_i \tag{1}$$

The feasible region is represented by the following constraints.

Flow Conservation Constraints:

$$\sum_{j} \sum_{k} x_{ijk} = 1 \qquad \forall i \in N_{cu}(\tau) \tag{2}$$

$$\sum_{i} \sum_{k} x_{ijk} = 1 \qquad \forall j \in N_{u}(\tau) \tag{3}$$

$$\sum_{i} x_{ihk} - \sum_{j} x_{hjk} = 0 \qquad \forall\, h \in N_{u}(\tau), k \in K \tag{4}$$

$$\sum_j x_{ijk_i} = 1 \qquad \forall i \in N_c(\tau) \tag{5}$$

$$\sum_j x_{0jk} \le 1 \qquad \forall k \in K_0(\tau) \tag{6}$$

$$x_{ijk} \in \{0,1\} \qquad \forall i \in N_{cu0}(\tau), j \in N_{u0}(\tau), k \in K \tag{7}$$

Time Window and Departure Time Constraints:

$$e_i \le a_i \le l_i \qquad \forall i \in N_u(\tau) \tag{8}$$

$$a_{0k} \le l_0 \qquad \forall k \in K \tag{9}$$

$$d_i - (a_i + s_i) \ge 0 \qquad \forall i \in N_{cu}(\tau) \tag{10}$$

$$d_i - \tau \ge 0 \qquad \forall i \in N_c(\tau) \tag{11}$$

$$d_{0k} - \tau + [1 - x_{0jk}]\, M \ge 0 \qquad \forall j \in N_u(\tau), k \in K \tag{12}$$

Vehicle Capacity Constraints:

$$G_{ik} \ge q_i \quad \text{if} \quad x_{ijk} = 1 \qquad \forall i \in N_u(\tau), j \in N_u(\tau), k \in K \tag{13}$$

$$Q_{ik} \ge q_i' \quad \text{if} \quad x_{ijk} = 1 \qquad \forall i \in N_u(\tau), j \in N_u(\tau), k \in K \tag{14}$$

Definitional Constraints:

$$d_i + c_{ij} = a_j \quad \text{if} \quad x_{ijk} = 1 \qquad \forall i \in N_{cu}(\tau), j \in N_u(\tau), k \in K \tag{15}$$

$$d_{0k} + c_{ij} = a_j \quad \text{if} \quad x_{0jk} = 1 \qquad \forall j \in N_u(\tau), k \in K \tag{16}$$

$$d_i + c_{i0} = a_{0k} \quad \text{if} \quad x_{i0k} = 1 \qquad \forall i \in N_{cu}(\tau), k \in K \tag{17}$$

$$G_{jk} = G_{ik} - q_i \quad \text{if} \quad x_{ijk} = 1 \qquad \forall i \in N_u(\tau), j \in N_u(\tau), k \in K \tag{18}$$

$$G_{jk} = \overline{G}_{ik}(\tau) - q_i \quad \text{if} \quad x_{ijk} = 1 \qquad \forall i \in N_c(\tau), j \in N_u(\tau), k \in K \tag{19}$$

$$G_{jk} = G_k \quad \text{if} \quad x_{0jk} = 1 \qquad \forall j \in N_u(\tau), k \in K \tag{20}$$

$$Q_{jk} = Q_{ik} - q_i' + q_i \quad \text{if} \quad x_{ijk} = 1 \qquad \forall i \in N_u(\tau), j \in N_u(\tau), k \in K \tag{21}$$

$$Q_{jk} = \overline{Q}_{ik}(\tau) - q_i' + q_i \quad \text{if} \quad x_{ijk} = 1 \quad \forall i \in N_c(\tau), j \in N_u(\tau), k \in K \tag{22}$$

$$Q_{jk} = Q_k - G_k \quad \text{if} \quad x_{0jk} = 1 \qquad \forall j \in N_u(\tau), k \in K \tag{23}$$

$$w_i = d_i - (a_i + s_i) \qquad \forall i \in N_{cu}(\tau) \tag{24}$$

The objective of the RT-VRPTWDP, as shown in Eqn (1), is constructed as a weighted function of travel times for all links and waiting times before departure at all nodes. The respective weights are α and β with the relationships of $\alpha > \beta$ which is determined due to the fact: for each vehicle moving cost is generally higher than stopping cost because the former needs to pay for gasoline, depreciation, and additional social costs such as traffic congestion, air pollution and risk of traffic incidents whereas the latter only accounts for the depreciation.

