

# Optimal management of logistic activities in multi-site environments

Rodolfo Dondo, Carlos A. Méndez, Jaime Cerdá\*

*INTEC (UNL-CONICET), Güemes 3450, 3000 Santa Fe, Argentina*

Received 3 May 2007; received in revised form 21 August 2007; accepted 4 October 2007

Available online 18 October 2007

## Abstract

The new emerging area of Enterprise Wide Optimization (EWO) has focused the attention in effectively solving the combined production/distribution scheduling problem. The importance of logistic activities performed in multi-site environments comes from the relative magnitude of the associated transportation costs and the good chance of getting large savings on such expenses. This paper first develops an exact MILP mathematical formulation for the multiple vehicle time-window-constrained pickup and delivery (MVPDPTW) problem. The approach is able to account for many-to-many transportation requests, pure pickup and delivery tasks, heterogeneous vehicles and multiple depots. Optimal solutions for a variety of benchmark problems with cluster/random distributions of pickup and delivery locations and limited sizes in terms of customer requests and vehicles have been discovered. However, the computational cost exponentially grows with the number of requests. For large-scale *m*-PDPTW problems, a local search improvement algorithm steadily providing a better solution through two evolutionary steps is also presented. A neighborhood structure around the starting solution is generated by first allowing multiple request exchanges among nearby trips and then permitting the reordering of nodes on every individual route. If a better set of routes is found, both steps are repeated until no improved solution is discovered. Compact MILP mathematical formulations for both sub-problems have been developed and solved through an efficient branch-and-bound algorithm. A significant number of large-scale *m*-PDPTW benchmark problems, some of them including up to 100 transportation requests, were successfully solved in reasonable CPU times.

© 2007 Elsevier Ltd. All rights reserved.

**Keywords:** Pickup and delivery problem; Optimization; Multiple vehicles; Logistics

## 1. Introduction

Major cost factors within a supply chain come from production, material handling, inventory and distribution tasks. Although their relative importance may largely vary with the type of industry, the current trend towards a geographically distributed business requires regional and global coordinated operations. Decentralized production and distribution schemes based on multi-site environments are becoming more popular and, consequently, goods are almost never produced and consumed in the same geographical location. As a result, transport and distribution activities are emerging as central issues in many of today's companies because of the significant increase in their complexity and costs. Consequently, the development of effective computational tools for logistics management has attracted a great attention from both industry and academy on the field

of supply chain management (SCM) and, more recently, on the new emerging area of Enterprise Wide Optimization (EWO).

One of the most typical transportation problems studied in the literature is the so-called *vehicle routing problem* (VRP) which deals with the delivery (collection) of goods by a fleet of trucks from (to) a central or multiple depots to (from) many customer locations. In the general case, the main goal of the VRP problem is to generate the optimal routes for the vehicle fleet based on a given road network so as to meet customer demands and satisfy capacity and time constraints at minimum travel cost (Dondo, Méndez & Cerdá, 2003). A generalization of the VRP, called the *pickup and delivery problem* (PDP), has been intensively studied in the last twenty years. It is a combinatorial optimization problem aimed at satisfying a set of customer requests involving simultaneously pickup and delivery tasks by means of a vehicle fleet at minimum cost. Each customer request specifies the size of the load to be transported, the locations where the goods are to be collected (the origins) and the locations to which are to be delivered (the destinations). Each demand has to be fulfilled by a single vehicle transporting goods from origins to destinations

\* Corresponding author. Tel.: +54 342 4559175; fax: +54 342 4550944.  
E-mail address: [jcerda@intec.unl.edu.ar](mailto:jcerda@intec.unl.edu.ar) (J. Cerdá).

## Nomenclature

### Sets

$I$	nodes
$I_r$	nodes related to transport request $r$
$I^+$	pickup nodes
$I_r^+$	pickup nodes related to request $r$
$I^-$	delivery nodes
$I_r^-$	delivery nodes related to request $r$
$I_v^F$	fixed nodes already allocated to vehicle $v$
$P$	depots
$R$	transport requests that may involve multiple pickup nodes $i \in I_r^+$ and multiple delivery nodes $i' \in I_r^-$
$R^M$	subset of mobile requests that can be reallocated in sub-problems I and III
$R_v^M$	subset of mobile requests that can be reallocated to vehicle $v$
$R^F$	subset of fixed requests that cannot be reallocated to another vehicle in sub-problems I and III
$R_v^F$	subset of fixed requests already allocated to vehicle $v$
$V$	vehicles
$V_p$	vehicles already allocated to depot $p$
$V_r$	vehicles that can be allocated to request $r$

### Parameters

$a_i$	earliest service time at node $i$
$b_i$	latest service time at node $i$
$c_{ii'}^v$	vehicle-dependent travel cost between nodes $i$ and $i'$
$c_{pi}^v$	vehicle-dependent travel cost between depot and node $i \in I$
$cf_v$	$v$ th-vehicle fixed utilization cost
$M_C$	maximum vehicle traveling cost
$M_L$	maximum vehicle capacity
$M_T$	maximum vehicle arrival time
$q_v$	$v$ th-vehicle capacity
$st_i$	service time at node $i$
$t_v^{\max}$	maximum routing time for vehicle $v$
$t_{ii'}^v$	vehicle-dependent travel time between nodes $i$ and $i'$
$t_{pi}^v$	vehicle-dependent travel time between depot $p$ and node $i$
$\alpha_{ir}$	load to pickup at node $i \in I_r^+$ . If $i \in I_r^-$ then $\alpha_{ir} = 0$
$\beta_{ir}$	load to deliver at node $i \in I_r^-$ . If $i \in I_r^+$ then $\beta_{ir} = 0$
$\lambda$	penalty factor for earliness
$\Lambda$	penalty factor for routing time constraint violation
$\rho_i$	unit penalty cost for time-window violation at node $i$
$\rho_v$	penalty factor for routing time violation of vehicle $v$
$\omega$	penalty factor for tardiness

### Binary variables

$S_{ii'}$	binary variable denoting that node $i$ is visited before ( $S_{ii'} = 1$ ) or after ( $S_{ii'} = 0$ ) node $i'$ whenever both nodes are serviced by the same vehicle
$X_{vp}$	binary variable denoting the assignment of depot $p$ to vehicle $v$
$Y_{rv}$	binary variable denoting the assignment of transportation request $r$ to vehicle $v$

### Continuous variables

$B_i$	tardiness at node $i$
$C_i$	accumulated vehicle routing cost to reach node $i$
$E_i$	anticipation at node $i$
$L_i$	total cargo loaded on the assigned vehicle after completing the service at node $i$
$OC_v$	total routing cost for vehicle $v$
$OD_v$	maximum routing time violation for vehicle $v$
$OT_v$	total travel time for vehicle $v$
$T_i$	accumulated vehicle travel time to reach node $i$
$U_i$	total cargo unloaded from the assigned vehicle after completing the service at node $i$

without any transshipment at other locations. To accomplish the assigned transport requests, every vehicle departs from the central base and visits a number of sites along the selected route. At each stop location, the vehicle can either pick up or deliver an amount of load but not both. Loading and unloading times are incurred at every stop. Moreover, each vehicle starts and ends its journey at the central or the assigned depot. Vehicle data generally include the transport capacity, the alternative depots (if several ones are considered) and the subset of transport requests that can be allocated to every truck. If the pickup and/or the delivery location has a time interval within which the service must begin, then the problem is known as the pickup and delivery problem with time windows (PDPTW). Time windows are usually defined based on customer preferences. Since real-world pickup and delivery problems include time-window constraints, most of the papers have been focused on the PDPTW. In particular, the PDPTW variant involving transport requests with a single origin and a single destination, and a vehicle fleet departing and returning to a central depot was the most studied. In a more general context, the underlying ideas of the PDPTW problem have been also applied to the coordination of the simultaneous scheduling of production and distribution activities in multi-site systems (Méndez, Bonfill, Espuña, & Puigjaner, 2006).

Surveys on the pickup and delivery problem can be found in Bodin, Golden, Assad, and Ball (1983), Savelsbergh and Sol (1995), Fisher (1995), Desrosiers, Dumas, Solomon, and Soumis (1995), and Desaulniers, Desrosiers, Erdmann, Solomon, and Soumis (2002). Three classes of PDPTW have usually been tackled. One of them is the so-called single-vehicle pickup and delivery problem with time windows (1-PDPTW) where pickup and delivery services are all done by a single vehicle. If there are multiple vehicles available, the problem is known as the multi-

vehicle pickup and delivery problem (*m*-PDPTW). Most of the contributions have been devoted to these two PDPTW classes. In many practical situations, however, the cargo must be collected from multiple nodes and transported to a single delivery location or vice versa. Furthermore, the problem may involve a heterogeneous vehicle fleet, multiple depots and each vehicle may have its own starting/ending base. In addition, pure pickup nodes and/or delivery nodes demanding just a pickup or delivery service but not both can be simultaneously considered. A PDPTW problem with all these features is called the general pickup and delivery problem with multiple vehicle and time windows (*m*-GPDPTW). So far, the *m*-GPDPTW problem has received a rather limited attention. Problems 1-PDPTW, *m*-PDPTW or *m*-GPDPTW all assume a pre-defined set of customer requests that remains unchanged while the pickup and delivery services are being performed. They can be regarded as distinct variants of the static PDPTW. In real-world problems, new customer requests are continually received in real-time and immediately eligible for consideration. As a consequence, the current set of routes has to be re-optimized at some time point to include the new transport services. It is the dynamic *m*-GPDPTW. In any case, the goal is to periodically adjust the present set of routes and schedules in order to account for the updated problem data.

Different practical problems can be modeled as pickup and delivery problems. Among them, the VRP with backhauls (VRPB), the dial-a-ride problem (DARP), the handicapped person transportation problem (HTP) and the courier company pickup and delivery problem (CCPDP). The VRPB includes a set of customers to whom products are to be delivered (pure delivery nodes) and a set of nodes whose goods need to be transported back to the distribution center (pure pickup nodes). Moreover, all deliveries have to be made before all the pickups. Problems DARP and HTP are concerned with the transportation of people while CCPDP deals with transporting messages and parcels. A great deal of papers on PDPTW is on dial-a-ride problems. In contrast, fewer contributions have been focused on the pickup and delivery of packages and goods. In the last years, there is an increasing trend towards modeling such practical problems as particular cases of the *m*-GPDPTW problem formulation.

In this work, a new mixed-integer linear programming formulation for the general *m*-GPDPTW problem is presented. The proposed optimization approach is capable of handling transport requests with multiple origins and/or destinations, heterogeneous vehicles and multiple depots. Moreover, it can still be applied even if pure pickup and/or delivery nodes and multiple depots are also considered. However, the exact MILP approach can solve *m*-GPDPTW problem instances with at most 25 requests to optimality in a reasonable CPU time. In order to tackle *m*-GPDPTW problems with a larger number of requests, a model-based neighborhood search framework that iteratively improves a starting solution is subsequently developed. To do that, a neighborhood structure around the current solution is generated by just allowing (a) multiple exchanges of requests among neighboring trips and (b) reordering of nodes on every tour. The so-defined neighborhood domain is systematically

explored to find the best neighbor by solving a pair of low-size MILP sub-problems. In order to illustrate the performance of the proposed *m*-GPDPTW exact and improvement methods, they both have been applied to a sizable set of PDPTW benchmark examples with different time-window width and geographical node distributions. The results have been compared with the best solutions recently reported in the literature for such benchmark problems.

### 1.1. Previous heuristic approaches

Two types of PDPTW solution methodologies have been proposed: heuristic techniques and exact optimization approaches. Heuristic techniques can be classified into three types: tour construction heuristics, tour improvement heuristics and metaheuristics (Mitrovic-Minic, 1998). *Constructive heuristic algorithms* gradually build a feasible solution by keeping an eye on the solution cost but no improvement phase is next implemented. They can be divided into two groups: *decomposition methods* and *insertion procedures*. The best construction heuristic methods decompose the problem into three phases: clustering, routing and scheduling. Dumas, Desrosiers and Soumis (1989) applied a decomposition approach to the *m*-PDPTW based on the notion of mini-clustering. A mini-cluster is a route segment along which the demands from a small group of clients can be satisfied by a single vehicle starting and ending the route segment service with either the same load or empty. Then, a mini-cluster can be treated as an aggregate transport request entirely satisfied by a single vehicle. The optimization algorithm is then applied to a more compact set of transportation requests. Mini-clusters are constructed from known pickup and delivery plans by cutting them into pieces such that each piece starts and ends with an empty vehicle. In turn, *insertion heuristics* develop a set of routes by inserting one or multiple requests into one or several open routes at a time (Altinél & Öncan, 2005; Ioannou, Kritikos, & Prastacos, 2001; Jaw, Odon, Psaraftis, & Wilson, 1986; Lu & Dessouky, 2006; Toth & Vigo, 1997). The choice of which request to insert and where to insert it depends on the heuristic being applied.

