AN MILP FRAMEWORK FOR DYNAMIC VEHICLE ROUTING PROBLEMS WITH TIME WINDOWS

R. DONDO and J. CERDÁ

INTEC (Universidad Nacional del Litoral - CONICET)
Address: Güemes 3450 – (3000) Santa Fe – Argentina. Tel. +54(342)4559174/77 Fax: +54(342)4550944
E-mail: rdondo@ceride.gov.ar

***Abstract*— A key issue in logistics is the efficient management of a vehicle fleet servicing a set of customers with known demands. Every vehicle route must start and finish at the assigned depot, each customer is to be visited by a single vehicle and vehicle capacities must not be exceeded. These are the constraints for the capacitated vehicle routing problem (VRP) whose objective is usually the minimization of the travel distance. When every customer has an associated time window, we are dealing with the vehicle routing problem with time windows (VRPTW), an NP-hard problem extensively studied. In the static VPRTW, all the problem data are given. A more challenging subject is the dynamic VRPTW (DRVPTW) where routes must be periodically updated because of new service requests. In DVRPTW, the information on the problem is time-dependent since the data are in part given a priori and in part dynamically updated. As a result, the best solution must be periodically revised. There are two classes of DVRPTW solution methodologies: the immediate assignment that updates vehicle routes as soon as a new service request is received, and the deferred assignment retaining the new service calls for a certain time period before dispatching them all at once. The latter type has been adopted in this paper. At the time of revising their routes, the vehicles are already on duty and some nodes have already been visited. The remaining old customers that have designated vehicles are either being serviced or awaiting service. The customers to be considered in the DVRPTW include not only old customers still to be serviced but also new visit requests. The DVRPTW is tackled by solving a series of static VRPTW problems, with each one being defined every time the input data is updated. The approach assumes that each vehicle will start its new route at the location where it is servicing or to which it is traveling.**

***Keywords*— Dynamic routing problems, MILP reactive strategy, Deferred assignment.**

I. INTRODUCTION

The dynamic vehicle routing problem with time windows (DVRPTW) represents an interesting research issue since it presents some distinctive features with regards to the static VRPTW. In addition to the routing issue, another major topic is the dynamic scenario within which decisions are to be taken. Repeated changes in vehicle routing and scheduling have to be made at different times over a rolling time-horizon that should account for new service calls but also for earlier routing decisions. Real-world experience indicates that dynamic routing problems must be studied because:

i) The economic benefits of an efficient logistical system are very significant
ii) Distribution scenarios where the information is dynamically updated are more frequent.
iii) Real-time data processing is becoming a feasible option due to the dramatic advances in computation and communications technology.

In dynamic routing problems, when re-routing is executed, the vehicle fleet is already on duty and some nodes already serviced are no longer considered. The remaining "old" customers that have designated vehicles are either being serviced or waiting for the service. Therefore, the set of customers in the DVRPTW problem should include old nodes still to be serviced and new pickup requests. Since quick execution time is a pre-requisite for on-line solution of the DVRPTW, a good trade-off between the solution quality and the required computer time must be achieved. Most proposed solution algorithms for the DRPTW are heuristics/methaheuristics but little research has been focused on model-based reactive formulations. This work introduces a reactive solution strategy for the DVRPTW that is based on a novel mixed integer linear problem (MILP) formulation and accounts for heterogeneous fleets. Using both angular and Euclidean metrics to identify neighboring routes for a given node, a small set of candidate tours for the (re)insertion of old/new customers can be defined and embedded in the formulation. Each time the proposed MILP model is solved, multiple vehicle-to-node (re)assignment and reordering of nodes are simultaneously performed. The proposed mathematical model has been derived by reformulating the reactive approach of Dondo and Cerdá (2005). The new methodology applies an "*insertion & local search*" strategy each time the vehicle routes & schedules are updated. By using such a two-step search strategy, the new customers are first assigned to vehicles while allowing a partial reassignment of old nodes (the insertion step), and subsequently the nodes on a given route visited by the same vehicle are optimally reordered (the local search step). The approach can be regarded as a deferred assignment methodology that retains the new service calls for a certain period of time before dispatching them all at once. The method was applied to a DVRPTW example that involves 50 nodes and 8 vehicles yielding satisfactory results at low CPU time.