Eqn (2) requires one vehicle leaving from critical or unassigned node i once. Eqn (3) denotes only one vehicle can arrive at unassigned node j once. Eqn (4) states for each unassigned node *h*, the entering vehicle must leave the node eventually. Eqn (5) requires that vehicle k_i that arrived or is approaching critical node *i* must also leave that node once. Note that vehicle k_i is known at time τ. Eqn (6) designates each vehicle can leave the depot at most once. Eqn (7) designates x_{ijk} as 0-1 integers; set x_{ijk} equal 1 if vehicle k departs node *i* toward node *j*, 0, otherwise.

Eqn (8) requires that for each node, the arrival time must be within the time window. Eqn (9) indicates that all vehicles must return back before the depot is closed. Eqn (10) requires that for critical or unassigned node *i*, departure time d_i must be greater than or equal to the completion time of service, $a_i + s_i$. The following two Eqns are about the constraints of dispatch timing. Eqn (11) indicates that for critical node *i*, departure time d_i must be greater than or equal to τ. Eqn (12) indicates that vehicles cannot leave the depot before time τ.

Eqn (13) states that for each vehicle, the remaining loads to deliver on arrival at node i must be greater than or equal to the delivery demands at this node. Eqn (14) indicates that for each vehicle, the free capacity on arrival at node *i* must be greater than or equal to the pickup demands at this node. Eqns (15)~(17) defines that for each node (including the depot) the arrival time is equal to the departure time from the previous node plus the travel time. Eqns (18)~(20) defines that the conservation of the delivery loads for each node. Eqns (21)~(23) defines that the conservation of the free capacity for each node. Eqn (24) computes the waiting time before departure at node *i*..

Note that real time demands may sometimes result in a situation where not all customers can be served within their time windows. Here we simply delete the customers that violate the time window constraints from set $N_u(\tau)$ and solve the real-time vehicle routing problem with delivery/pickup demands and time windows, problem (1), for the rest of customers.

4. SOLUTION ALGORITHM

The VRPTW problem is known to be NP-hard and when temporal dimension is incorporated, it becomes more difficult to be solved in a reasonable period by an exact solution especially for large problems. To take care of both computational efficiency and real time response requirement, a heuristic comprising of route construction and route improvement is proposed for the RT-VRPTWDP. In the following sections, preliminaries about insertion cost and tabu search are described in Section 4.1. In Section 4.2 a systematic solution procedure embedding route construction, route improvement and tabu search is illustrated. Efficient strategy proposed for route construction is elaborated in Section 4.3. The method of route improvement with tabu search is illustrated in Section 4.4.

4.1 Peliminaries

4.1.1 Calculation of Insertion Cost

According to the objective function shown in Eqn (1), the generalized cost C_{ij} accrued from nodes *i* and *j* can be defined below as a weighted function of travel time c_{ij} and waiting time for departure from node *i*, w_i.

$$C_{ij} = \alpha c_{ij} + \beta w_i \tag{25}$$

Consider a vehicle route, …..*i*,*j*,*j*+1,*j*+2,….,*j*+*n*,0, where 0 represents the depot. If we insert node *h*, the total increased cost $\Delta C(h)$ can be expressed in terms of nodes *i*, *h*, and *j* as

$$\Delta C(h) = \min(C(i,h,j), C(0,h,0)) \tag{26}$$

$$C(i,h,j) = \alpha c(i,h,j) + \beta w(i,h,j) \tag{27}$$

$$c(i,h,j) = c_{ih} + c_{hj} - c_{ij} \tag{28}$$

$$w(i,h,j) = w_i^{'} - w_i + \sum_{j \in P_k(h)} \left(w_j^{'} - w_j\right) \tag{29}$$