On the other hand, *improvement heuristic procedures* start from a set of feasible routes and apply some kind of local search heuristics to get a better solution. By repeatedly performing a sequence of steps, it generates and examines a neighborhood around the current solution to hopefully find an improved set of routes. If so, the better solution is adopted as the new starting point and the process is repeated. The notion of neighbor depends on the improvement heuristic being applied (Thompson & Psaraftis, 1993). More recently, *metaheuristic methods*, including simulated annealing (Van Der Bruggen, Lenstra, & Schuur, 1993), constrained-direct search (Potvin & Rousseau, 1995), tabu search (Nanry & Wesley Barnes, 2000; Taillard et al., 1997; Tang Montané & Diéguez Galvao, 2006), threshold algorithms, neural networks (Potvin, Shen, & Rousseau, 1992) and genetic algorithms have also been proposed. Li and Lim (2003a, 2003b) developed a tabu-embedded simulated annealing algorithm for the general multiple-vehicle PDPTW which restarts a search procedure from the current

best solution after several non-improving search iterations. In addition, they generated a set of test cases for PDPTW based on Solomon's benchmark instances for VRPTW (Solomon, 1987). Tam and Kwan (2004) generated a systematic scheme to adapt the large neighborhood search (NLS) to efficiently solve PDPTW problems. The NLS is an iterative process of relaxation and re-optimization to continually improve on the current routing plan until the convergence to a local minimum or the resource exhaustion occurs. Their results compare favorably with those obtained with Li and Lim's metaheuristic search algorithm. Bent and Van Hentenryck (2006) developed a two-stage hybrid methodology for the  $m$ -PDPTW problem that repeatedly applies a simple simulated annealing algorithm to decrease the number of routes followed by a large neighborhood search to decrease the travel cost.

### 1.2. Exact optimization methods

The development of exact optimization methods started in the early 1980s. Generally speaking, the exact approaches can be grouped into two classes: branch-and-price and branch-and-cut methods. Branch-and-price techniques apply a branch-and-bound scheme in which lower bounds are computed by column generation. In branch-and-cut methods, valid inequalities or cuts are incorporated in the formulation at each node of the branch-and-bound tree to tighten the LP relaxation problems. Because of its intrinsic complexity, a limited number of exact methods for the  $m$ -PDPTW has been published. Dumas, Desrosiers, and Soumis (1991) presented a technique based on the Dantzig-Wolfe decomposition/column generation scheme and a pricing sub-problem involving a shortest path problem with capacity, coupling, precedence and time-window constraints. Savelsbergh and Sol (1998) proposed another branch-and-price approach for the PDPTW with some interesting features. It uses construction and improvement heuristics to solve the pricing sub-problem, a primal heuristic at each node of the search tree to compute upper bounds and a clever column handling procedure to keep the column generation master problem as small as possible. Lu and Dessouky (2004) developed an MILP optimisation-based framework and a branch-and-cut algorithm for solving the  $m$ -PDPTW problem. It permits to solve problem instances of up to 5 vehicles and 17 customer requests on problems without clusters. Recently, Cordeau (2006) developed a branch-and-cut algorithm for the dial-and-ride problem based on a three-index formulation.

## 2. The problem definition

The multiple vehicle pickup and delivery problem with time windows consists of finding the optimal routes for a vehicle fleet in order to fulfill a set of customer requests  $r \in R$  at minimum total cost while satisfying all problem constraints. Let us define the set  $R$  containing the transport requests, the set  $I$  comprising all pickup and delivery nodes to be serviced and the set  $V$  including the available vehicles. Moreover,  $I^+$  is the set of pickup nodes ( $I^+ \subset I$ ) while  $I^- (\subset I)$  just comprises the delivery nodes, i.e.  $I = \{I^+ \cup I^-\}$ . Other important sets are the one con-

taining all pickup and delivery nodes related to request  $r (I_r)$  as well as the pickup locations  $I_r^+$  and the delivery locations  $I_r^-$  for request  $r \in R$ . It may occur that the same site is associated to different requests to accomplish a similar or a different task. In such a case, it should be defined as many (pickup or delivery) nodes located on that site as the number of related requests. In other words, every problem node is associated to only one transportation order.

In general terms, a customer request  $r \in R$  involves the transportation of some load from multiple origins to multiple destinations. A given set of pickup nodes  $i \in I_r^+$  and/or the pre-assigned starting depot are usually the origin points. In turn, the destinations may include a given set of delivery nodes  $i' \in I_r^-$  and/or the pre-assigned final terminal. This general definition also accounts for requests involving the transportation of goods to a set of pure delivery nodes from a pre-defined supply depot or the collection of goods from a number of pickup nodes to transport them to a pre-defined end base. Therefore, the data set associated to any request  $r$  includes: (a) an initial cargo  $q_r^o$  to be loaded at the pre-defined starting depot; (b) a group of collection nodes  $i \in I_r^+$  from each one a load  $\alpha_{ir}$  is to be picked up; (c) a number of delivery nodes  $i' \in I_r^-$  to each one a load  $\beta_{i'r}$  is to be delivered and (d) a final cargo  $q_r^f$  destined to the pre-defined end base. Moreover, the data for a transportation request must satisfy the following condition:

$$q_r^o + \sum_{i \in I_r^+} \alpha_{ir} = \sum_{i' \in I_r^-} \beta_{i'r} + q_r^f, \quad \forall r \in R$$

Time windows ( $a_i, b_i$ ) are given for pickup and delivery tasks within which they must be accomplished. Vehicles depart and return to the same depot but the one assigned to a particular vehicle is a problem decision. Vehicle capacities and depot locations are also problem data. While the vehicles perform the pickup and delivery tasks, numerous constraints are to be satisfied. They are the following: (a) every used vehicle has its assigned depot (depot assignment constraint); (b) each customer request  $r \in R$  must be serviced by a single vehicle visiting a pickup node first except for pure delivery nodes (vehicle assignment, pairing and precedence constraints); (c) the capacity of a vehicle can never be exceeded after visiting a pickup node (capacity constraints at pickup nodes); (d) a vehicle must transport enough load to meet customer demand when stopping at a delivery node (capacity constraints at delivery nodes); (e) each used vehicle should return to its base (final destination constraints); (f) the total time/distance traveled by a vehicle from the starting depot to a particular node location must be greater than the one required to reach a preceding node on the tour (time-based sequencing constraints); (g) the service at each node must be started within the specified time window (hard time-window constraints); (h) the duration of any tour must never exceed a maximum travel time  $t_v^{\max}$ . In some practical VRPTW problems, time-window constraints can be violated but at the expense of a penalty cost payment (soft time-window constraints).

The problem goal is to minimize the total cost of providing pickup or delivery service to every node. Three different types of costs are usually considered. First, the total vehicle

fixed costs, including acquisition and maintenance expenses, aimed at minimizing the number of used vehicles. Second, the distance-based and/or the time-based transportation cost accounting for the fuel consumption, vehicle maintenance and driver wages. Third, the customer inconvenience originated by pickups or deliveries performed either sooner or later than the desired service time windows. The customer inconvenience is usually regarded as a linear function of the time-window constraint violations. Often, the authors selected a weighted combination of the travel distance cost, the service time cost and the customer's inconvenience cost as the problem objective to minimize.

### 3. The problem mathematical formulation

Consider a road network represented by a graph  $G\{I^+, I^-, P, R, A\}$ , where  $I^+ = \{i_1, i_2, \dots, i_n\}$  denotes the set of pickup nodes,  $I^- = \{i'_1, i'_2, \dots, i'_n\}$  is the set of delivery nodes,  $P = \{p_1, p_2, \dots, p_l\}$  represents the set of depots,  $R = \{r_1, r_2, \dots, r_s\}$  stands for the transportation requests each one featuring a set of pickup and delivery nodes  $I_r = \{I_r^+ \cup I_r^-\}$ , and  $A = \{a_{ii'} / (i, i') \in I^+ \cup I^- \cup P\}$  defining the set of minimum cost arcs between nodes. To fulfill pickup and delivery tasks, a set of vehicles  $V = \{v_1, v_2, \dots, v_m\}$  is available. The service time at location  $k \in \{I^+ \cup I^-\}$  by vehicle  $v$  is denoted by  $st_k^v$ . In addition, there is a vehicle-dependent distance-based traveling cost matrix  $C = \{c_{ii'}\}^v$  and a vehicle-dependent travel time matrix  $\Gamma = \{t_{ii'}\}^v$  both associated to the set  $A$ . For a pickup node  $i \in I_r^+$  related to request  $r$ , there is a given load  $\alpha_{ir}$  to be collected within a time window  $[a_i, b_i]$ . Similarly, a given load  $\beta_{i'r}$  is to be delivered to node  $i' \in I_r^-$  within a time window  $[a_{i'}, b_{i'}]$ .

Three types of decision variables are included in the mathematical model: depot-to-vehicle assignment variables ( $X_{pv}$ ), vehicle-to-request allocation variables ( $Y_{rv}$ ) and node sequencing variables ( $S_{ii'}$ ). Binary variable  $X_{pv}$  is equal one if depot  $p \in P$  is assigned to vehicle  $v \in V$ . If request  $r \in R$  is serviced by vehicle  $v$ , then the 0–1 variable  $Y_{rv}$  becomes equal to one. In turn, the sequencing variable  $S_{ii'}$  is equal one whenever the pair of nodes  $i, i' \in I$  are on the same route and node  $i$  is visited earlier. The variable  $S_{ii'}$  is defined for any pair of nodes, regardless of whether they are pickup or delivery nodes.

The proposed MILP formulation also includes six important non-negative continuous variables:  $C_i$ ,  $OC_v$ ,  $T_i$ ,  $OT_v$ ,  $L_i$  and  $U_i$ . Variable  $C_i$  is the distance-based transport cost from the starting depot to node  $i \in I_r$  along the route assigned to the vehicle servicing request  $r$ . The travel time to go from the starting depot to node  $i \in I$  is given by  $T_i$ . In turn,  $OC_v$  is the overall routing cost for the tour assigned to vehicle  $v$ , and  $OT_v$  is the total time required by vehicle  $v$  to complete the tour. The overall load collected by vehicle  $v$  while going from the starting depot up to node  $i \in I$ , including the initial cargo, is given by  $L_i$ . In contrast,  $U_i$  is the overall cargo delivered by the assigned vehicle from the starting depot to node  $i \in I$ . Additional continuous variables whose values become set by the previously defined variables include the time-window constraint violations ( $E_i, B_i$ ) and the maximum routing time constraint violation  $OD_v$ .

### 3.1. The problem constraints

#### 3.1.1. Assignment constraints

- **Allocation of depots to vehicles:** If used, a vehicle  $v \in V$  must be assigned to a single depot  $p \in P$  from which it starts the tour and to which it will return after accomplishing all the assigned pickup/delivery tasks.

$$\sum_{p \in P} X_{vp} \leq 1, \quad \forall v \in V \quad (1)$$

- **Assignment of vehicles to transportation requests:** Each transportation request  $r \in R$  must be fulfilled by just a single vehicle  $v \in V$ .

$$\sum_{v \in V} Y_{rv} = 1, \quad \forall r \in R \quad (2)$$

If some request  $r$  has been pre-assigned to depot  $p'$ , it should then be visited by a vehicle  $v \in V$  allocated to depot  $p'$ , i.e.  $X_{vp'} = 1$ . Therefore,  $Y_{rv} \leq X_{vp'}, \forall v \in V$ .

- **Vehicle availability condition:** An unused vehicle  $v \in V$  has no designated depot ( $X_{vp} = 0$ ). If  $X_{vp} = 0$ , then the vehicle  $v$  is not available to fulfill any customer request  $r \in R$ . Consequently, none of the variables  $Y_{rv}$  can be equal one as prescribed by Eq. (3). The parameter  $M_X$  is an upper bound on the number of transport requests allocated to any vehicle  $v$ .

$$\sum_{r \in R} Y_{rv} \leq M_X \sum_{p \in P} X_{vp}, \quad \forall v \in V \quad (3)$$

Alternatively, the constraint (3) can be equivalently so written:

$$Y_{rv} \leq \sum_{p \in P} X_{vp}, \quad \forall r \in R, v \in V \quad (3')$$

#### 3.1.2. Routing-cost defining constraints

- **Minimum routing cost from the vehicle base to node  $i$ :** If node  $i \in I_r$  is serviced by vehicle  $v$  ( $Y_{rv} = 1$ ) housed in depot  $p$  ( $X_{pv} = 1$ ), then the traveling cost from depot  $p$  to node  $i$  ( $C_i$ ) must always be greater than or equal to  $c_{pi}^v$ . The parameter  $c_{pi}^v$  represents the least travel cost from depot  $p$  to node  $i$ . The value of  $C_i$  will be exactly equal to  $c_{pi}^v$  if node  $i$  is the first one visited by the assigned vehicle  $v$  and depot  $p$  is the selected base for  $v$ .

$$C_i \geq c_{pi}^v (X_{vp} + Y_{rv} - 1), \quad \forall i \in I, r \in R, v \in V, p \in P \quad (4)$$

If vehicle  $v$  starts the trip empty, then a pickup node  $i \in I_r^+$  for some request  $r$  will first be visited. In some cases, however, the PDP problem involves transport requests with a positive  $q_r^0$  and, therefore, some vehicles will leave their bases transporting a finite load and may visit a delivery node  $i' \in I_r^-$  first.