II. PROBLEM STATEMENT

Consider a route-network represented by an undirected graph G{I, P, N} with $I = \{i_1, i_2, ..., i_n\}$ denoting the set of nodes or customers, and $P = \{p_1, p_2, ..., p_l\}$ representing the set of depots. Nodes and depots are connected by a set of route segments $N = \{(i,j) / i,j \in I \cup P\}$. For each customer $i \in I$, there is a known load l_i to be picked-up (delivered) within a time window $[a_i, b_i]$, where a_i is the earliest service time and b_i is the latest service time. There is set of vehicles $V = \{v_1, v_2, ..., v_m\}$ that must transport the load from (to) nodes $i \in I$ to (from) depots $p \in P$. In addition, vehicle-dependent unit costs $C = \{c_{ij}^v\}$, vehicle-dependent travel times $\Gamma = \{t_{ij}^v\}$ and travel distances $D = \{d_{ij}\}$ are given data for any route segment $(i,j) \in N$. The service time on node i is denoted st_i^v. To apply the proposed approach, the set I is split into a pair of non-intersecting subsets $I = \{ I^{old} \cup I^{new}\}$, where I^{old} denotes the set of old nodes already scheduled o serviced and I^{new} stands for the new customers to be inserted on the vehicle routes. In turn, the set I^{old} comprises the nodes earlier serviced ($I_1^{\ old}$) and those still to be visited or being currently serviced ($I_2^{\ old}$) at the update time; i.e. $I^{old} = \{I_1^{\ old} \cup I_2^{\ old}\}$. Problem DVRPTW should only account for the nodes included in $I_2^{\ old}$ and I^{new}. The sets I^{old} , I^{new} , $I_1^{\ old}$ and $I_2^{\ old}$ all change with the update time t_n. Customers already visited are transferred from $I_2^{\ old}$ to $I_1^{\ old}$. In the same way, the vehicle set V can be decomposed into the pair of non-intersecting sets $\{V^{old} \cup V^{new}\}$, where V^{old} stands for the vehicle fleet on duty and V^{new} represent stand-by vehicles that have not yet been used. Vehicles that have already completed their tours are ignored by deleting them from the set V^{old}. For the multi-depot case, only vehicles belonging to V^{old} have designated depots while the ones for stand-by vehicles will be chosen by the DVRPTW. The problem goal is to find the updated vehicle routes & schedules that minimize a combination of vehicle fixed costs, travel time, distance-based and inconvenience costs. The solution must satisfy the following constraints: (i) each vehicle route must end at the same depot from which it departures; (ii) though the vehicle assignment may change when updating the routes, each node must be visited by a single vehicle; (iii) the total amount of load assigned to vehicle v along the tour must never exceed its capacity q_v ; (iv) the duration of the trip for any vehicle v should be shorter than a maximum allowed routing time tv_v^{max}; (v) the service at node i should start within the time window $[a_i, b_i]$. The proposed approach assumes that each vehicle will start its updated route at the node to (at) which is traveling (servicing) at the update time. Vehicle assignments for such old nodes $i \in (I_2^{\ old})_F \subset I_2^{\ old}$ that become the starting points of the updated routes are consequently frozen. If vehicle v is the designated vehicle for node $i \in (I_2^{\ old})_F$ at the update time t_n, must still be visiting i at time t_{n+1}. Therefore, the set $(I_2^{\ old})_F$ will contain as many elements as the cardinality of the set V^{old}, i.e. a single frozen node for every vehicle already on duty. Each element of $(I_2^{\ old})_F$ will be represented by i_v^{start} with $v \in V^{old}$ being the vehicle allocated to node i at the previous routing revision. On the other hand, non-frozen old nodes $i \in (I_2^{\ old})_{NF}$ and new nodes $i \in I^{new}$ can be (re)assigned but to a small set of neighboring routes in order to reduce the DVRPTW problem size. In other words, any node $i \in \{(I_2^{\ old})_{NF} \cup I^{new}\}$ can be (re)allocated to a vehicle $v \in V_i \subset V$, where V_i is the set of vehicles traveling along neighboring routes around node i. To define the notion of neighboring route, two different metrics are used. A metric based on angles is defined to measure the "distance" between a node i and a given route traveled by vehicle v. In addition, an Euclidean metric is defined to measure the distance between a node i and the starting location of vehicle v. If both the angle and the Euclidean distances are less than some specified values, the vth-route is regarded as a neighboring trip and the node i can be (re)assigned to vehicle v during the rerouting process. As already mentioned, the vehicle $v \in V^{old}$ starts its updated route provided by DVRPTW at node i_v^{start}. Therefore, i_v^{start} will precede any node $j \in \{(I_2^{\ old})_{NF} \cup I^{new}\}$ located on the same route and visited by vehicle $v \in V_j$. If PR_j stands for the set of nodes that should precede node j, then node i_v^{start} will belong to PR_j. Selected customer time windows often prescribe beforehand some precedence relationship between some pairs of nodes if a common vehicle visits them. Before solving the dynamic VRPTW problem, some TW-based rules are applied to identify such precedence relationships between nodes and subsequently include them into the set PR_i for each node $i \in I$.

III. THE DVRPTW FORMULATION

The proposed MILP formulation requires to define the following binary variables: (i) The assignment variable Y_{iv} to allocate vehicle $v \in V$ to customer $i \in \{(I_2^{\ old})_{NF} \cup I^{new}\}$. (ii) The assignment variable X_{vp} to allocate vehicles $v \in V^{new}$ to depots $p \in P_v$. (iii) The sequencing variable S_{ij} to denote that the customer site $i \in \{I_2^{\ old} \cup I^{new}\}$ is visited before (S_{ij} =1) or after the node j (S_{ij} = 0). A single S_{ij} is required to define the precedence relationship for a pair of nodes (i,j). In the case node i belongs to $PR_j \subseteq \{I_2^{\ old} \cup I^{new}\}$ and both nodes (i,j) are serviced by the same vehicle ($Y_{iv} = Y_{jv} = 1$), then S_{ij} =1 and, consequently, such a sequencing variable can be deleted from the problem formulation.