$$C(0,h,0) = c_{0h} + c_{h0} \tag{30}$$

where $C(i,h,j)$ represents the total increased cost after the insertion of node h between nodes *i* and *j*, $C(0,h,0)$ denotes the total cost for serving node h from the depot, and w_i, $w_i^{'}$ denote the waiting time for departure from node *i* before and after the insertion of node *h*. The total increased cost $C(i,h,j)$ is a weighted sum of increased travel time $c(i,h,j)$ and increased waiting time $w(i,h,j)$. The definitions of $c(i,h,j)$ and $w(i,h,j)$ can be expressed in Eqn (28) and (29). The last term of Eqn (29) is the total increased waiting cost along the route after the inserted node *h*.

4.1.2 Tabu Search

The tabu search heuristic was first introduced in Glover (1986). Starting from some initial solution, a neighborhood of the solution is generated through different classes of transformations. Then, the best solution in this neighborhood is selected as the new current solution, even if it is worse than the current solution. Since the current solution may deteriorate during the search, anti-cycling rules must be implemented. Thus, a memory (tabu list) is used to remember the recent search trajectory. In addition, diversification mechanisms can be used to help the method to escape from local optima and explore a broad portion of the search space. The details of tabu search are discussed in Section 4.4.

4.2 Unified Framework of Solution Procedure

This Section describes a unified framework of solution procedure for the RT-VRPTWDP. The main concept is to take care of real time information and in the meantime to improve the quality of the solution in responding to the ever-changing environment along the rolling time horizon. By constantly checking whether (1) departure time for critical node is up; (2) new demands have been generated; strategies to dispatching en route and/or on-call vehicles with the right time to the assigned customers, to reconstructing routes and to improving the quality of the existing routes are repeatedly applied.

Note that the heuristics for RT-VRPTWDP must be interrupted at checkpoints due to the requirement of real time response (Shieh and May, 1998). The real-time route improvement procedure follows the general guidelines of anytime algorithm (Zilberstein and Russell, 1996). If the earliest departure time of the critical nodes has arrived, the heuristic is interrupted at the checkpoints and outputs the current solution to the dispatcher. The main components of this solution procedure are briefly introduced:

First, we make the initial routing schedule according to known customers beforehand. All the static vehicle routing algorithms with time windows and simultaneous delivery/pickup demands can be adopted to find the initial solution. If new demands appear, we insert the new demands to the initial routing schedule. Finally, a better real-time routing schedule is repeatedly searched before earliest departure time of the critical nodes. In the real-time route improvement procedure, Or-opt, 3-opt*/2-opt*, and Swap-opt algorithms are adopted for obtaining an inferior improvement. The basic idea of these route improvement methods are stated as follows:

Or-opt: This exchange procedure is described in Or (1976). For a sequence of three consecutive customers, two consecutive customers or a single customer in a route is removed and inserted at another location within the same or within another route. However, some restrictions apply due to the constraints of vehicle capacity, time window, delivery loads and dispatch timing.

2-opt*: This is an extension of the 2-opt neighborhood for problem with multiple routes. Considering a link (i_1, j_1) on a route, a link (i_2, j_2) on another route, we replace them by two new links (i_1, j_2) and (i_2, j_1) if this feasible exchange can result in a lower cost. Note that a feasible exchange stated herein means that feasibility with respect to all the above constraints.

Swap-opt: This exchange procedure is described in Duhamel *et al*. (1997). Two customers are selected in the dame route or in two different routes, and exchange their position. For the same reason, feasible move are constrained by the side constraints.

For the sake of further improvement solution, the tabu search is integrated with Or-opt, 3-opt*/2-opt*, and Swap-opt algorithms.

4.3 Method for Real-Time Route Construction

For VRPTW problem, the insertion-type method has been proven effective in constructing static routes (Solomon, 1987). We assume it performs equally effective for the RT-VRPTWDP problems and therefore is adopted with modifications as follows.