- **Distance-cost-based sequencing constraints.** Let  $c_{ii'}^v$  stand for the least travel cost from node  $i \in I_r$  to node  $i' \in I_r$  ( $i' \neq i$ )

on vehicle  $v$ , where  $r, r' \in R$  and  $I_r$  includes pickup/delivery nodes related to request  $r$ . If both nodes  $(i, i')$  are on the same tour ( $Y_{rv} = Y_{r'v} = 1$ , for some vehicle  $v$ ) and node  $i$  is earlier visited ( $S_{ii'} = 1$ ), then the travel cost from the base to node  $i'$  ( $C_{i'}$ ) must always be greater than  $C_i$  by at least  $c_{ii'}^v$ . If node  $i$  is visited later ( $S_{ii'} = 0$ ), the reverse statement holds. Such conditions are enforced by constraints (5a) and (5b) which become redundant whenever nodes  $(i, i') \in I$  are serviced by different vehicles ( $Y_{rv} + Y_{r'v} < 2$ ). By definition,  $M_C$  is an upper bound on the travel cost from the depot to any node  $i \in I$ . It is important to remark that just a single sequencing variable  $S_{ii'} (i < i')$  is to be defined for every pair of nodes  $(i, i')$ .

$$C_{i'} \geq C_i + c_{ii'}^v - M_C(1 - S_{ii'}) - M_C(2 - Y_{rv} - Y_{r'v}),$$

$$\forall i \in I_r, i' \in I_{r'} (i < i'), r, r' \in R, v \in V \quad (5a)$$

$$C_i \geq C_{i'} + c_{i'i}^v - M_C S_{ii'} - M_C(2 - Y_{rv} - Y_{r'v}),$$

$$\forall i \in I_r, i' \in I_{r'} (i < i'), r, r' \in R, v \in V \quad (5b)$$

In case  $r=r'$  and every customer request involves a single pickup node  $i \in I_r^+$  and a single delivery node  $i' \in I_r^-$ , then pickup node  $i$  should be visited earlier and  $S_{ii'} = 1$  assuming  $i < i'$ . Therefore, the distance-based sequencing constraints ((5a) and (5b)) reduce to the following pairing condition:

$$C_{i'} \geq C_i + c_{ii'}^v - M_C(1 - Y_{rv}),$$

$$\forall i \in I_r^+, i' \in I_r^-, r \in R, v \in V \quad (5')$$

- **Overall routing cost for the tour assigned to vehicle  $v$ :** The overall traveling cost incurred by vehicle  $v$  ( $OC_v$ ) to satisfy the assigned requests  $r \in R_v$  must always be greater than the traveling expenses from the origin-depot  $p$  to any node  $i \in I_r$  on the tour (i.e.  $C_i$ ) by at least the amount  $c_{ip}^v$  standing for the travel cost from node  $i$  to depot  $p$ . Indeed, the last node visited by vehicle  $v$  is the one defining the largest value of  $OC_v$  and therefore the constraint (6) for such a node and the visiting vehicle  $v$  is just binding at the optimum. If nodes  $i, i' \in I_r$  are both on the tour and  $i'$  is the last visited, then constraint (6) for node  $i$  will become redundant because the travel cost  $c_{ip}^v$ , by definition, is smaller than or at most equal to the traveling expenses from  $i$  to  $p$  through at least another node  $i'$ , i.e.  $c_{ip}^v \leq c_{ii'}^v + c_{i'p}^v$ .

$$OC_v \geq C_i + \sum_{p \in P} c_{ip}^v X_{vp} - M_C(1 - Y_{rv}),$$

$$\forall i \in I_r, r \in R, v \in V \quad (6)$$

### 3.1.3. Arrival-time defining constraints

- **Earliest visiting time for node  $i$ :** The assigned vehicle  $v$  will never arrive at node  $i \in I$  before time  $t_{pi}^v$ , where  $t_{pi}^v$  is the least travel time from depot  $p$  to node  $i$ . Constraint (7) assumes that vehicle  $v$  is ready at  $t=0$ . Otherwise, the ready time for vehicle  $v$  should be added to  $t_{pi}^v$ .

$$T_i \geq t_{pi}^v (X_{vp} + Y_{rv} - 1), \quad \forall i \in I_r, r \in R, v \in V, p \in P \quad (7)$$

- **Time-based sequencing constraints:** Let us assume that nodes  $i \in I_r$  and  $i' \in I_{r'} (i' \neq i)$  are both serviced by the same vehicle  $v$ . If node  $i$  is visited before ( $S_{ii'} = 1$ ), then the arrival time at node  $i'$  ( $T_{i'}$ ) should be greater than  $T_i$  by at least the sum of both the service time ( $st_i$ ) at node  $i$  and the traveling time  $t_{ii'}^v$  from  $i$  to  $i'$ . If not ( $S_{ii'} = 0$ ), the reverse statement holds. If one of the nodes is not on the tour, then  $Y_{rv} + Y_{r'v} < 2$  for any vehicle  $v$  and, therefore, constraints (8a) and (8b) both become redundant.  $M_T$  is an upper bound on the duration of any tour.

$$T_{i'} \geq T_i + st_i + t_{ii'}^v - M_T(1 - S_{ii'}) - M_T(2 - Y_{rv} - Y_{r'v}) \quad (8a)$$

$$T_i \geq T_{i'} + st_{i'} + t_{i'i}^v - M_T S_{ii'} - M_T(2 - Y_{rv} - Y_{r'v}),$$

$$\forall i \in I_r, i' \in I_{r'} (i < i'), r, r' \in R, v \in V \quad (8b)$$

In case  $r=r'$  and every customer request involves a single pickup node  $i \in I_r^+$  and a single delivery node  $i' \in I_r^-$ , then node  $i$  should be visited before node  $i'$  and  $S_{ii'} = 1$ . Moreover, the time-based sequencing constraints (8a) and (8b) reduce to the following pairing condition:

$$T_{i'} \geq T_i + t_{ii'}^v - M_T(1 - Y_{rv}),$$

$$((\forall i \in I_r^+, i' \in I_r^-, r \in R, v \in V) \quad (8c)$$

- **Overall routing time along the tour assigned to vehicle  $v$ :** The total time required by vehicle  $v$  to accomplish the assigned transport requests is found by adding both the service time  $st_i$  and the travel time  $t_{ip}^v$  along the edge  $(i, p)$  to the arrival time at the node last visited. Since the last node visited by vehicle  $v$  is not known beforehand, then Eq. (9) is written for every node  $i \in I_r$ .

$$OT_v \geq T_i + st_i + \sum_{p \in P} t_{ip}^v X_{vp} - M_T(1 - Y_{rv}),$$

$$\forall i \in I_r, r \in R, v \in V \quad (9)$$

- **Time-window constraint-violation due to early service at node  $i \in I_r$**

$$E_i \geq a_i - T_i, \quad \forall i \in I_r, r \in R \quad (10)$$

If the assigned vehicle arrives early at node  $i$ , it can wait until the  $i$ th-time window is open ( $T_i = a_i$ ) to avoid paying a penalty cost.

- **Time-window constraint-violation due to late service at node  $i$**

$$B_i \geq T_i - b_i, \quad \forall i \in I_r, r \in R \quad (11)$$

- **Maximum service time constraint-violation for vehicle  $v$**

$$OD_v \geq OT_v - t_v^{\max}, \quad \forall v \in V \quad (12)$$

Variables  $E_i, B_i$  and  $OD_v$  are all non-negative.

### 3.1.4. Vehicle-load defining constraints

- **Capacity constraint on the load transported by vehicle  $v$  after visiting node  $i$ :** Eq. (13a) states that the cargo transported by vehicle  $v \in V$  just after servicing a pickup node  $i \in I_r^+$  ( $Y_{rv} = 1$ ) must not exceed the vehicle capacity  $q_v$ . Such a load can be computed as the difference between non-negative variables  $L_i$  and  $U_i$ . Variable  $L_i$  stands for the overall load collected by vehicle  $v$  along the route, including the initial cargo, just after servicing node  $i$  while  $U_i$  represents the overall cargo unloaded from vehicle  $v$  from the starting depot up to node  $i$ . The capacity of vehicle  $v$  is given by  $q_v$ . In turn, Eq. (13b) specifies that the cargo transported by vehicle  $v \in V$  after servicing a delivery node  $i' \in I_r^-$  ( $Y_{rv} = 1$ ) must be non-negative.

$$L_i - U_i \leq \sum_{v \in V} q_v Y_{rv}, \quad \forall i \in I_r^+, \quad r \in R \quad (13a)$$

$$L_i - U_i \geq 0, \quad \forall i \in I_r^-, \quad r \in R \quad (13b)$$

- **Load-based sequencing constraints:** Let us assume that nodes  $(i, i') \in I$  are linked to requests  $r \in R$  and  $r' \in R$ , respectively, both serviced by vehicle  $v \in V$  ( $Y_{rv} + Y_{r'v} = 2$ ). If node  $i$  is visited before node  $i'$  ( $S_{ii'} = 1$ ), then the total cargo loaded on vehicle  $v$  from the starting depot to node  $i'$  ( $L_{i'}$ ) should be greater than  $L_i$  by at least  $\alpha_{i'r}$ . Such a condition is enforced by Eq. (14a). The parameter  $\alpha_{i'r}$  is zero if  $i'$  is a delivery node. Conversely, if node  $i' \in I_r'$  is visited earlier, then Eq. (14b) holds and  $L_i$  must be greater than  $L_{i'}$  by at least  $\alpha_{ir}$ . Similarly, Eq. (15a) states that the total load delivered by vehicle  $v$  from the starting depot up to node  $i' \in I_r'$  ( $U_{i'}$ ) should be greater than  $U_i$  by at least  $\beta_{i'r}$  if both nodes are serviced by  $v$  and node  $i$  is visited earlier. The parameter  $\beta_{i'r}$  is zero if  $i'$  is a pickup node. If node  $i \in I$  is visited later ( $S_{ii'} = 0$ ), the reverse condition holds and Eq. (15b) may become active. If one of the requests  $r, r' \in R$  or both are not serviced by vehicle  $v$ , then  $Y_{rv} + Y_{r'v} < 2$ , and all constraints (14) and (15) become redundant. By definition  $M_L$  is an upper bound on the total load collected or delivered by a vehicle along the assigned tour.

$$L_{i'} \geq L_i + \alpha_{i'r} - M_L(1 - S_{ii'}) - M_L(2 - Y_{rv} - Y_{r'v}), \\ i \in I_r, \quad i' \in I_r'(i < i'), \quad r, r' \in R, \quad v \in V \quad (14a)$$

$$L_i \geq L_{i'} + \alpha_{ir} - M_L S_{ii'} - M_L(2 - Y_{rv} - Y_{r'v}), \\ i \in I_r, \quad i' \in I_r'(i < i'), \quad r, r' \in R, \quad v \in V \quad (14b)$$

$$U_{i'} \geq U_i + \beta_{i'r} - M_L(1 - S_{ii'}) - M_L(2 - Y_{rv} - Y_{r'v}), \\ i \in I_r, \quad i' \in I_r'(i < i'), \quad r, r' \in R, \quad v \in V \quad (15a)$$

$$U_i \geq U_{i'} + \beta_{ir} - M_L S_{ii'} - M_L(2 - Y_{rv} - Y_{r'v}), \\ i \in I_r, \quad i' \in I_r'(i < i'), \quad r, r' \in R, \quad v \in V \quad (15b)$$

- **Upper bounds on the values of  $L_i$  and  $U_i$ :** The total cargo loaded on a vehicle  $v$  after visiting a node  $i$  ( $L_i$ ) can never be greater than the total load collected by vehicle  $v$  along the

tour plus the initial cargo with which it leaves the assigned base.

$$L_i - \sum_{r' \in R} q_{r'}^0 Y_{r'v} - \sum_{r' \in R, i' \in I_r^+} \alpha_{i'r'} Y_{r'v} \leq M_L(1 - Y_{rv}), \\ \forall i \in I_r, \quad r \in R, \quad v \in V \quad (16)$$

Similarly, the total cargo unloaded from vehicle  $v$  after visiting a node  $i$  ( $U_i$ ) can never be greater than the total amount of goods delivered along the whole tour.

$$U_i - \sum_{r' \in R, i' \in I_r^-} \beta_{i'r'} Y_{r'v} \leq M_L(1 - Y_{rv}), \\ \forall i \in I_r, \quad r \in R, \quad v \in V \quad (17)$$

- **Lower bounds on the values of  $L_i$  and  $U_i$ :** Eq. (18a) states that the total cargo loaded on the vehicle up to any node  $i \in I_r$  must at least be equal to the initial cargo plus  $\alpha_{ir}$ . The parameter  $\alpha_{ir}$  takes a positive value just for pickup nodes. Symmetrically, Eq. (18b) states that the total cargo unloaded from the vehicle up to node  $i \in I_r$  must at least be equal to  $\beta_{ir}$ . The parameter  $\beta_{ir}$  takes a positive value just for delivery nodes.

$$L_i \geq \sum_{r' \in R} q_{r'}^0 Y_{r'v} + \alpha_{ir} - M_L(1 - Y_{rv}), \quad \forall i \in I_r, \quad r \in R \quad (18a)$$

$$U_i \geq \beta_{ir}, \quad \forall i \in I_r, \quad r \in R \quad (18b)$$

### 3.2. The problem objective function

The problem goal is to minimize a weighted combination of vehicle fixed costs, distance and time-based routing costs as well as some measure of the total customer inconvenience (time-window constraint violations).

$$\min \sum_{v \in V} (OC_v + \mu OT_v + \tau OD_v) + \sum_{i \in I} (\lambda E_i + \omega B_i) \quad (19)$$

Assignment constraints (1)–(3) together with the distance-based sequencing constraints (4)–(6) all define an alternative formulation to the traditional  $m$ -TSP. Timing constraints are considered by the model through time-based sequencing time constraints (7)–(9) and time-window constraints (10)–(12) while capacity constraints are handled through Eqs. (13)–(18). Timing window constraints can be treated as hard constraints by driving the variables  $E_i, B_i$  and  $OD_v$  all to zero in constraints (10)–(12) and removing the associated penalty cost terms from the objective function. The proposed mathematical model can account for heterogeneous fleets and multiple depots.