Objective function: The goal is to find an updated set of routes & schedules for the vehicle fleet that accounts for the new service requests and the current state of the distribution in progress scenario and minimizes an appropriate combination of vehicles fixed costs ($\Sigma_{v\in V}\Sigma_{p\in P}\ cf_v$), distance costs ($\Sigma_{v\in V}\ CV_v$), travel time costs ($\Sigma_{v\in V}\ \rho_t TV_v$) and inconvenience costs ($\Sigma_{i\in I}\ \rho_i\ (\Delta a_i+\Delta b_i) + \Sigma_{v\in V}\ \rho_v\ \Delta v_v$). Costs due to time windows violations ($\Sigma_{i\in I}\ \rho_i\ (\Delta a_i+\Delta b_i)$) are also known as "customer dissatisfaction".

$$Min\sum_{v\in V}\left(cf_v\sum_{p\in P}X_{pv}+CV_v+\rho_t TV_v+\rho_v\Delta T_v\right)+\sum_{i\in I}\rho_i(\Delta a_i+\Delta b_i) \qquad \textbf{(1)}$$

Constraints:

Assignment of vehicles to customers: Every node $i \in \{(I_2^{old})_{NF} \cup I^{new}\}$ must be assigned to a single vehicle $v\in V_i$. The summation is extended over the subset of vehicles V_i that can be allocated to node i.

$$\sum_{v\in V_i} Y_{iv}=1 \qquad \forall i\in\{(I_2^{old})_{NF}\cup I^{new}\} \qquad \textbf{(2)}$$

Forbidden vehicle assignments. If the pair of nodes $(i,j)\in\{I_2^{old}\cup I^{new}: i<j\}$ satisfies the following condition $\left[\left(a_i+st_i^v+t_{ij}^v\right)>b_j \wedge \left(a_j+st_j^v+t_{ij}^v\right)>b_i\right]$, then they are said to be incompatible nodes in the sense that they cannot be serviced by the same vehicle. Otherwise, at least one of the time window constraints would be violated. If the pair (i,j) are incompatible nodes, then the constraint (3.a) is to be incorporated in the problem formulation. If a particular node $j\in\{(I_2^{old})_{NF}\cup I^{new}\}$ is incompatible with the frozen old node i_v^{start}, then Eq. (3.a) reduces to constraint (3.b), Y_{jv} is made equal to 0 and deleted from the problem formulation.

$$\{Y_{iv}+Y_{jv}\le 1\} \qquad \textbf{(3.a)}$$

$\forall i,j\in(I_2^{old})_{NF}\cup I^{new}$, (i,j), $v\in V_i\cap V_j$ are incompatible

$$\{Y_{jv}=0\} \qquad \textbf{(3.b)}$$

$\forall i\in I_F^{old},\ j\in(I_2^{old})_{NF}\cup I^{new},\ v\in V_i$ j is i^{th} incompatible

Assignment of vehicles to depots: A single depot $p\in P_v$ should be assigned, if used, to every vehicle $v\in V^{new}$ where P_v is the set of feasible depots for v. In such a case, the summation on Eq. (4) will be equal one; otherwise it should be zero.

$$\sum_{p\in P_v} X_{pv}\le 1 \qquad \forall v\in V^{new} \qquad \textbf{(4)}$$

Vehicle capacity constraints: Equations (5) state that the overall cargo transported by vehicle v to the assigned depot, including those ones picked-up at nodes $i\in(I_1^{old})_v$ must never exceed its capacity q_v. $(I_1^{old})_v$ stands for old nodes already serviced by vehicle v while $(I_2^{old})_v$ comprises the old nodes that can be visited by vehicle $v\in V_i$, including i_v^{start}. Eq. (5.a) is written for vehicles on duty ($v\in V^{old}$) with known depots while Eq. (5.b) applies to stand-by vehicles $v\in V^{new}$ that, if used, are housed at depots selected by the DVRPTW.

$$\sum_{i\in(I_1^{old})_v} l_i+\sum_{i'\in(I_2^{old})_v\cup I_v^{new}} l_{i'}Y_{i'v}\le q_v \qquad \forall v\in V^{old} \qquad \textbf{(5.a)}$$

$$\sum_{i\in(I_2^{old})_v\cup I_v^{new}} l_i Y_{iv}\le q_v\left(\sum_{p\in P_v}X_{pv}\right) \qquad \forall v\in V^{new} \qquad \textbf{(5.b)}$$

Time windows and maximum service time constraints: Constraints (6) ensure that the updated schedule does not violate time windows and maximum service time hard constraints. If regarded as soft constraints, they can be violated at a penalty cost that is proportional to the violation size given by the terms Δa_i, Δb_i and ΔT_v, respectively. Timing constraints can be treated as hard constraints by driving Δa_i, Δb_i and Δt_v in Eqs. (6.a) and (6.b) to zero.