Steps for Real-Time Route Construction

Step 1: Input data

Input real time delivery/pickup demands $\{q_i, q_i^{'}\}$ and estimated time for exploitation, τ .

Step 2: Classify customers and calculate relevant data

Step 2.1: Classify customers into critical nodes $N_c(\tau)$ and unassigned nodes $N_u(\tau)$.

Step 2.2: For dispatched vehicle *k*, calculate the remaining loads and free capacity at critical node *i*, G_{ik} and Q_{ik} , using Eqn (18) and Eqn (21).

Step 3: Find the optimal place and departure time for each unassigned node and calculate its accrued cost

For each unassigned node $u \in N_u(\tau)$, find the optimal place and departure time for insertion and calculate its accrued cost.

Step 3.1: Set *k*=1.

Step 3.2: If vehicle *k* is en route,

Step 3.2.1: Calculate insertion cost, $C(i,h,j)$, after checking the time window constraints, free capacity constraints and remaining load constraints for unassigned node *u* in all possible places along the moving route of vehicle *k*.

Step 3.2.2: Record the minimal insertion cost and the associated place for unassigned node u along the moving route of vehicle *k*.

Step 3.3: If *k*=*K*, continue. Otherwise, set *k*=*k*+1 and go to Step 3.2.

Step 3.4: If some vehicles are available for dispatching in the depot, calculate the accrued cost, $C(0,h,0)$, and check time window constraints for unassigned node *u* for the newly dispatched vehicle.

Step 3.5: Select the minimal insertion cost and its place for node *u* in all possible routes among vehicles either en route or being dispatched. If either capacity constraints or time constraints cannot be fulfilled, exclude that customer from the service list.

Step 3.6:If every unassigned node $u \in N_u(\tau)$ has been examined, continue. Otherwise, go back to Step 3.1 for next unassigned node *u*.

Step 4: Insert the newly accrued customer and update the relevant data

Update the system by inserting node *u* into the place with minimal accrued cost in an appropriate route (of vehicle *k**) and redefine relevant data such as departure time, arrival time, free capacity and remaining loads for each affected node. Once inserted, node *u* is then removed from $N_u(\tau)$.

Step 5: Stopping Check for Assignment

If set $N_u(\tau)$ is empty, enter the unified framework of solution procedure. Otherwise, do the following:

Step 5.1: For each unassigned node *u*, compute the insertion cost and its corresponding place along the route of vehicle *k**. Note that for unassigned nodes, their corresponding cost and optimal place for insertion in all routes other than that taken by vehicle *k** are calculated already and not changed and hence need not be recomputed.

Step 5.2: Compare and select the optimal place for inserting unassigned node *u* with the minimal increased costs among all possible routes. If unassigned node *u* cannot be included for service subject to the current vehicle capacity and time window constraints, exclude node *u* from the service list. When set $N_u(\tau)$ is empty, enter the unified framework of solution procedure. Otherwise, go to Step 4.

4.4 Method for Real-Time Route Improvement with Tabu Search

Our tabu search heuristic is inspired by a similar work for VRPTW (Badeau *et al.*, 1997; Cordeau *et al.*, 2001; Potvin *et al.*, 1996). We also use a tabu list of the fixed length. However, solution feasibility is always maintained for the real-time VPRTW (Duhamel *et al.*, 1997). Due to the dispatching requirements, the heuristic can be interrupted at checkpoints. In addition, we always keep the best solution found so far. The initial set of routes that are obtained by real-time route construction and improvement procedures must be remembered until a better solution is found by the tabu search.

The tabu search heuristic can be summarized as follows:

Steps for Real-Time Route Improvement with Tabu Search

Step 1: Initialization

Set the current and best solution to the initial route scheduling that are obtained by real-time route construction and improvement procedures.

Step 2: Tabu search step

Repeat Step 2 until maximum number of iterations is performed, or until maximum number of consecutive iterations is performed without any improvement to the best known solution.