### 3.3. Time-window-based variable and constraint elimination rules

In order to improve the computational efficiency of the MILP branch-and-bound solution procedure, exact elimination rules based on time-window specifications were proposed to remove a significant number of sequencing variables and constraints from

the mathematical model. In this way, a more compact solution space can be generated. Since they result from assuming hard time windows, the proposed rules can rigorously be applied to just the hard-TW version of the PDPTW problem. The narrower the time windows the larger the impact of the elimination rules on the problem size. To maximize the effect of the elimination rules, time windows are to be shrunk before applying the elimination rules by repeatedly using the so-called time-window contraction rules. They were derived in a straightforward manner from similar rules proposed by Cyrus (1988) and Desrochers, Desrosiers, and Solomon (1992) for the VRPTW problem. Contraction rules just eliminate some early and/or late intervals  $[(a_k^{\text{new}} - a_k^{\text{old}}), (b_k^{\text{old}} - b_k^{\text{new}})]$  from the time window of node  $k$  only if the service at node  $k$  can never start within them. In other words, no feasible route is withdrawn from the solution space and optimality is still guaranteed.

$$a_k^{\text{new}} = \max \left\{ a_k^{\text{old}}, \min_{\substack{a_{\ell k} \in A \\ v \in V}} (a_{\ell} + st_{\ell}^v + t_{\ell k}^v) \right\} \quad (20)$$

$$b_k^{\text{new}} = \min \left\{ b_k^{\text{old}}, \max_{\substack{a_{\ell k} \in A \\ v \in V}} (b_{\ell} + st_{\ell k}^v + t_{\ell k}^v), \max_{\substack{a_{ki} \in A \\ v \in V}} (b_i - st_k^v - t_{ki}^v) \right\} \quad (21)$$

Two types of exact elimination rules are defined. Rules of type I are based on the concept of incompatible transport requests. In turn, rules of type II rely on the fact that some pickup/delivery nodes associated with compatible requests must be serviced in a certain order when visited by the same vehicle.

- **Incompatible pickup/delivery nodes:** Two nodes  $(i, k) \in I$  that cannot be assigned to the same vehicle are called incompatible. Otherwise, a time-window constraint will be violated. They are said to be incompatible if any vehicle coming from node  $i$  at the earliest possible time arrives at node  $k$  when the  $k$ th-time window has already closed and vice versa. The incompatibility conditions for nodes  $(i, k)$  are therefore given by:

$$\begin{aligned} a_i + st_i^v + t_{ik}^v &> b_k \\ a_k + st_k^v + t_{ki}^v &> b_i \end{aligned} \quad (22)$$

where  $(a_i, b_i)$  and  $(a_k, b_k)$  stand for the time-windows of nodes  $i$  and  $k$ , respectively.

- **Incompatible transportation requests:** Two transportation requests  $r: (I_r^+, I_r^-)$  and  $r': (I_{r'}^+, I_{r'}^-)$  that cannot be assigned to the same vehicle are said to be incompatible. If they are serviced by the same vehicle, some time-window constraints will be violated. Requests  $r, r' \in R$  are said to be incompatible if at least a pair of pickup/delivery nodes  $(i, i')$ , with  $i \in I_r$  and  $i' \in I_{r'}$ , is incompatible according to definition (22). For a pair of incompatible requests  $(r, r') \in IR$ , the following condition

holds:

$$Y_{rv} + Y_{r'v} \leq 1, \quad \forall v \in V, (r, r') \in IR \quad (23)$$

If requests  $r$  and  $r'$  cannot be fulfilled by the same vehicle, then Eqs. (5a) and (5b), (8a) and (8b), (14a) and (14b) and (15a) and (15b) for the pair  $(r, r') \in IR$  and their related nodes  $i \in I_r$  and  $i' \in I_{r'}$  together with the related sequencing variables  $S_{ii'}$  can all be deleted from the problem formulation. When a single-vehicle PDPTW problem includes  $n$  pairs of incompatible requests involving a single pickup node and a single delivery node, then  $4n$  sequencing variables and  $(8n + 8n + 2n + 2n) = 20n$  sequencing constraints can be eliminated.

- **Preordered pickup/delivery nodes of compatible requests:** A pair of nodes  $(i, k) \in I$  is said to be preordered one if they should be visited in a certain order by the same vehicle in order to satisfy the time-window constraints. For instance, node  $i$  should be serviced before node  $k$  by the same vehicle if the following conditions hold:

$$a_i + st_i^v + t_{i,k}^v \leq b_k, \quad \forall v \in V \quad (24a)$$

$$a_k + st_k^v + t_{k,i}^v > b_i, \quad \forall v \in V \quad (24b)$$

Let us assume that the pickup/delivery nodes  $i \in I_r$  and  $k \in I_{r'}$  are related to a pair of compatible requests  $(r, r')$  and satisfy conditions (24a) and (24b). Then, nodes  $(i, k)$  can be serviced by the same vehicle only if node  $i$  is visited first, i.e.  $S_{ik} = 1$ . Therefore, Eqs. (5a) and (5b) and (8a) and (8b) reduce themselves to simpler constraints (25) and (26), respectively.

$$\begin{aligned} C_k &\geq C_i + c_{ik}^v - M_C(2 - Y_{rv} - Y_{r'v}), \\ i &\in I_r, k \in I_{r'}(i < k), r, r' \in R, v \in V \end{aligned} \quad (25)$$

$$\begin{aligned} T_k &\geq T_i + st_k^v + t_{ik}^v - M_T(2 - Y_{rv} - Y_{r'v}), \\ i &\in I_r, k \in I_{r'}(i < k), r, r' \in R, v \in V \end{aligned} \quad (26)$$

In addition, Eqs. (14a) and (14b) and (15a) and (15b) turn into constraints (27) and (28):

$$\begin{aligned} L_k &\geq L_i + \alpha_{kr'} - M_L(2 - Y_{rv} - Y_{r'v}), \\ i &\in I_r, k \in I_{r'}(i < k), r, r' \in R, v \in V \end{aligned} \quad (27)$$

$$\begin{aligned} U_k &\geq U_i + \beta_{kr'} - M_L(2 - Y_{rv} - Y_{r'v}), \\ i &\in I_r, k \in I_{r'}(i < k), r, r' \in R, v \in V \end{aligned} \quad (28)$$

This rule can be viewed as the generalization of the one developed for Langevin, Desrochers, Desrosiers, Gelinas, and Soumis (1993) for the TSPTW. When a single-vehicle PDPTW problem includes  $n$  pairs of preordered nodes, then  $n$  sequencing variables  $S_{ik}$  and at most  $6n$  constraints can be eliminated.

#### 4. The MILP-based local search improvement strategy

##### 4.1. The neighborhood structure definition

Real-world  $m$ -PDPTW problems are commonly characterized by a high combinatorial complexity that easily exceeds the capabilities of current optimization codes. To overcome this limitation, effective modeling techniques and solution strategies are required. In this particular logistic problem, the solution strategy can take advantage of the geographical location of the nodes to be serviced in order to generate a set of representative sub-problems with manageable size. To this end, a compact neighborhood domain around the current solution is defined in such a way that the whole original problem can be effectively optimized through a rigorous local-search methodology.

In the context of the definition of the neighborhood structure, transportation requests can be classified into two types: fixed ( $R^F$ ) and mobile ( $R^M$ ) depending on whether they should stay on the current route or can be transferred to neighboring tours. Pickup and delivery nodes associated to a fixed request  $r$  are forced to remain on the current route because the chance of reducing the overall traveling costs by moving them to a neighboring tour is rather low. However, their relative locations on the current route  $v$  can be modified. Some simple criterion to estimate the likelihood of getting savings in traveling costs from moving request  $r$  to another vehicle  $v'$  is then required. Obviously, the set of fixed requests  $R^F$  may change from one iteration  $k$  to the next ( $k+1$ ) since the set of nodes on the route  $v$  generally varies. On the contrary, it is expected that the transfer of a mobile request  $r \in R^M$  to a closer route  $v'$  probably leads to a lower-cost solution. Every vehicle route  $v$  can have one or several neighboring tours  $v'$  but the set of candidate routes  $v' \in V_r^k$  for a particular mobile request  $r$  at iteration  $k$  includes just a few tours in addition to the current one. In short, the neighborhood structure will comprise a rather small set of feasible solutions that can be generated by: (a) transferring mobile requests  $r \in R^M$  from the current route  $v$  to some arbitrary position on a neighboring route  $v' \in V_r$  ( $v' \neq v$ ) and (b) reordering pickup/delivery nodes of fixed requests  $r \in R^F$  located on the same vehicle route  $v$ .

To classify the nodes on a route into fixed or mobile ones, an easy-to-compute criteria based on the widely known sweep heuristic has been developed (Gillet and Miller, 1974). Such a sweep heuristic groups the nodes based on their angular coordinates with respect to some line radiating from the central depot. As the line moves clockwise or counterclockwise, trips are constructed by allocating neighboring nodes with similar angular coordinates to the same vehicle while its capacity is not exceeded. In this way, it is generated a set of routes with an optimal topology pattern consisting of non-overlapping petal-shaped routes, usually observed on a wide range of routing problems. In a similar way, this paper defines a pair of characteristic angular distances  $\varphi_0$  and  $\varphi_1$ , with  $\varphi_1 < \varphi_0$ , to categorize a node as fixed or mobile. The model parameters  $(\varphi_0, \varphi_1)$  define a pair of cones surrounding each trip  $v$  (see Fig. 1). Let us assume that the whole vehicle fleet is housed in a central depot. Each characteristic cone for a trip  $v$  has its origin at the central depot, its

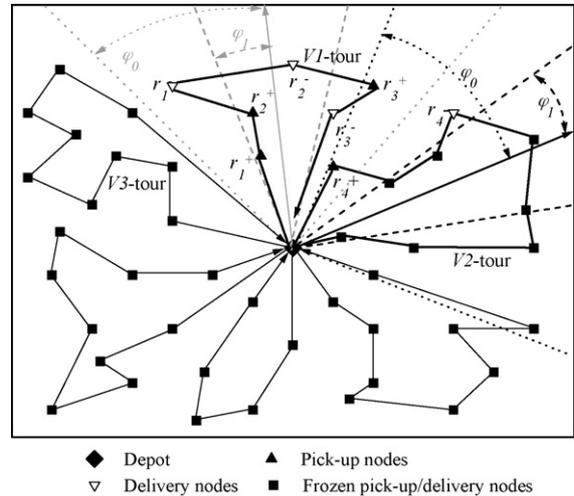


Fig. 1. Illustrative representation of model parameters  $(\varphi_0, \varphi_1)$  defining fixed and mobile requests.

axis is a ray going from the origin through the centre of gravity of the trip, and its two delimiting rays form an angle  $\varphi_0$  or  $\varphi_1$  with the trip axis. Values for the model parameters  $(\varphi_0, \varphi_1)$  are set by the user.

The  $\varphi_1$ -slim cone defines a narrow zone around the trip axis containing the nodes that, a priori, are likely to stay on the tour at the end of the improvement process; i.e. the set of fixed nodes  $I_v^F$  on trip  $v$ . In other words, a fixed node  $i \in I_v^F$  is one featuring an angle  $\theta_{iv} < \varphi_1$ , where  $\theta_{iv}$  is measured from the  $v$ th-trip axis to the ray connecting node  $i$  to the central depot (see Fig. 1). Any node  $i' \in I_v$  outside the  $\varphi_1$ -cone ( $\theta_{i'v} > \varphi_1$ ) will be regarded as a potential mobile node on trip  $v$  that may be transferred to neighboring trips. The angle  $\theta_{i'v}$  is determined from the polar coordinates of both the node  $i'$  and the centre of gravity of route  $v$  to which belongs at the starting solution. Such polar coordinates are computed from the Cartesian coordinates of both the node  $i'$  ( $x_{i'}, y_{i'}$ ) and the depot  $p \in P$  housing vehicle  $v$  ( $x_p, y_p$ ), respectively. Once a preliminary classification of nodes based on the  $\varphi_1$ -cone has been performed, the transport requests are categorized into fixed or mobile according to the condition of the related pickup and delivery nodes. Therefore, a fixed request is one whose pickup and delivery nodes (or just an arbitrary number  $k_1$  of them) have all been classified as fixed nodes. In turn, a mobile request means that at least one (or an arbitrary number  $k_2$  adopted by the user) of the associated pickup or delivery nodes is mobile. Here, it is worth remarking that the whole request must always be serviced by a single vehicle. Consequently, one or more mobile nodes in a request will force the other fixed nodes of the request to be re-categorized as mobile ones.