$$\left\{\begin{array}{l}\Delta a_i\ge a_i-T_i\\ \Delta b_i\ge T_i-b_i\end{array}\right\} \qquad \forall i\in I_2^{old}\cup I^{new} \qquad \textbf{(6.a)}$$

$$\Delta T_v\ge TV_v-tv_v^{\max},\ \forall v\in V \qquad \textbf{(6.b)}$$

Node sequencing constraints: If a pair of nodes $i,j\in\{I_2^{old}\cup I^{new}\}$ are on the same tour and node i is visited before, then the pickup service at node j can never start at a time T_j earlier than the vehicle departure time from node i increased by the travel time t_{ij}. The departure time from node i is found by simply adding the service time st_i to the time T_i at which the service begins. This conditional constraint given by Eqs. (7.a) should become active only if both requirements are satisfied: (i) both nodes have been assigned to the same vehicle ($Y_{iv}=Y_{jv}$, = 1, for some vehicle v) and (ii) node i is visited earlier ($S_{ij}=1$). If both are on the same tour but node j is first serviced ($S_{ij}=0$), then Eq. (7.b) will be binding and Eqs. (7.a) will become redundant. Otherwise, $Y_{iv}+Y_{jv}<2$ and constraints (7.a) and (7.b) both become redundant. Similar relationships between the distance-based traveling costs for any pair nodes $(i,j)\in\{I_2^{old}\cup I^{new}\}$ can also be written .

$\forall i,j\in I_2^{old}\cup I^{new},\ v\in V_i\cap V_j \quad : i<j$

$$\left\{\begin{array}{l}C_j\ge C_i+c_{ij}^v-M_C(1-S_{ij})-M_C(2-Y_{iv}-Y_{jv})\\ T_j\ge T_i+st_i+t_{ij}^v-M_T(1-S_{ij})-M_T(2-Y_{iv}-Y_{jv})\end{array}\right. \qquad \textbf{(7.a)}$$

$$\left.\begin{array}{l}C_i\ge C_j+c_{ij}^v-M_C S_{ij}-M_C(2-Y_{iv}-Y_{jv})\\ T_i\ge T_j+st_j+t_{ij}^v-M_T S_{ij}-M_T(2-Y_{iv}-Y_{jv})\end{array}\right\} \qquad \textbf{(7.b)}$$

If $i\in PR_j$, then every node j located on the same route will be preceded by node i. In such a case, Eqs. (7.a) and (7.b) reduce to constraints (7.c) that become active only if nodes i and j are visited by the same vehicle v ($Y_{iv}=Y_{jv}=1$).

$\forall i,j\in(I_2^{old})\cup I^{new}$ $i\in PR_j,\ v\in V_i\cap V_j$

$$\left\{\begin{array}{l}C_j\ge C_i+c_{ij}^v-M_C(2-Y_{iv}-Y_{jv})\\ T_j\ge T_i+st_i+t_{ij}^v-M_T(2-Y_{iv}-Y_{jv})\end{array}\right\} \qquad \textbf{(7.c)}$$

On the other hand, $i=i_v^{start}$ will be the starting point of the new route for vehicle v and, therefore, it will precede every node j located on the same route. In such a case, $Y_{iv}=1$ and Eq. (7.c) reduces to constraint (7.d).

$i=i_v^{start},\ j\in(I_2^{old})_{NF}\cup I^{new},\ v\in V^{old}$

$$\left\{\begin{array}{l}C_j\ge C_i+c_{ij}^v-M_C(1-Y_{jv})\\ T_j\ge T_i+st_i+t_{ij}^v-M_T(1-Y_{jv})\end{array}\right\} \qquad \textbf{(7.d)}$$

Furthermore, the following condition must be satisfied for the initial node of a tour traveled by vehicle $v\in V^{new}$,

$$\forall i \in (I_2^{old})_{NF} \cup I^{new}, v \in V_i^{new}, \; p \in P_v$$
$$\begin{Bmatrix} C_i \geq c_{ip}{}^v - M_C\,(2 - X_{pv} - Y_{iv}) \\ T_i \geq t_{ip}{}^v - M_T\,(2 - X_{pv} - Y_{iv}) \end{Bmatrix} \qquad \textbf{(7.e)}$$

Since the initial node of the *v*th-tour is not known beforehand, the condition (7.e) is applied to any node $i\in\{I_2^{old} \cup I^{new}\}$ that can be (re)assigned to vehicle $v \in V_i^{new}$.