Step 2.1: Tabu search with Or-opt

Repeat Step 2.1 until maximum number of consecutive iterations is performed without any improvement to the best known solution.

Step 2.1.1: Generate the neighborhood of the current solution by applying Or-opt exchanges. Note that constraints must be satisfied.

Step 2.1.2: Select the best non-tabu solution in this neighborhood and define this solution to be the new current solution.

Step 2.1.3: Update its tabu list. The inverse move is declared tabu for T iterations.

Step 2.2: If the earliest departure time of the critical nodes has arrived, go to Step 3. Otherwise, continue.

Step 2.3: Tabu search with 3-opt and 2-opt

Repeat Step 2.3 until maximum number of consecutive iterations is performed without any improvement to the best known solution.

Step 2.3.1: Generate the neighborhood of the current solution by applying 3-opt and 2-opt exchanges. Check these exchanges must satisfy all the constraints.

Step 2.3.2: Select the best non-tabu solution in this neighborhood and define this solution to be the new current solution.

Step 2.3.3: Update its tabu list. The inverse move is declared tabu for T iterations.

Step 2.4: If the earliest departure time of the critical nodes has arrived, go to Step 3. Otherwise, continue.

Step 2.5: Tabu search with Swap-opt.

Repeat Step 2.5 until maximum number of consecutive iterations is performed without any improvement to the best known solution.

Step 2.5.1: Generate the neighborhood of the current solution by applying Swap-opt exchanges. Check these exchanges must satisfy all the constraints.

Step 2.5.2: Select the best non-tabu solution in this neighborhood and define this solution to be the new current solution.

Step 2.5.3: Update its tabu list. The inverse move is declared tabu for T iterations.

Step 3: Return the best overall solution of the tabu search

If the new current solution of routes is better than the initial solution of routes, set the best overall solution to the new current solution of routes. Otherwise, the best overall solution is still the initial solution of routes that are obtained through real-time route construction and improvement procedures.

5. NUMERICAL EXAMPLES

5.1 Problem Set

The heuristic was tested on the Edclidean problems of Gélinas' *et al.* (1995). Gélinas' problems were obtained from Solomon's 100-customer Edclidean VRPTWs known as problem R101 to R105. To simplify the demonstration, we suppose each customer has either delivery demands or pickup demands and randomly select 10%, 30% and 50% of the 100 customers as pickup customers. However, our formulation has taken the delivery/pickup demands of the same customer into account.

In test problems, the travel times are equal to the corresponding Euclidean distances. A fixed amount of 10 time units is needed to unload or load the goods at each customer location. The width of the time window at the depot is set at 230 time units in all problems.

In addition, a discrete-time simulator was developed to test our anytime algorithm. The simulator uses relevant data to produce new service requests by the following formula:

$$\max(0, e_i - \theta c_{0i} - \xi) \tag{35}$$

where c_{0i} denotes the distance between depot 0 and node i, parameter ξ is a random number smaller than $e_i - \theta c_{0i}$, and adjustment parameter θ is designed to avoid generating new demands in the neighborhood of time window at node i. Here we set $\theta = 1$. The weights for travel time and waiting time are assumed as $\alpha = 0.7$, $\beta = 0.3$, and the estimated computation time for the initial solution of the RT-VRPTWDP is set as $\Delta = 5$ time units.

5.2 Parameter Setting of the Tabu Search

Based on Taguchi orthogonal arrays approach, parameter setting for tabu search was analyzed through experimentation on the real-time VRPTW problems. The best and least sensitive parameter setting of tabu search is selected by means of signal-to-noise ratio (*s/n)*. The size of the Taguchi orthogonal arrays is determined by the number of important parameters to be considered (Bounou, *et al.*, 1995). Based on our judgment, we define four control factors: length of tabu list, consecutive number of failed iterations, number of iterations and initial loads. Each factor is divided into three levels. Thus, a Taguchi orthogonal array corresponds to four factors and three levels. The combinations of the parameter setting for tabu search are reported in Table 1. Due to the stochastic features of RT-VRPTWDP, 10 independent runs were performed on each problem.