On the other hand, the  $\varphi_0$ -expanded cone defines a geographical area beyond the  $\varphi_1$ -cone that may contain nodes from neighboring routes  $v' \neq v$  (see Fig. 1). In this way, the  $\varphi_0$ -cone around trip  $v$  permits to identify the set of mobile requests from neighboring routes that can be transferred to trip  $v$ . They are requests with at least one (or an arbitrary number  $k_3$ ) mobile node lying inside the  $\varphi_0$ -cone around trip  $v$ . Therefore, a mobile request from a neighboring trip  $v'$  with a mobile node  $i'$  featuring an angle  $\theta_{i'v} \leq \varphi_0$  measured from the  $v$ th-trip axis to the ray con-

necting node  $i'$  to the central depot will feature the route  $v$  as one of the candidate trips to which it can be transferred, i.e.  $v \in V_{i'}^k$ . In short, the  $\varphi_0$ -cone aims to determine the candidate routes for mobile requests. It permits to determine the set of requests that may potentially be transferred to trip  $v$  on the next iteration  $k$  ( $R_v^k$ ). In addition to such candidate requests from nearby tours, the set  $R_v^k$  will also include fixed and moving requests visited by vehicle  $v$  at the previous iteration ( $R_v^{k-1}$ ). The angle  $\varphi_0$  must be sufficiently small to generate a problem formulation that can be solved to optimality at low CPU time but large enough to define a feasible space containing better solutions than the current one. An allocation variable  $Y_{r'v}$  is just defined for a mobile request  $r'$  only if  $r' \in R_v^k$ . From Fig. 1, it can be observed that request  $r_2$  is a fixed one on tour V1 because both the pickup  $r_2^+$  and the delivery node  $r_2^-$  lie inside the  $\varphi_1$ -cone around the axis of trip V1 while  $r_1$  and  $r_3$  are mobile ones because at least one of the nodes of such requests lies outside the  $\varphi_1$ -cone for trip V1. Then, requests  $r_1$  and  $r_3$  can potentially be transferred to neighboring routes. Since one of the nodes of request  $r_3$  lies inside the  $\varphi_0$ -cone around trip V2, then  $r_3$  can either be transferred to tour V2 ( $r_3 \in R_{V2}$ ) or remain on tour V1 ( $r_3 \in R_{V1}$ ) in the next iteration. Conversely,  $r_4$  is a mobile request currently on tour V2 that can be transferred to tour V1 because its pickup node lies inside the  $\varphi_0$ -cone around trip V1.

Finally, it is defined a third model parameter  $d^{\max}$  representing the maximum allowed Euclidean distance between a trip  $v$  and a mobile node  $i$  from a neighboring route. Despite  $\theta_{iv} \leq \varphi_0$ , every node  $i$  farther than  $d^{\max}$  from the center of gravity of tour  $v$  cannot be transferred to  $v$ . If all pickup and delivery nodes (or  $k_3$  of them) associated to a request  $r$  cannot be moved to tour  $v$ , then  $Y_{rv}$  can be deleted from the problem formulation. In all the examples solved in this paper, the value of  $d^{\max}$  was large enough to never exclude a potential request exchange between neighboring trips.

#### 4.2. The MILP-based optimization sub-problems

A pair of local optimization problems whose solution spaces jointly define the neighborhood structure to be explored is next presented. They are called sub-problems I and II, respectively. Both sub-problems arise from a proper simplification of the exact model previously presented and lead to mathematical formulations that are simpler and easier to solve. Sub-problem I is based on the underlying idea that a non-optimal solution can be improved by exchanging mobile requests among neighboring routes. For this reason, it just includes a subset of vehicle-request assignment variables, i.e.  $Y_{rv}$ ,  $\forall r \in R_v$ ,  $v \in V$ . In turn, sub-problem II only allows the reordering of all nodes on every tour by properly adjusting the sequencing variables. By sequentially solving both sub-problems, multiple re-allocations of requests to other tours and node reordering on every trip can occur at each major iteration and an improved feasible solution will hopefully be discovered. The choice of proper values for the parameters ( $\varphi_0$ ,  $\varphi_1$ ) defining the neighborhood domain usually leads the proposed methodology to obtain substantial improvements in the objective function with rather modest computational effort. However, the improvement search can be trapped in a local opti-

num. To restart the search towards a better feasible solution, a relaxed version of sub-problem I in which all transportation requests can be exchanged among neighboring routes is to be solved. Such a relaxation of sub-problem I, called sub-problem III, is derived by driving  $\varphi_1$  to zero. Since it requires a higher computational cost, sub-problem III will be solved only if the normal local optimization procedure fails to improve the current solution.

To start the search, an initial feasible solution must be available. In case that multiple depots are involved in the starting solution, the method assumes that every transport request  $r \in R$  and every vehicle  $v \in V$  has a pre-assigned depot  $p \in P$ . In other words, a request  $r'$  should be visited by a vehicle  $v'$  whose base is depot  $p'$  if  $r'$  has been assigned to  $p'$ , i.e. if  $r' \in R_{p'}$  then it will be serviced by  $v' \in V_{p'}$ . The sets  $R_p$  and  $V_p$  for any  $p \in P$  can be defined based on the starting solution. During the search, a request  $r \in R_p$  can only be transferred to other tours starting and ending at depot  $p$ , and the base  $p$  for any vehicle  $v \in V_p$  cannot be changed. However, it may occur that a vehicle  $v \in V_p$  in use at the starting solution is no longer needed at the set of improved routes for depot  $p$  because the related tour does not comprise any request. Since node exchanges between tours related to different depots are not allowed, the method should be repeatedly applied to improve the tours pre-assigned to any depot  $p \in P$ . Therefore, a single depot is assumed in the formulation of sub-problems I–III. The optimality of the improved solution provided by the algorithm is never guaranteed for either single-depot or multiple-depot PDPTW problems.

##### 4.2.1. Sub-problem I: exchanging requests among neighboring routes

In this sub-problem, the set of requests  $R$  is partitioned into two subsets involving mobile requests ( $R^M$ ) and fixed requests ( $R^F$ ), respectively, in such a way that  $R = R^F \cup R^M$ . The basic idea is to fix all the initial depot-vehicle assignments and iteratively revise a limited number of request-vehicle allocation and node sequencing decisions. For fixed requests  $R^F$ , assignment variables ( $Y_{rv}$ ) and sequencing variables  $S_{ii'}$  (with  $i \in I_r$ ,  $i' \in I_{r'}$ ,  $r, r' \in R^F$ ) are all frozen at their initial values. Conversely, there will be as many assignment variables  $Y_{rv}$  associated to a mobile request  $r \in R^M$  as the number of candidate tours  $V_r$  for request  $r$  including the one where is currently located. If transferred to another tour  $v \in V_r$ , the insertion points of nodes  $i \in I_r$  are optimally selected by defining the sequencing variables for the following pairs of nodes: (a) pickup/delivery nodes for request  $r$ ,  $i \in I_r$ , (b) nodes related to other requests  $r' \neq r$  currently located on the candidate tours for request  $r$ :  $i' \in I_{r'}$ ,  $r' \in R_v$ ,  $v \in V_r$  and (c) nodes related to other requests  $r'' \neq r$  that can be transferred to a candidate tour  $v \in V_r$  (i.e.  $V_r \cap V_{r''} \neq \emptyset$ ). This leads to different types of sequencing constraints depending on the pair of fixed/mobile nodes being considered. Some of them are redundant and can be deleted from the problem formulation while others turn to be much simpler, thus reducing the combinatorial nature of the problem. The resulting mathematical formulation for sub-problem I is presented in Appendix A.

#### 4.2.2. Sub-problem II: reordering nodes on every individual route

The definition of the neighborhood structure is completed by introducing sub-problem II where the reordering of nodes on every route is only permitted. In contrast to sub-problem I, no exchange of nodes among neighboring routes is allowed and the vehicle allocation variables  $Y_{rv}$  are deleted from the problem formulation. Therefore, sub-problem II is to be solved as many times as the number of tours in the solution provided by sub-problem I. Although sub-problem II is characterized by a much lower computational cost than sub-problem I, the re-optimization of individual routes may jointly generate significant improvements in the objective function. The mathematical formulation for sub-problem II is presented in Appendix B.

#### 4.2.3. Sub-problem III: simultaneous reordering and exchange of nodes

Since the search can be trapped in a local optimum, a larger neighborhood needs to be considered in order to move towards a better solution. To this end, every transport request is regarded as mobile and can be transferred to some neighboring route. This is achieved by making  $\varphi_1 = 0$  and adopting a finite  $\varphi_0$  to define the candidate tours for every request. Moreover, the reordering of nodes on every trip is also allowed. Therefore, the mathematical model for sub-problem III is similar to the exact formulation introduced in Section 3 except for: (i) a single depot is assumed and no vehicle-depot assignment variable is needed, and (ii) a smaller  $\varphi_0$  is adopted to decrease the number of candidate routes for each request, i.e.  $|V_r|$ . However, the use of an extended neighborhood will surely lead to a larger problem formulation. For this reason, sub-problem III will be solved only if sub-problems I and II both fail to improve the current solution.

#### 4.3. Spatial decomposition of large $m$ -PDPTW problems

Since the neighborhood structure accounts for solutions generated from the current solution by reordering nodes on each individual tour or relocating requests to neighboring trips, there is no sense in tackling the whole  $m$ -PDPTW problem at once. In a local search environment, each tour just exchanges nodes with a few routes closed to it and the attention should therefore be focused on a much smaller geographical area where such interacting trips are confined. In order to take advantage of such a problem feature, a rotating angular sector (RAS) is defined with origin at the central depot (DC) and delimiting rays emanating from the DC with angular coordinates  $\Omega_1$  and  $\Omega_2$ , respectively (see Fig. 2). The angle  $\Omega (= \Omega_2 - \Omega_1)$  between the extreme rays remains fixed as the RAS rotates. In order to sweep the whole service region, the RAS will turn around the DC by equally increasing the extreme ray angular coordinates  $\Omega_1$  and  $\Omega_2$  by a fixed quantity  $\Omega$ . In this way, the RAS will take a series of angular positions  $\{\Omega^{(1)}, \Omega^{(2)}, \dots\}$  before rotating  $2\pi$  and start a new turn. If some but not all nodes of a trip  $v$  are inside the RAS<sup>(m)</sup> at a particular position  $\Omega^{(m)} = 0.5 (\Omega_2^{(m)} + \Omega_1^{(m)})$  of the RAS-axis, then the procedure assumes that the whole trip  $v$  is contained in RAS<sup>(m)</sup>. In other words, a node  $i$  will pertain to

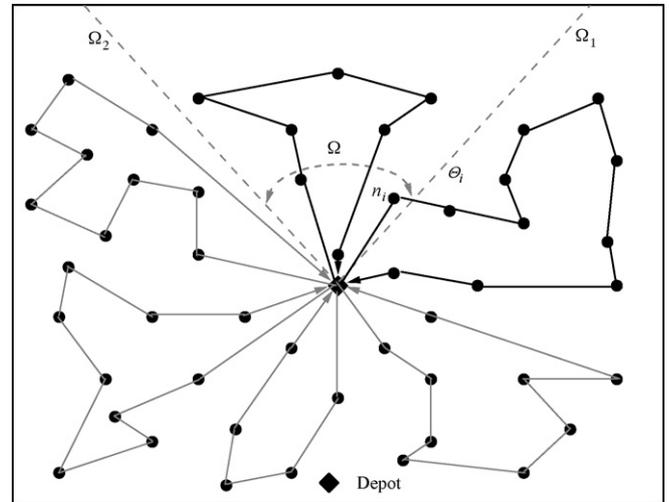


Fig. 2. The rotating angular sector (RAS) decomposing the service region into smaller zones.

RAS<sup>(m)</sup> if either (a) its angular coordinate  $\Theta_i$  is between  $\Omega_1^{(m)}$  and  $\Omega_2^{(m)}$  ( $\Omega_1^{(m)} \leq \Theta_i \leq \Omega_2^{(m)}$ ) or alternatively (b) the trip to which it currently belongs has at least a node  $i'$  satisfying the condition  $\Omega_1^{(m)} \leq \Theta_{i'} \leq \Omega_2^{(m)}$ . At any location  $\Omega^{(m)}$ , therefore, the RAS<sup>(m)</sup> will contain a limited number of complete tours together with the nodes located on all of them. Moreover, a trip  $v$  may belong to the rotating sector at two consecutive locations  $\Omega^{(m)}$  and  $\Omega^{(m+1)}$ . Initially,  $\Omega_1^{(1)}$  is set equal to zero. If  $N$  is the number of RAS-locations per turn, then  $N = (2\pi/\Omega)$ .