End tour conditions: Eqs. (8) state that both the total distance-based traveling cost (CV_v) and the total travel time (TV_v) associated to the *v*th-tour can be obtained from the travel cost/time (C_i/T_i) to spend up to the last node serviced by *v*, by simply adding to it the cost ($c_{ip}{}^v$) /time ($st_i+t_{ip}{}^v$) required to return to the starting depot. Since the last node visited by vehicle *v* is not known beforehand, the constraints (8) are written for any node $i\in\{I_2^{old} \cup I^{new}\}$. Constraints (8.a) apply to old vehicles $v\in V^{old}$ while constraints (8.b) are written for new vehicles.

$$i \in (I_2^{old})_{NF} \cup I^{new}, v \in V_i^{old}$$
$$\begin{Bmatrix} CV_v \geq C_i + c_{ip}{}^v - M_C(1-Y_{iv}) \\ TV_v \geq T_i + st_i + t_{ip}{}^v - M_T(1-Y_{iv}) \end{Bmatrix} \qquad \textbf{(8.a)}$$

$$\forall i \in (I_2^{old})_{NF} \cup I^{new}, v \in V_i^{new}, \; p \in P_v$$
$$\begin{Bmatrix} CV_v \geq C_i + c_{ip}{}^v - M_C\,(2 - X_{pv} - Y_{iv}) \\ TV_v \geq T_i + st_i + t_{ip}{}^v - M_T\,(2 - X_{pv} - Y_{iv}) \end{Bmatrix} \qquad \textbf{(8.b)}$$

IV. AN INSERTION & *LOCAL SEARCH* STRATEGY

The mathematical model introduced in the previous section has been embedded into an *insertion & local search* procedure that is summarized in Figure 1. The procedure parameters are the following: (1) φ_1: maximum angular distance from node *i* to the currently assigned *v*th-route axis θ_v, below which node *i* must still be serviced by the same vehicle *v* in the next iteration. If node *i* is farther than φ_1, then it could be transferred to another tour. (2) φ_0: maximum angular distance with respect to another *v*th-route axis θ_v below which node *i* can be serviced by vehicle $v\in V_i$ on the next iteration. (3) d_1^{max}: maximum Euclidean distance from node *i* to the Cartesian location of the currently assigned vehicle $v\in V_i$, below which node *i* will be serviced by the same vehicle $v\in V_i$ on the next iteration. If node *i* is farther than d_1^{max}, then it could be transferred to a neighboring tour. (4) d_0^{max}: maximum Euclidean distance from node *i* to the Cartesian location of another vehicle *v*, below which node *i* can be serviced by vehicle *v* on the next iteration. These parameters govern the shapes and sizes of the operational zones for old vehicles $v\in V^{old}$ and determine the levels of zone overlapping. In the initial step of the procedure, parameters φ_1, φ_0, d_1^{max}, d_0^{max} are all tuned. In addition, the time step Δt is adopted, the iteration number *n* is set equal 0 and the update time t_o is made equal to 0. Initially, $I^{old} = I_1^{old}\cup I_2^{old}$ is an empty set and I^{new} includes all service requests available at t= 0. Moreover, the sets $(I_2^{old})_F$ & $(I_2^{old})_{NF}$ are also empty sets and the available vehicles all belong to the set V^{new}. In step 2, the resulting DVRPTW mathematical formulation is solved.

Step 1. Initialize model parameters.

1a. Settings: Set the values for the procedure parameters $\varphi 1$, $\varphi 0$, d_1^{max}, d_0^{max} and the update time step Δt.

1b. Iteration number: n = 0; re-routing time: t_0 = 0.

1c. Set $I^{old} = I_1^{old} = I_2^{old} = (I_2^{old})_F = (I_2^{old})_{NF} = \varnothing$; I^{new} ={all service calls at t_0 = 0}; $V^{old} = \varnothing$; V^{new} ={all available vehicles at t_0 = 0}; $V_i = V$ for any $i\in I^{new}$.

Step 2. Solve the DVRPTW mathematical formulation to find the best initial vehicle routes.

Step 3. Update the model parameters.

3a. $n = n$+1, $t_{n+1} = t_n + \Delta t$.

3b. Update the node sets I_1^{old}, I_2^{old}, $(I_2^{old})_F$, $(I_2^{old})_{NF}$ and I^{new} as well as the vehicle sets V^{old} and V^{new}. If $n > n_{max}$ or $I_2^{old} = I^{new} = \varnothing$, END. Otherwise, go to step (3c).

3c. Update the preceding node sets PR_i, $i\in I_2^{old}$

3d. Determine the equivalent Cartesian coordinates for used vehicles (x_v, y_v)

3e. Compute equivalent vehicle polar coordinates (θ_v, r_v) from their Cartesian coordinates

3f. Determine the Euclidean distance from nodes to vehicles (d_{iv}) and the angular distances from nodes to routes (φ_{iv})

3h. Define the set of neighboring tours for every node $i\in I_2^{old}\cup I^{new}$ through the sets V_i, $i\in(I_2^{old})_{NF} \cup I^{new}$.

Step 4. Solve the DVRPTW model in two steps.

4a. Re(assignment) of vehicles to nodes. Solve the DVRPTW while keeping unchanged the ordering of old nodes if they remain on the same route through the sets PR_i.

4b. Reordering of nodes on every route. Solve the DVRPTW keeping unchanged the set of nodes visited by each used vehicle found in step 4a.

4c. Return to step 3.