In addition, the average length of tabu list is 4.2. This value is slightly smaller than the one for solving static VRP. The average of consecutive number of failed iterations and total number of iterations is 10.3 and 51.3, respectively. Roughly speaking, they obey the empirical rules of parameter setting. The total number of iterations is more than ten times the length of tabu list. The consecutive number of failed iterations is about one fifth of total number of iterations. Finally, the average of initial loads is 143.3. The ratio of initial loads to vehicle capacity is about 0.72.

5.3 Testing Results

The heuristic proposed in Section 3 was implemented with the C programming language. The tests were run on Pentium IV personal commuter. Due to the stochastic features of RT-VRPTWDP, 30 independent runs were performed on each problem.

With the above input data, two modifications of our algorithm were further hypothesized for testing: (1) Partial algorithm 1: only real-time route construction procedure is performed; (2) Partial algorithm 2: real-time route construction procedure and real-time route improvement procedure without tabu search are performed.

The differences of solutions obtained by different algorithms in terms of total route costs and number of vehicles are summarized in left sides of Table 4 and Table 5, respectively. In right sides of Table 2 and Table 3, the percentages of improvement of solution obtained by the algorithms with route improvement procedure over the solution obtained by the algorithm without route improvement procedure are shown for each problem. In addition, the last columns of Table 2 and Table 3 are the differences of solutions obtained by complete algorithm and partial algorithm 2.

As a result, complete algorithm and partial algorithm 2 are superiorly than partial algorithm 1 in terms of total route costs. The total route costs has been lowered up to 12.94% and 16.63% in average for partial algorithm 2 and complete algorithm, respectively. We also observe that complete algorithm is superiorly than partial algorithm 2 in terms of total route costs. The total route costs has been lowered up to 3.69% in average.

The percentage of improvement with respect to the number of routes is rather erratic. The number of routes has been lowered up to 8.80% and 7.73% in average for partial algorithm 2 and complete algorithm, respectively. But the algorithm with tabu search isn't certainly superiorly than the algorithm without tabu search in terms of the number of vehicles. Perhaps it is because the number of vehicles is not taken into account the objective function of route improvement. When minimizing the total route costs, the exchange that leads to the largest decrease in route costs is preferred over all other exchanges, independently of the number of routes.

6. CONCLUDING REMARKS

In this paper, the RT-VRPTWDP is studied and formulated as a mixed integer programming model. The decision variables include not only link flows x_{ijk} but also departure time at node i, d_i, and at the depot 0 by vehicle k, d_{0k}. The problem is more difficult to handle, due to the simultaneous capacity constraints of the pickup and the delivery. Furthermore, the dispatcher wishes to know the solution to the current problem as soon as possible (preferably within minutes or seconds). The time limit of dispatching implies that rerouting is often done by adopting local improvement heuristics. It is therefore essential that updating information mechanism are integrated into the solution method. The route improvement procedure follows the general guidelines of anytime algorithm in our study. A unified framework comprising of initial route construction, real-time route construction, real-time route improvement and tabu search was proposed and validated for the RT-VRPTWDP.

15 problems made up by Gélinas were taken with minor modifications. Based on Taguchi orthogonal arrays approach, parameter setting for tabu search was analyzed through experimentation on the real-time VRPTW problems. The average of length of tabu list was slightly smaller than the one for solving static VRP. The algorithms with route improvement procedure are superiorly than the algorithm without route improvement procedure. We also observe that the elaboration of tabu search in the anytime algorithm can further reduce the total route costs.

The RT-VRPTWDP can be improved or extended in a number of ways. First, the computation of the new waiting time w' is the key. Computation of all the waiting times in the route is not practical. Parallel algorithms should be useful for the fast computation. Second, the occurrence of new requests is the only source of uncertainty in this study. The stochastic properties of demands, customers, service time and travel time would be considered in an extended model. These research topics are currently undergoing by the authors.