Each time sub-problems I and II for a particular location  $\Omega^{(m)}$  are solved, only routes inside the RAS<sup>(m)</sup> will be considered. Nodes and trips beyond RAS<sup>(m)</sup> are ignored. Therefore, the mathematical formulations for sub-problems I and II will change with  $\Omega^{(m)}$  as long as the set of nodes  $I^{(m)}$  and the set of tours  $V^{(m)}$  inside the RAS both depend on  $\Omega^{(m)}$ . At each RAS-location, sub-problems I and II will be repeatedly solved until the procedure converges to a local optimum (the normal mode). It may happen that no improvement at all has been achieved through the normal mode after sweeping the  $N$  locations, i.e. the whole service area. In order to avoid getting stuck on a local optimum, sub-problem III will be activated (the perturbation or mixed mode), if necessary, just on the next turn to move forward towards a feasible/infeasible solution with a better value of the objective function. If the perturbation move is successful, then the normal mode is applied again. The procedure is repeated until the normal mode becomes trapped on a local optimum and the perturbation mode fails to get an improved solution. When this happens, the procedure is stopped and the best set of routes is given by the current incumbent solution.

In short, the solution algorithm described in Fig. 3 will have an outer loop that iterates over the RAS-angular location  $\Omega^{(m)}$  and an inner loop repeatedly executed to find the best routing of the vehicles servicing the geographical area covered by RAS<sup>(m)</sup>. Within the inner loop, either the normal or the perturbation mode is applied depending on whether or not the normal mode has yielded a reduction on the total transportation cost during the previous RAS-rotation. When the procedure completes another

- 
- **Initialization Phase**
    - a) Find an initial solution to start the improvement process.
    - b) Choose values for each procedure parameter ( $q_0, q_1, \Omega, d_{max}$ ) to be applied under the normal mode and the perturbation mode, respectively. Compute:  $N = 2\pi / \Omega$ .
    - c) Set the values for the parameters  $T$  and  $\epsilon$ .
  - **Improvement Phase**  
 Set:  $m = 1$ ,  $MODE = normal$   
 For the angular sector or service zone  $m$ :
    - a) Find the set of candidate tours for each request  $V_r$  defining the solution-space to be explored
      - Identify the tours lying on the angular sector ( $m$ ).
      - Identify the fixed requests  $R^F$  and mobile requests  $R^M$  on each vehicle  $v \in V$ .
      - Identify the candidate tours  $V_r$  for every mobile request  $R^M$ .
    - b) Solve the proper MVDPTW improvement formulation  
 If  $MODE = normal$ :
      - Solve Subproblem I
      - Eliminate any tour  $v$  featuring  $R_v = \emptyset$ .
      - Update the sets  $R_v, R^M$  and  $V_r$  for  $r \in R^M$ .
      - Solve Subproblem II.
      - Go to Step (c).
 If  $MODE = perturbation$ :
      - Solve Subproblem III.
      - Eliminate any tour  $v$  featuring  $R_v = \emptyset$ .
      - Update the sets  $R_v, R^M$  and  $V_r$  for  $r \in R^M$ .
    - c) Compute the cost savings ( $\Delta z^{(m)}$ ) achieved on Step (b).
      - If  $(\Delta z^{(m)} > \epsilon)$  then Repeat Step (b)
      - If  $(\Delta z^{(m)} \leq \epsilon$  and  $m < N)$  then  $m = m+1$  and Go to Step (b)
  - **Termination Phase**  
 Compute the overall cost decrease ( $\Delta Z$ ) during the last turn of the rotating angular sector.
    - If  $(\Delta Z > \epsilon)$  then Restart the Improvement Phase.
    - If  $(\Delta Z \leq \epsilon$  and  $MODE = normal)$  then Set  $MODE = perturbation$  and restart the Improvement Phase.
 Otherwise, STOP. The algorithm has converged to a stationary point.
- 

Fig. 3. The PDPTW neighborhood improvement algorithm.

turn  $h$ , the new incumbent solution is obtained by considering the best vehicle routes found at each of the  $N$  locations of the rotating sector. If a trip  $v$  belongs to a pair of consecutive RAS-locations  $m$  and  $m+1$ , the latter one will define its structure at the new best solution, i.e. the subset  $I_v$  and the sequence of nodes on trip  $v$ . Convergence of the procedure to a local or global optimum is checked out by comparing the total cost of the new incumbent solution after completing turn  $h$  with that of the old one at the end of turn  $(h-1)$ . If the normal and the perturbation modes both fail to provide a better solution or the improvement on the objective function is less than a small positive scalar  $\epsilon$ , then the procedure must be stopped. In other words, the method has converged if no improvement at all has been obtained on the last two turns of the rotating angular sector.

In multi-depot  $m$ -PDPTW problems, the improvement method assumes that requests and vehicles have their pre-assigned depots (see Section 4.1) and exchanges of requests among tours linked to different depots are not allowed. Therefore, depots and their related tours can be considered one by one. In other words, the solution algorithm described in Fig. 3 should be applied as many times as the number of depots involved in the starting solution.

## 5. Results and discussion

The proposed  $m$ -PDPTW exact and improvement approaches have both been tested by solving a set of PDPTW problem instances introduced by Li and Lim (2003a). Such examples

were generated from known solutions to VRPTW Solomon benchmark problems (Solomon, 1987) by randomly pairing customer locations on the same tour (Li & Lim, 2003a). All Solomon's VRPTW problems originally feature 100 real nodes and a single central depot, but some dummy nodes, if necessary, may be added for pairing purpose. In this manner, multi-vehicle PDPTW benchmark problems with nearly 50 customer requests, each one involving a single pickup node and a single delivery node, were developed. Similarly to Solomon's VRPTW benchmark problems, the new ones proposed by Li and Lim (2003a) have been grouped into three different categories: LC, LR and LRC. The data set for every category comprises several problem instances with the same geographical node distribution, a central depot, multiple vehicles with a similar load capacity and different time-window width distributions. Problems of class LC have *clustered* customers whose associated time windows have been generated based on known solutions. In LR-class problems, customer locations are *randomly* generated over a square while LRC-class problems result from a combination of clustered and randomized customer distributions. Problems of each class are further classified into two types called "1" and "2". Type-1 PDPTW problems are characterized by narrow time windows and low-capacity vehicles, while type-2 problems involve wider time windows and vehicles with a larger capacity. Therefore, solutions to type-2 PDPTW problems feature fewer and longer tours. For instance, the example LR-102 is a LR1-problem involving randomly distributed nodes and low-capacity vehicles, i.e. a R-class problem of type-1. The last two digits 02

Table 1  
Optimal solutions for some PDPTW benchmark problems using the exact MILP approach

Example	Nodes	Vehicles	Optimal value	CPU time (s)	Binary variables	Continuous variables	Constraints
LR-101	24	8	599.1	5.83	156	112	6,265
	30	10	700.2	50.07	245	140	12,418
	36	11	833.6	12.29	312	166	19,702
	50	14	1138.4	527.42	1108	228	52,098
LR-102	24	8	543.9	8.81	200	112	6,446
	30	9	673.5	24.98	295	138	11,376
	36	11	824.6	2476.20	415	166	20,181
	40	11	860.2	5821.40	563	182	25,334
LR-103	24	5	432.2	120.81	228	106	4,129
	30	6	616.5	519.94	306	132	7,602
	36	7	782.8	1715.69	392	158	12,658
LR-104	24	4	493.3	9.25	268	104	3,411
	30	5	534.4	711.98	393	130	6,586
	36	6	681.2*	7200*	524	156	11,250
LR-201	24	2	557.1	22.83	69	100	1,124
	30	2	645.8	287.23	101	124	1,718
	36	2	760.7	715.91	136	162	2,429
LC-201	24	2	257.6	0.91	36	100	1,024
	30	2	269.1	1.47	46	124	1,552
	36	2	290.0	1.77	56	162	2,188
LC-102	24	3	249.8	7.83	108	102	2,067
	30	4	313.5	22.69	161	128	4,529
	36	4	348.9	121.8	282	130	8,581
LR-103m	36	7	759.8	88.74	387	158	12,628

\* Best solution after 7200 s of CPU time.

permit to characterize such a particular LR1-problem instance. In short, six classes of benchmark problems were generated: LC1, LC2, LR1, LR2, LRC1 and LRC2. Problem data include the number of available vehicles, Euclidean distances among customer sites and normalized vehicle speeds such that traveling times and Euclidean distances are numerically identical. The reported solutions to benchmark problems proposed by Li and Lim (2003a) all assume that time-window constraints should be strictly satisfied and the total distance traveled by the vehicle fleet is the objective function to be minimized. Some LC and LR problem instances were solved by applying both the proposed exact MILP problem formulation and the local search improvement algorithm using ILOG OPL Studio (Script mode) on a 2.8 MHz 1 GB RAM Pentium IV PC.

Optimal solutions for different instances of problems LR-101/102/103/104, LR-201 and LC-102/201, all involving transport requests with a single pickup and a single delivery site, found through the exact MILP mathematical model are all summarized in Table 1. Such examples were derived by just considering the first 24–50 entries on the list of nodes of the original Li and Lim's 100-node benchmark problems. In this manner, problem instances with at most 25 requests and 50 nodes have been created. For narrow time-windows, the behavior of the exact approach greatly improves due to the better performance of the exact elimination rules and, consequently, the CPU time increases almost linearly with the problem size. This fact leads to the conclusion that the proposed formulation is capable of efficiently finding the optimal set of routes for large  $m$ -PDPTW

problems with narrow time-windows. For instance, the version of problem LR-101 with 50 nodes and 14 vehicles was solved to optimality in just 527 s. Table 2 and Fig. 4 both describe the optimal solution for the example LR-101 with 50 nodes. It can be noted that the minimum travel distance for LR-101 (50 nodes) is achieved at the expense of a total vehicle waiting time as large as 889.9. Fig. 5 shows a drawing of the optimal petal-shaped routes found for the 36-node version of problem instances LC-

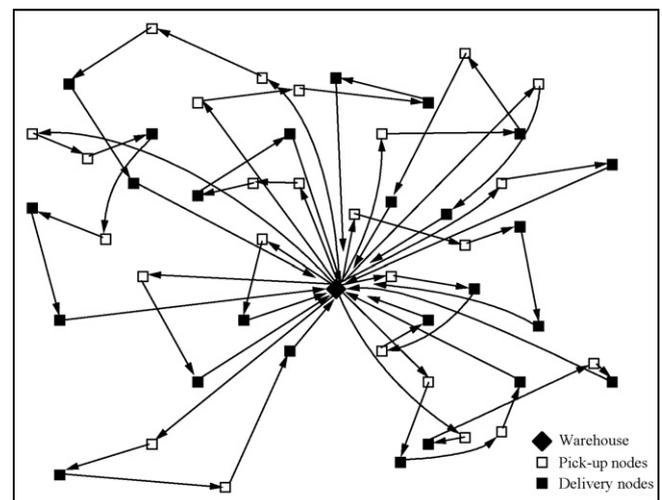


Fig. 4. Optimal solution found for the 50-node version of benchmark problem LR-101.