Figure 1: The solution strategy

After finding the best routes for the used vehicles, values of the distances φ and *d* for every pair (*i*,*v*) are determined in order to find the set of neighboring tours for every non-frozen node at the next iteration (step 3). Moreover, scheduled & serviced nodes are transferred from I^{new} to I_2^{old} while used vehicles are moved from V^{new} to V^{old}. In addition, $n = n$+1, $t_{n+1} = t_n + \Delta t$, the old nodes already visited are transferred from I_2^{old} to I_1^{old} and new service calls are incorporated in I^{new}. Furthermore, unused and new available vehicles are included in V^{new} and the node i_v^{start} on every route is identified to define the set $(I_2^{old})_F = \{i_v^{start}, v \in V^{old}\}$ and, consequently, $(I_2^{old})_{NF} = I_2^{old} - (I_2^{old})_F$. Additionally, the precedence ordering of old nodes $i\in I_2^{old}$ on the current routes are stored in the sets PR_i. After that, the step 4 is executed and the DVRPTW model is solved again but this time in two steps to first find the nodes to be visited by each vehicle (the insertion step) and then the way they are sequenced on each route (the local search step). While making the insertion step, the ordering of old nodes located on the same route at the previous iteration, given by PR_i, are preserved if they are still visited by a common vehicle on the next iteration. By doing that, a great deal of sequencing variables can be deleted from the problem formulation. Similarly, assignment variables Y_{iv} and constraints (2)-(5) can be omitted during the local search step. After updating the routes, step 3 is repeated again and the procedure ends when either the last routing update has been executed ($n > n_{max}$) or all customers have been serviced ($I_2^{old} = I^{new} = \emptyset$).

V. AN ILLUSTRATIVE EXAMPLE

The proposed dynamic VRPTW framework has been tested by tackling a variant of the fifty-node Solomon benchmark problem R-110 (Solomon, 1987). Changes were introduced in problem R-110 to mimic a dynamic scenario. The example involves a central depot hosting an homogeneous fleet of 8 vehicles (V1-V8) all featuring a capacity q_v = 200 and a maximum service time tv_v^{max} = 230. Six of them, namely V1-V6, are ready to perform pickup tasks at t = 0. The other two vehicles V7-V8 will be held in reserve waiting for additional service calls that can hardly be satisfied by the vehicles on duty without violating time constraints (See Table 1). At t = 0, just 25 service requests represented by the first 25 nodes of the benchmark problem R-110 have been received. The selected objective function includes traveled distances and inconvenience costs ($\rho_i = \rho_v = 1$) but no fixed costs (cf_v = 0). In addition, it was adopted $\rho_t = 1.10^{-4}$ to also minimize both the overall idle time and the vehicle schedule makespan. Time window-based rules identifying incompatible nodes have been applied before tackling the formulation in order to initialize the sets PR_i. The resulting VRPTW problem was solved to establish the optimal routes & schedules for the six vehicles V1-V6 on a 733 Mhz 256 MB RAM Pentium III PC using ILOG OPL Studio 3.7 with the CPLEX 9.0 solver. The optimal solution featuring just non-overlapping routes and no time window constraint violations has been found in 4.12 s and is shown in Table 2 and Fig 2. Nodes already serviced at the next update time are highlighted in this table. Vehicle tours are assumed to be periodically updated after some fixed period of time (Δt = 40) to schedule additional pickup services. While the six vehicles are servicing customers, eight new requests have been accepted within the time interval [t = 0, t = 40] (see Table 1). Since Δt = 40, the first update of vehicle routes and schedules is made at t = 40. At that time, services at nodes n2 and n5 have been completed while pickup tasks at nodes n12 and n21 by vehicles V3 and V6 are just being performed (see Table 2). Therefore, vehicles V3 and V6 will be ready to start the updated routes from such nodes after completing the ongoing services. In turn, vehicles V1 and V4 are moving along the route segments connecting the pairs of nodes (n2, n15) and (n5, n17), respectively. Therefore, vehicles V1 and V6 will still be allocated to nodes n15 and n17, respectively, from which they are going to start the updated routes. All other vehicle-node assignments can be redefined and the unused vehicles V7 and V8 are now feasible choices. As the scheduled vehicle V5 is still idle at t = 40, then every node previously assigned to V5, even the earliest visited n7, can be transferred to another route. In particular, n7 was reallocated to vehicle V2.

Table 1: Services requested while the vehicles are on duty

Service requests arriving within the time period [t = 0, t = 40]									
Node	n28	n31	n29	n47	n30	n40	n49	n44	
a_i	40	40	40	40	40	51	93	40	
b_i	79	85	98	116	123	128	132	138	
Load	16	27	9	27	21	9	30	18	
Service requests arriving within the time period [t = 40, t = 80]									
Node	n34	n38	n46	n50	n26	n27	n43	n41	
a_i	100	80	93	105	117	80	115	80	
b_i	143	145	150	152	156	156	158	159	
Load	14	16	1	13	17	16	7	5	
Service requests arriving within the time period [t = 80, t =120]									
Node	n37	n42	n32	n36	n35	N33	n39	n45	n48
a_i	120	120	120	120	120	120	120	120	125
b_i	160	172	175	178	178	179	186	189	192
Load	8	5	23	5	8	11	31	16	36