Table 1. Combinations of Parameter Settings for the Tabu Search

Percentage of Pickup Customers	Level	Control Factor			
		Length of Tabu Lists	Consecutive Number of Failed Iterations	Total Number of Iterations	Initial Loads
10%	1	3	6	30	110
	2	5	10	50	150
	3	7	14	70	190
30%	1	3	6	30	90
	2	5	10	50	130
	3	7	14	70	170
50%	1	3	6	30	70
	2	5	10	50	110
	3	7	14	70	150

Table 2. Differences of Total Route Costs

Problem	Percentage of Pickup Customers	Total Route Costs			Percentage of Improvement		
		A	B	C	D	E	F
		Partial Algorithm 1	Partial Algorithm 2	Complete Algorithm	(A – B)/A	(A – C)/A	E – D
R101	10%	10094.30	8359.09	7910.14	17.19%	26.13%	8.94%
	30%	10176.30	8311.31	7922.17	18.33%	22.15%	3.82%
	50%	10125.90	8402.60	7983.28	17.02%	21.16%	4.14%
R102	10%	9607.05	8183.44	8086.84	14.82%	18.58%	3.76%
	30%	9876.86	8336.47	8072.95	15.60%	18.26%	2.67%
	50%	9554.49	8173.78	7920.86	14.45%	17.10%	2.65%
R103	10%	8700.01	8247.41	7780.14	5.20%	11.15%	5.95%
	30%	8979.16	8012.10	7879.52	10.77%	12.25%	1.48%
	50%	8712.76	8046.18	7724.04	7.65%	11.35%	3.70%
R104	10%	7648.98	7198.13	6972.52	5.89%	9.40%	3.50%
	30%	7658.91	7087.62	6804.53	7.46%	11.16%	3.70%
	50%	7595.26	7214.08	6751.54	5.02%	11.11%	6.09%
R105	10%	8860.25	7308.20	7250.88	17.52%	22.02%	4.50%
	30%	8860.25	7229.71	7227.45	18.40%	18.43%	0.03%
	50%	8886.92	7211.36	7175.61	18.85%	19.26%	0.40%

Table 3. Differences of the Number of Routes

Problem	Percentage of Pickup Customers	Number of Routes			Percentage of Improvement		
		A	B	C	D	E	F
		Partial Algorithm 1	Partial Algorithm 2	Complete Algorithm	(A – B)/A	(A – C)/A	E – D
R101	10%	26.10	23.80	23.50	8.81%	9.96%	1.15%
	30%	26.20	23.60	23.50	9.92%	10.31%	0.38%
	50%	26.20	23.60	23.60	9.92%	9.92%	0.00%
R102	10%	23.70	21.60	21.40	8.86%	9.70%	0.84%
	30%	24.00	22.15	21.85	7.71%	8.96%	1.25%
	50%	24.00	21.50	21.85	10.42%	8.96%	-1.46%
R103	10%	22.00	20.30	20.65	7.73%	6.14%	-1.59%
	30%	22.00	20.85	20.95	5.23%	4.77%	-0.45%
	50%	22.00	20.15	20.85	8.41%	5.23%	-3.18%
R104	10%	19.30	17.70	18.00	8.29%	6.74%	-1.55%
	30%	19.00	16.50	17.35	13.16%	8.68%	-4.47%
	50%	18.70	17.90	18.00	4.28%	3.74%	-0.53%
R105	10%	19.00	17.30	17.70	8.95%	6.84%	-2.11%
	30%	19.00	17.15	17.55	9.74%	7.63%	-2.11%
	50%	19.00	17.00	17.40	10.53%	8.42%	-2.11%