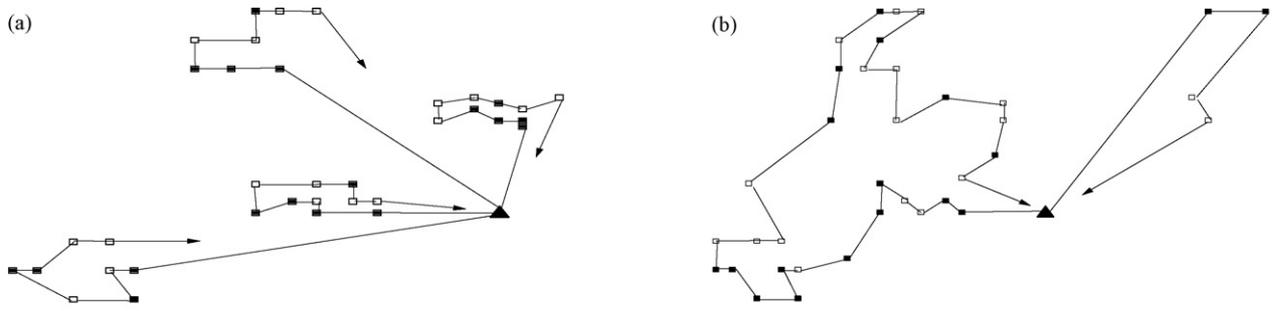
Table 2  
Optimal solution for the 50-node version of problem LR-101

Vehicle	Pickup/delivery node	Waiting time	Initial service time	Final service time	Service duration	Pickup/delivery load	Vehicle load	Vehicle routing time	Traveled distance																																																																																																																																																																																																																																																																																																																																																								
V1	P83	22.90	44.00	54.00	10.00	11	11	175.20	60.30																																																																																																																																																																																																																																																																																																																																																								
	D37	72.00 94.90	144.00	154.00	10.00	−11	0			V2	P33	12.20	37.00	47.00	10.00	11	11	173.10	73.30	D34	67.60 79.80	127.00	137.00	10.00	−11	0	V3	P75	27.60	69.00	79.00	10.00	18	18	225.50	91.20	D22	13.90	97.00	107.00	10.00	−18	0	P55	16.80	146.00	156.00	10.00	2	2	D25	22.40 80.70	182.00	192.00	10.00	−2	0	V4	P36	0.00	41.40	51.40	10.00	5	5	207.40	132.40	P47	0.00	58.60	68.60	10.00	27	32	D19	0.00	76.70	86.70	10.00	−5	27	P08	0.60	105.00	115.00	10.00	9	36	D46	2.60	127.00	137.00	10.00	−27	9	D17	11.80 15.00	167.00	177.00	10.00	−9	0	V5	P31	32.50	50.00	60.00	10.00	27	27	169.50	68.40	P88	9.00	74.00	84.00	10.00	9	36	D07	0.00	90.30	100.30	10.00	−9	27	D10	19.60 61.10	134.00	144.00	10.00	−27	0	V6	P52	40.70	52.00	62.00	10.00	9	9	120.20	35.70	D06	23.80 64.50	99.00	109.00	10.00	−9	0	V7	P69	37.80	50.00	60.00	10.00	6	6	182.80	70.80	P76	10.00	83.00	93.00	10.00	13	19	D03	0.00	123.90	133.90	10.00	−6	13	D54	0.00 47.80	150.00	160.00	10.00	−13	0	V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60	D41	11.70	97.00	107.00	10.00	−11	0	P56	10.90	130.00	140.00	10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20									
V2	P33	12.20	37.00	47.00	10.00	11	11	173.10	73.30																																																																																																																																																																																																																																																																																																																																																								
	D34	67.60 79.80	127.00	137.00	10.00	−11	0			V3	P75	27.60	69.00	79.00	10.00	18	18	225.50	91.20	D22	13.90	97.00	107.00	10.00	−18	0		P55	16.80	146.00	156.00	10.00	2	2			D25	22.40 80.70	182.00	192.00	10.00	−2	0	V4	P36	0.00	41.40	51.40	10.00	5	5	207.40	132.40	P47	0.00	58.60	68.60		10.00	27	32	D19	0.00	76.70	86.70			10.00	−5	27	P08	0.60	105.00	115.00	10.00	9	36	D46	2.60	127.00	137.00	10.00	−27	9	D17	11.80 15.00	167.00	177.00	10.00	−9	0	V5	P31	32.50	50.00	60.00	10.00	27	27	169.50	68.40	P88		9.00	74.00	84.00	10.00	9	36	D07			0.00	90.30	100.30	10.00	−9	27	D10	19.60 61.10	134.00	144.00	10.00	−27	0	V6	P52	40.70	52.00	62.00	10.00	9	9	120.20	35.70	D06	23.80 64.50	99.00	109.00	10.00	−9	0	V7	P69	37.80	50.00	60.00	10.00	6	6		182.80	70.80	P76	10.00	83.00	93.00	10.00			13	19	D03	0.00	123.90	133.90	10.00	−6	13	D54	0.00 47.80	150.00	160.00	10.00	−13	0	V8	P21	44.00	62.00	72.00		10.00	11	11	184.00	76.60	D41	11.70			97.00	107.00	10.00	−11	0	P56	10.90	130.00	140.00	10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63		9.10	44.00	54.00	10.00	10	10	212.80			107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9		0	V10	P28	32.70	39.00	49.00	10.00			16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00		142.00	10.00	−9	0	V11	P11	33.50			67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3		D32	4.20 61.40	151.00	161.00	10.00	−3	0			V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20
V3	P75	27.60	69.00	79.00	10.00	18	18	225.50	91.20																																																																																																																																																																																																																																																																																																																																																								
	D22	13.90	97.00	107.00	10.00	−18	0																																																																																																																																																																																																																																																																																																																																																										
	P55	16.80	146.00	156.00	10.00	2	2																																																																																																																																																																																																																																																																																																																																																										
	D25	22.40 80.70	182.00	192.00	10.00	−2	0			V4	P36	0.00	41.40	51.40	10.00	5	5	207.40	132.40	P47	0.00	58.60	68.60	10.00	27	32	D19	0.00	76.70	86.70	10.00	−5	27	P08	0.60	105.00	115.00	10.00	9	36	D46	2.60	127.00		137.00	10.00	−27	9	D17	11.80 15.00	167.00			177.00	10.00	−9	0	V5	P31	32.50	50.00	60.00	10.00	27	27	169.50	68.40	P88	9.00	74.00	84.00	10.00	9	36	D07	0.00	90.30	100.30	10.00	−9	27	D10	19.60 61.10	134.00	144.00	10.00	−27	0	V6	P52	40.70	52.00	62.00	10.00	9	9	120.20	35.70	D06	23.80 64.50	99.00	109.00	10.00	−9	0	V7	P69	37.80	50.00	60.00	10.00	6	6	182.80	70.80	P76	10.00	83.00	93.00	10.00	13	19	D03	0.00	123.90	133.90	10.00	−6	13	D54	0.00 47.80	150.00	160.00	10.00	−13	0	V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60	D41	11.70	97.00	107.00	10.00	−11	0	P56	10.90	130.00	140.00	10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																						
V4	P36	0.00	41.40	51.40	10.00	5	5	207.40	132.40																																																																																																																																																																																																																																																																																																																																																								
	P47	0.00	58.60	68.60	10.00	27	32																																																																																																																																																																																																																																																																																																																																																										
	D19	0.00	76.70	86.70	10.00	−5	27																																																																																																																																																																																																																																																																																																																																																										
	P08	0.60	105.00	115.00	10.00	9	36																																																																																																																																																																																																																																																																																																																																																										
	D46	2.60	127.00	137.00	10.00	−27	9																																																																																																																																																																																																																																																																																																																																																										
	D17	11.80 15.00	167.00	177.00	10.00	−9	0			V5	P31	32.50	50.00	60.00	10.00	27	27	169.50	68.40	P88	9.00	74.00	84.00	10.00	9	36	D07	0.00	90.30	100.30	10.00	−9	27	D10	19.60 61.10	134.00	144.00	10.00	−27	0	V6	P52	40.70	52.00	62.00	10.00	9	9	120.20	35.70	D06	23.80 64.50	99.00	109.00	10.00	−9	0	V7	P69	37.80	50.00	60.00	10.00	6	6	182.80	70.80	P76	10.00	83.00	93.00	10.00	13	19	D03	0.00	123.90	133.90	10.00	−6	13	D54	0.00 47.80	150.00	160.00	10.00	−13	0	V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60	D41	11.70	97.00	107.00	10.00	−11	0	P56	10.90	130.00	140.00	10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																						
V5	P31	32.50	50.00	60.00	10.00	27	27	169.50	68.40																																																																																																																																																																																																																																																																																																																																																								
	P88	9.00	74.00	84.00	10.00	9	36																																																																																																																																																																																																																																																																																																																																																										
	D07	0.00	90.30	100.30	10.00	−9	27																																																																																																																																																																																																																																																																																																																																																										
	D10	19.60 61.10	134.00	144.00	10.00	−27	0			V6	P52	40.70	52.00	62.00	10.00	9	9	120.20	35.70	D06	23.80 64.50	99.00	109.00	10.00	−9	0	V7	P69	37.80	50.00	60.00	10.00	6	6	182.80	70.80	P76	10.00	83.00	93.00	10.00	13	19	D03	0.00	123.90	133.90	10.00	−6	13	D54	0.00 47.80	150.00	160.00	10.00	−13	0	V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60	D41	11.70	97.00	107.00	10.00	−11	0	P56	10.90	130.00	140.00	10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																					
V6	P52	40.70	52.00	62.00	10.00	9	9	120.20	35.70																																																																																																																																																																																																																																																																																																																																																								
	D06	23.80 64.50	99.00	109.00	10.00	−9	0			V7	P69	37.80	50.00	60.00	10.00	6	6	182.80	70.80	P76	10.00	83.00	93.00	10.00	13	19		D03	0.00	123.90	133.90	10.00	−6	13			D54	0.00 47.80	150.00	160.00	10.00	−13	0	V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60	D41	11.70	97.00	107.00		10.00	−11	0	P56	10.90	130.00	140.00			10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64		4.60	73.00	83.00	10.00	9	19	D49			12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16		153.20	47.90	D12	4.80	63.00	73.00	10.00			−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00		10.00	12	12	195.00	103.60	P90	16.80			95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30		45.50	71.00	81.00	10.00	21	21	196.20			95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																				
V7	P69	37.80	50.00	60.00	10.00	6	6	182.80	70.80																																																																																																																																																																																																																																																																																																																																																								
	P76	10.00	83.00	93.00	10.00	13	19																																																																																																																																																																																																																																																																																																																																																										
	D03	0.00	123.90	133.90	10.00	−6	13																																																																																																																																																																																																																																																																																																																																																										
	D54	0.00 47.80	150.00	160.00	10.00	−13	0			V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60	D41	11.70	97.00	107.00	10.00	−11	0	P56	10.90	130.00	140.00	10.00	6	6	D04	10.80 77.40	159.00	169.00	10.00	−6	0	V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																					
V8	P21	44.00	62.00	72.00	10.00	11	11	184.00	76.60																																																																																																																																																																																																																																																																																																																																																								
	D41	11.70	97.00	107.00	10.00	−11	0																																																																																																																																																																																																																																																																																																																																																										
	P56	10.90	130.00	140.00	10.00	6	6																																																																																																																																																																																																																																																																																																																																																										
	D04	10.80 77.40	159.00	169.00	10.00	−6	0			V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30	P64	4.60	73.00	83.00	10.00	9	19	D49	12.30	108.00	118.00	10.00	−10	9	D48	39.50 65.50	175.00	185.00	10.00	−9	0	V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																																																				
V9	P63	9.10	44.00	54.00	10.00	10	10	212.80	107.30																																																																																																																																																																																																																																																																																																																																																								
	P64	4.60	73.00	83.00	10.00	9	19																																																																																																																																																																																																																																																																																																																																																										
	D49	12.30	108.00	118.00	10.00	−10	9																																																																																																																																																																																																																																																																																																																																																										
	D48	39.50 65.50	175.00	185.00	10.00	−9	0			V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90	D12	4.80	63.00	73.00	10.00	−16	0	P40	7.90	95.00	105.00	10.00	9	9	D26	19.90 65.30	132.00	142.00	10.00	−9	0	V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																																																																																			
V10	P28	32.70	39.00	49.00	10.00	16	16	153.20	47.90																																																																																																																																																																																																																																																																																																																																																								
	D12	4.80	63.00	73.00	10.00	−16	0																																																																																																																																																																																																																																																																																																																																																										
	P40	7.90	95.00	105.00	10.00	9	9																																																																																																																																																																																																																																																																																																																																																										
	D26	19.90 65.30	132.00	142.00	10.00	−9	0			V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60	P90	16.80	95.00	105.00	10.00	3	15	D20	6.90	126.00	136.00	10.00	−12	3	D32	4.20 61.40	151.00	161.00	10.00	−3	0	V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																																																																																																																		
V11	P11	33.50	67.00	77.00	10.00	12	12	195.00	103.60																																																																																																																																																																																																																																																																																																																																																								
	P90	16.80	95.00	105.00	10.00	3	15																																																																																																																																																																																																																																																																																																																																																										
	D20	6.90	126.00	136.00	10.00	−12	3																																																																																																																																																																																																																																																																																																																																																										
	D32	4.20 61.40	151.00	161.00	10.00	−3	0			V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30	D09	1.00	97.00	107.00	10.00	−21	0	P66	14.40	135.70	145.70	10.00	25	25	D01	0.00 60.90	171.00	181.00	10.00	−25	0	V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																																																																																																																																																	
V12	P30	45.50	71.00	81.00	10.00	21	21	196.20	95.30																																																																																																																																																																																																																																																																																																																																																								
	D09	1.00	97.00	107.00	10.00	−21	0																																																																																																																																																																																																																																																																																																																																																										
	P66	14.40	135.70	145.70	10.00	25	25																																																																																																																																																																																																																																																																																																																																																										
	D01	0.00 60.90	171.00	181.00	10.00	−25	0			V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																																																																																																																																																																																
V13	P14	0.00	32.00	42.00	10.00	20	20																																																																																																																																																																																																																																																																																																																																																										

Table 2 (Continued)

Vehicle	Pickup/delivery node	Waiting time	Initial service time	Final service time	Service duration	Pickup/delivery load	Vehicle load	Vehicle routing time	Traveled distance
V14	D38	29.80	83.00	93.00	10.00	-20	0	190.20	95.60
	P43	20.90	132.00	142.00	10.00	7	7		
	D13	3.90	169.00	179.00	10.00	-7	0		
		54.60							
V14	P71	37.30	77.00	87.00	10.00	15	15	151.00	80.00
	D50	23.70	124.00	134.00	10.00	-15	0		
		61.00							

Total traveled distance: 1138.4. Total waiting time: 889.9. Total service time: 500.0.