Node locations are detailed in Solomon (1987)

Results for this routes & schedules adjust are summarized in Table 2 and depicted in Fig. 2. It can be noted that node n13 previously assigned to vehicle V1 was reallocated to V6. Similarly, old nodes n1, n6, n7, n9, n20 and n24 were inserted in other routes. The second routing update is performed at t = 80 to incorporate the service calls received within the time period [t = 40, t = 80] while permitting the use of the additional vehicle V8. Nonetheless, time window constraint violations cannot be avoided during the second update and their sizes are given by: Δb_{n27} = 20.90, Δb_{n43} = 3.30, Δb_{n50} = 1.00. The last solution update was performed at t = 120 to meet service requests received during the period [t = 80, t = 120]. Several time window (Δb_{n50} = 1.00, Δb_{n27} = 20.90, Δb_{n24}= 27.30, Δb_{n43} = 3.30, Δt_{n42} = 8.40) and a maximum service time constraint (Δt_{v7} = 27.30) violation arise. This indicates that at least one more vehicle is necessary to avoid customer's dissatisfactions. The final tours assigned to each vehicle are the following:

V1: D - n2 - n15 - n40 – n26 – n37 – n45 – D
V2: D - n31 – n7 – n18 – n8 – n46 – n36 – n48 – D
V3: D - n12 – n30 – n20 – n34 – n35 – n1 – D
V4: D - n5 – n17 – n16 – n44 – n6 – n33 – D
V5: D - n19 – n47 – n49 – n11 – n10 – n32 – D
V6: D - n21 – n4 – n25 – n23 – n22 – n41 – n39 – D
V7: D - n28 – n29 – n9 – n3 – n50 – n27 – n24 – D
V8: D - n14 – n38 – n43 – n42 – n13 – D

Computational data for successive solved problems are summarized in Table 3. This table also reports the customer-status at each update-time and problem-resolution-times (including both re-assignment and re-sequencing stages).

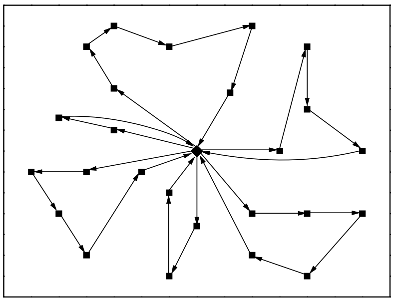
Table 2: Vehicle routing & schedules for the dynamic instance of problem R-110

Vehicle	Node	Waiting Time	Arrival time	Load
Initial solution				
V1	n2	2.00	20.00	
	n15	0.00	43.00	
	n13	0.00	73.00	38
V2	n18	61.20	77.00	
	n8	0.00	97.40	21
V3	n12	23.00	38.00	
	n9	0.00	73.50	
	n3	7.50	106.00	
	n24	0.00	130.10	51
V4	n5	0.00	20.60	
	n17	10.40	51.00	
	n16	0.00	72.20	
	n14	0.00	93.40	
	n6	0.00	125.80	70
V5	n7	44.80	66.00	
	n19	0.00	87.20	
	n11	0.00	104.30	
	n10	0.00	125.50	
	n20	0.00	151.30	
	n1	0.00	177.80	69
V6	n21	19.00	37.00	
	n4	14.00	71.00	
	n25	0.00	91.00	
	n23	0.00	119.00	
	n22	0.00	140.20	83
First solution update (*t* = 40)				
V1	n15	0.00	43.00	
	n40	0.00	75.40	24
V2	n31	0.00	57.50	
	n7	0.00	78.70	
	n18	0.00	98.70	
	n8	0.00	119.10	53
V3	n30	0.00	74.90	
	n3	0.00	106.10	
	n20	0.00	138.50	
	n1	0.00	165.00	72
V4	n17	10.40	51.00	
	n16	0.00	72.20	
	n44	0.00	88.30	
	n14	0.00	104.00	
	n6	0.00	136.40	88
V5	n19	21.00	53.00	
	n47	0.00	74.20	
	n49	0.00	96.40	
	n11	0.00	120.70	
	n10	0.00	141.90	102
V6	n4	14.00	71.00	
	n25	0.00	91.00	
	n23	0.00	119.00	
	n22	0.00	140.20	
	n13	0.00	171.40	106
V7	n28	0.00	46.30	
	n29	0.00	79.80	
	n9	0.00	109.90	
	n24	0.00	146.80	44
Second solution update (*t* = 80)				
V1	t26	24.50	117.00	41
V2	t18	0.00	98.70	
	t8	0.00	119.10	
	t46	0.00	138.40	54
V3	t20	17.00	109.00	
	t34	0.00	141.40	
	t24	0.00	171.40	66
V4	t44	0.00	88.30	
	t6	0.00	119.60	68
V5	t49	0.00	93.30	
	t11	0.00	117.60	
	t10	0.00	138.80	
	t1	0.00	164.40	112
V6	t25	0.00	91.00	
	t23	0.00	119.00	
	t22	0.00	140.20	
	t41	0.00	154.40	88
V7	t9	0.00	109.90	
	t3	0.00	134.80	
	t50*	0.00	153.00	
	t27*	0.00	176.90	83
V8	t14	0.00	112.00	
	t38	0.00	133.20	
	t43*	0.00	161.30	
	t13	0.00	194.40	66
Third solution update (*t* = 120)				
V1	t37	0.00	153.90	
	t45	0.00	186.70	65
V2	t46	0.00	138.40	
	t36	0.00	160.40	
	t48	0.00	184.00	85
V3	t34	0.00	141.40	
	t35	0.00	161.60	
	t1	0.00	198.80	81
V4	t33	0.00	165.20	79
V5	t10	0.00	138.80	
	t32	0.00	159.10	125
V6	t22	0.00	140.20	
	t41	0.00	154.40	
	t39	0.00	183.10	119
V7**	t3	0.00	134.80	
	t50*	0.00	153.00	
	t27*	0.00	176.90	
	t24*	0.00	217.30	86
V8	t38	0.00	133.20	
	t43	0.00	161.30	
	t42*	0.00	180.40	
	t13	0.00	194.70	71