REFERENCES

1. Badeau, P., Guertin, F., Gendreau, M., Potvin, J.Y., Taillard, E. (1997) A parallel tabu search heuristic for the vehicle routing problem with time windows, **Transportation Research, Vol. 5(C), No. 2,** 109-122.
2. Bertsimas, D., Ryzin, G.V. (1991) A stochastic and dynamic vehicle routing problem in the Euclidean plane, **Operations Research, Vol. 39, No. 4,** 601-615.
3. Bertsimas, D., Ryzin, G.V. (1993) Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles, **Operations Research, Vol. 41, No. 1,** 60-76.
4. Bertsimas, D.J., Simchi-Levi, D. (1996) A new generation of vehicle routing research: robust algorithms, addressing uncertainty, **Operations Research, Vol. 44, No. 2,** 286-303.
5. Bounou, M., Lefebvre, S., Do, X.D. (1995) Improving the quality of an optimal power flow solution by Taguchi Method, **Electrical Power and Energy Systems, Vol. 17, No. 2**, 113-118.
6. Chen, H.K., Hsueh, C.F., Chang, M.S. (2002) An effective heuristic for real-time time-dependent vehicle routing problem with time-windows, **Proceeding of International Conference on Inter-City Transportation**, Beijing, China, 370-379.
7. Cordeau, J.F., Laporte, G., Mercier, A. (2001) A unified tabu search heuristic for vehicle routing problems with time windows, **Journal of the Operational Research Society, Vol. 52,** 928-936.

8. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F. (1995) **Handbooks in Operations Research and Management Science, Vol. 8**, 35-140, Elsevier Science, Amsterdam.
9. Duhamel, C., Potvim, J., Rousseau, J. (1997) A tabu search heuristic for the vehicle routing problem with backhauls and time windows, **Transportation Science, Vol. 31, No. 1,** 49-59.
10. Fisher, M.L. (1995) **Network Routing, Vol. 8**, 1-33, Elsevier Science, Amsterdam.
11. Gélinas, F., Desrochers, M.J., Descrosiers, J., Solomon, M.M. (1995) A new Branching Strategy for time constrained routing problems with application to backhauling, **Annuals of Operations Research, Vol. 61,** 91-110.
12. Gendreau, M., Potvin, J.Y. (1998) **Fleet Management and Logistics**, 115-126, Kluwer Academic Publishers.
13. Gendreau, M., Guertin, F., Potvin, J.Y., Taillard, E. (1999) Parallel tabu search for real-time vehicle routing and dispatching, **Transportation Science, Vol. 33, No. 4,** 381-390.
14. Glover, F. (1986) Future paths for integer programming and links to artificial intelligence, **Computers and Operations Research, Vol. 5,** 533-549.
15. Hung, C.L. (2001) **Vehicle Routing Problem with Real Time Demands and Time Widow Constraint**, Master Thesis, Department of Business, Chung-Hwa University, Taiwan. (In Chinese)
16. Larson, A. (2001) **The Dynamic Vehicle Routing Problem**, Ph.D. Thesis, IMM.
17. Liu, C.W. (2000) **Study on Time Dependent Vehicle Route Problem with Time Window**, Ph.D. Thesis, Department of Civil Engineering, National Central University, Taiwan. (In Chinese)
18. Psaraftis, H.N. (1995) Dynamic vehicle routing: status and prospects, **Annuals of Operations Research, Vol. 61,** 143-164.
19. Potvin, J.Y., Kervahut, T., Garcia B.L., Rousseau, J.M. (1996) The vehicle routing problem with time windows part I: Tabu search, **INFORMS Journal on Computing, Vol. 8, No. 2,** 158-164.
20. Powell, W.B., Jaillet, P., Odoni, A. (1995) **Network Routing, Vol. 8**, 141-295, Elsevier Science, Amsterdam.
21. Shieh, H.M., May, M.D. (1998) On-line vehicle routing with time windows: Optimization-based heuristics approach for freight demands requested in real-time, **Transportation Research Record, No. 1617,** 171-178.
22. Solomon, M.M. (1987) Algorithms for the vehicle routing and scheduling problems with time windows constraints, **Operations Research, Vol. 35, No. 2,** 254-265.
23. Zilberstein, S., Russell, S.J. (1996) Optimal composition of real-time systems, **Artificial Intelligence, Vol. 82, No. 1-2,** 181-213.