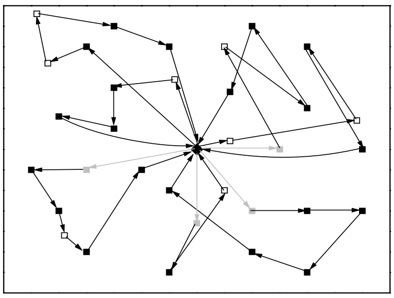
DISCUSSION AND CONCLUSIONS

In this work, a novel MILP model-based algorithmic procedure for solving the dynamic version of the VRPTW problem has been developed. The proposed DVRPTW approach is capable of handling multiple depots and heterogeneous vehicles fleets. The MILP problem representation was embedded into an *insertion & local search* solution strategy. In this way, each time the vehicle routes are updated, the DVRPTW formulation is tackled by using a two-step solution strategy. First, dynamically revealed service requests are inserted in the current routes while allowing some degree of re-assignment of scheduled nodes. However, the precedence relationships among old nodes on every tour are preserved during Step 1. A better solution is subsequently found by swapping nodes of any given tour at Step 2. A relatively hard DVRPTW instance that initially involves 25 nodes and then gradually incorporates 25 further customers to service with a fleet of 8 vehicles was successfully solved. Vehicle routes have been updated three times while the vehicles were on duty. Efficient tours were obtained through solving MILP formulations with rather low requirement of binary variables. Therefore, the proposed approach seems to be very promising to utilize it in "real time" environments.

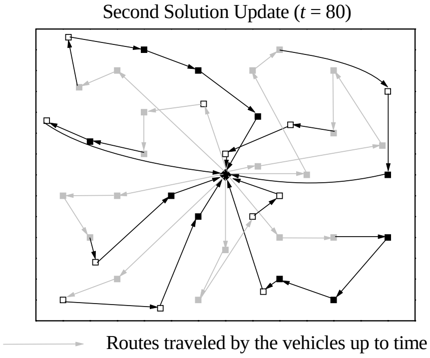
Initial Solution (*t* = 0)



First Solution Update (*t* = 40)

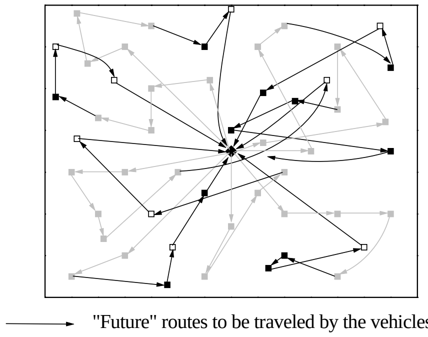


Second Solution Update (*t* = 80)



Routes traveled by the vehicles up to time *t*.

Third Solution Update (*t* = 120)



"Future" routes to be traveled by the vehicles

Figure 2: Geographical view of successive solution updates

Table 3: Computational results for successive solution updates

Schedule update time	Nodes status			Stage	Problem size			CPU time(s)
	S	R	I		B	R	C	
t = 40	4	21	8	Insertion	89	156	1597	
				Reorder	61	156	100	60.4
t = 80	12	17	8	Insertion	83	126	1849	
				Reorder	20	126	54	62.3
t = 120	12	13	9	Insertion	75	108	1434	
				Reorder	18	108	49	5.2

S: Nodes serviced up to the next update-time; B: Binary variables
R: Nodes subject to re-programming; R: Real variables
I: Inserted nodes; C: Constraints

References

Dondo R and J. Cerdá. (2005) "A reactive MILP approach to the Vehicle Routing Problem with Time Windows". Sixth International Conference on Industrial Logistics (ICIL 2005), 91-102. Montevideo February 14-18.

ILOG OPL Studio 3.7. (2003) *User´s Manual*. ILOG S.A. France..

Solomon M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Oper. Res. 32, 254-265.*

Received: December 19, 2005.
Accepted for publication: July 12, 2006.
Recommended by Editor A. Bandoni.