Routes [Routing time; Travelled distance]		Routes [Routing time; Travelled distance]	
D-13-17-18-19-15-16-14-12-D [ $t_v = 833.8$ ; $c_v = 113.8$ ]		D-97-92-7-4-D [ $t_v = 1536.6$ ; $c_v = 94.9$ ]	
D-20-24-25-27-29-30-26-23-22-21-D [ $t_v = 1022.6$ ; $c_v = 113.8$ ]		D-20-22-24-27-30-29-6-32-33-31-35-37-38-39-36-34-28-23-18-19-16-14-12-15-17-13-25-9-11-10-8-21-D	
D-5-3-7-8-10-11-9-6-4-2-D [ $t_v = 981.9$ ; $c_v = 59.4$ ]		[ $t_v = 3244.7$ ; $c_v = 162.7$ ]	
D-32-33-31-35-38-37-36-34-D [ $t_v = 882.2$ ; $c_v = 92.7$ ]		Total travelled distance	290.0
Total travelled distance	348.9	Total travelled distance	4781.3
Total travelling time	3720.5	Total travelling time	1251.3
Total idle time	131.6	Total idle time	

Fig. 5. Optimal solutions for (a) problem LC-102 (36 nodes) and (b) problem LC-201 (36 nodes).

102 and LC-201. In the figure, a triangle represents a warehouse, solid squares stand for pickup sites and hollow squares denote delivery sites. In contrast, the CPU time required to solve different instances of problems LR-102, LR-103 and LR-104 to

optimality significantly increases as the number of nodes climbs from 24 to 36 nodes, especially for problem LR-104. For such less constrained examples, the exact elimination rules no longer eliminate an important number of constraints and variables. To

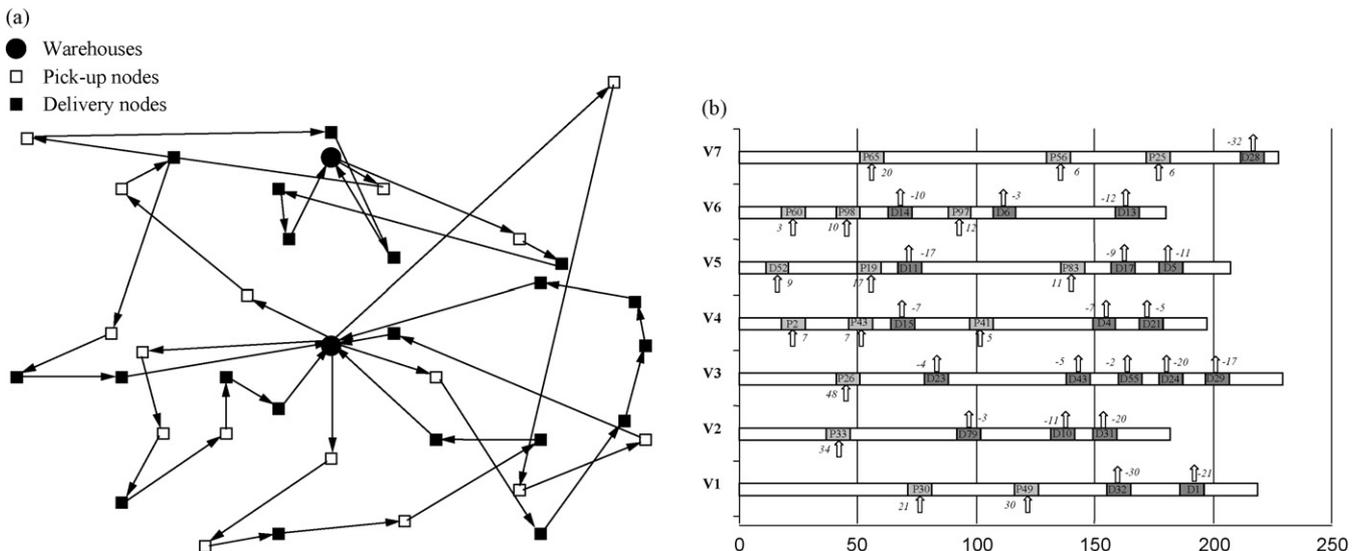


Fig. 6. (a) A sketch of the optimal routes for the modified LR-103 problem with 36 nodes. (b) Sequence of pickup/delivery tasks on every tour at the optimal solution of example LR-103m (36 nodes).

illustrate the application of the exact approach to  $m$ -PDPTW examples involving customer requests with multiple pickup and multiple delivery points, a modified version of the LR-103 problem instance, called LR-103m, featuring 36 nodes, 2 depots and 7 vehicles, has been tackled and the results appear on the last row of Table 1. A drawing of the optimal routes and a Gantt chart describing the sequence of pickup/delivery tasks on each tour for the example LR-103m are given in Fig. 6(a and b), respectively.

### 5.1. Validating the proposed PDPTW local search improvement algorithm

In order to assess the performance of the proposed PDPTW improvement algorithm in terms of the solution quality and the computational cost, several validation examples with 36–50 nodes (some of them already included in Table 1) were solved by applying both the exact and the improvement approaches. The best solutions found for each example through the two methodologies are compared in Table 3. To start the PDPTW improvement approach, an initial solution should be available. It was developed by first arbitrarily assigning a single request to each available vehicle and then successively inserting every non-assigned request into the nearest incomplete route. Let us  $CG_r$  denote the center of gravity of the request  $r$ ,  $CG_v$  represent the center of gravity of the incomplete tour traveled by vehicle  $v$  and  $\theta_{rv}$  stand for the angular distance between the ray  $H_r$ , connecting the central depot D and  $CG_r$ , and the ray  $H_v$  connecting D and  $CG_v$ . A non-assigned request  $r$  is allocated to the incomplete route  $v$  featuring the lowest  $\theta_{rv}$ . The allocation of requests to a particular vehicle  $v$  should be stopped when a new assignment leads to violations of either capacity or time-window constraints. The construction of the initial solution ends when every transportation request has been assigned to some route. The starting point for each validation example and the related total travel distance are also given in Table 3. In addition, this table includes the adopted values for the search parameters ( $\varphi_0$ ,  $\varphi_1$  and  $\Omega$ ), the total travel distance for the best solution found through this approach, the CPU solution time and the percentage optimality gap for every validation example.

The adopted value for the parameter  $\Omega$  mostly ranges from  $\pi/6$  to  $\pi/8$  to spatially decompose the servicing area and therefore the problem formulation into a set of 12–16 smaller sub-problems to be sequentially solved to develop an improved solution. The angular parameter  $\varphi_0 > \Omega$  is aimed at defining a wider servicing area for each sub-problem to allow the exchange of customer requests among routes lying on neighboring sectors. As a result, it increases the number of requests considered in the sub-problem formulation by including additional ones located in the vicinity. In turn, the parameter  $\varphi_1 < \Omega$  defines a freezing area for each sub-problem so that every request lying on it must remain on the current route in the next iteration  $k$ . Therefore, the vehicle assignment variables associated to such frozen requests are deleted from the sub-problem formulation to be solved at iteration  $k$ . If the search becomes trapped in a non-optimal solution, it is restarted by reducing  $\varphi_1$  to zero, thus freeing every frozen request. At the same time,  $\varphi_0$  is decreased to prevent from a significant increase in the model size. When this perturbation mode fails to provide a better solution in every sector, the solution procedure is stopped.

Validation examples of types LC1, LC2, LR1 and LR2 were tackled. Similarly to Li and Lim (2003a), the problem goal was to minimize the total traveled distance subject to hard time-window constraints. It can be seen that the proposed PDPTW solution algorithm can efficiently find the optimal vehicle tours for most of the validation examples (see Table 3). Though quite similar for some examples, the solution time shows on average a sizable decrease with regards to the exact method, especially for LR1 validation problems. In some cases, the CPU time is reduced by a factor of 40 and the optimal solution is still found. Moreover, low CPU times are needed to solve validation examples LC-102 (36 nodes) and LC-201 (36 nodes) involving clustered customers to optimality. Similarly to VRPTW, it is more difficult to find the optimum of LR-problem instances of type “2”. Nonetheless, the largest CPU time of 326 s was required to solve the 50-node version of problem LR-101 involving 14 different tours. In cases where sub-optimal solutions were found, the sub-optimality gap was rather small and below the 7.5% threshold. From Table 4, it follows that the optimal solution to the 36-node version of exam-

Table 3  
Best solutions found for some validation examples through the proposed improvement algorithm

Example	Optimal solution <sup>a</sup>				Best solution found									
	Nodes	Vehicles	Distance	CPU	Distance		Search parameters							
					Initial	Final	Gap	CPU	Normal			Perturbation		
									$\varphi_1$	$\varphi_0$	$\Omega$	$\varphi_0$	$\Omega$	
LC-102	36	4	348.9	121.8	424.3	348.9	0.0%	108	0.4	0.8	$\pi/4$	0.6	$\pi/4$	
LC-201	36	2	290.0	1.8	318.0	290.0	0.0%	5	0.4	1.2	$\pi/2$	0.8	$\pi/2$	
LR-101	36	11	833.6	12.3	850.3	833.6	0.0%	23	0.5	1.5	$\pi/5$	1.0	$\pi/5$	
LR-101	50	14	1138.4	527.4	1502.2	1138.4	0.0%	326	0.5	1.5	$\pi/5$	1.0	$\pi/5$	
LR-102	36	11	856.8	2476.2	889.1	856.8	0.0%	52	0.5	1.5	$\pi/5$	1.0	$\pi/5$	
LR-103	36	7	782.8	1715.7	830.2	797.5	1.9%	12	0.5	1.5	$\pi/5$	1.0	$\pi/5$	
LR-104	30	5	534.4	712.0	626.3	574.3	7.5%	26	0.5	1.5	$\pi/5$	1.0	$\pi/5$	
LR-104	36	6	681.2 <sup>b</sup>	7200 <sup>b</sup>	765.0	605.1	–	210	0.5	1.5	$\pi/5$	1.0	$\pi/5$	
LR-103m	36	7	759.8	89	831.6	772.0	1.6%	222	0.3	0.9	$\pi/8$	0.6	$\pi/8$	

<sup>a</sup> Provided by the exact MILP formulation.

<sup>b</sup> Best solution found after 7200 s of CPU time.

Table 4  
Best solutions found for several 100-node benchmark problems through the improvement algorithm

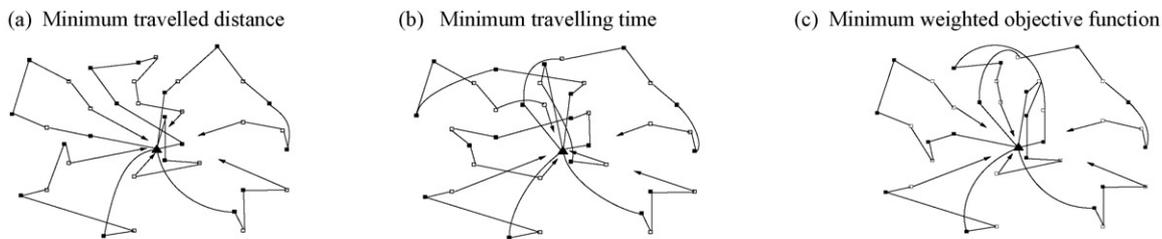
Example	Best known solution <sup>a</sup>			Best solution found								
	Vehicles	Distance	Vehicles	Distance		Search parameters						
				Initial	Final	Gap	CPU	Normal		Perturbation		
								$\varphi_1$	$\varphi_0$	$\Omega$	$\varphi_0$	$\Omega$
LC-102	10	828.9	10	878.3	851.0	2.7%	360	0.90	0.30	$\pi/8$	0.30	$\pi/12$
LC-201	3	591.6	3	813.9	605.4	2.3%	132	0.90	0.30	$\pi/4$	0.45	$\pi/4$
LR-103	13	1292.7	14	1535.6	1383.5	6.9%	396	0.30	0.90	$\pi/6$	0.60	$\pi/8$
LR-104	9	1013.4	10	1194.7	1082.8	6.8%	1311	0.50	1.50	$\pi/6$	1.00	$\pi/6$
LR-105	14	1377.1	15	1452.6	1391.7	1.1%	214	0.50	1.50	$\pi/6$	1.00	$\pi/6$
LR-106	12	1252.6	14	1309.6	1299.9	3.8%	1096	0.40	1.20	$\pi/8$	0.80	$\pi/8$
LR-107	10	1111.3	14	1237.6	1167.8	5.1%	423	0.40	1.20	$\pi/8$	0.80	$\pi/8$
LR-108	9	969.0	10	1069.7	1007.0	3.9%	1504	0.20	0.60	$\pi/10$	0.40	$\pi/10$
LR-202	3	1197.0	4	1495.2	1395.8	16.6%	2174	0.50	1.50	$\pi/3$	1.00	$\pi/3$

<sup>a</sup> Bent and Van Hentenryck (2006).

ple LR-104 could not be discovered through the exact approach after 7200 s. The total travel distance for the non-optimal solution found amounts to 681.2. The new PDPTW improvement technique was able to find a better solution that reduces the total travel distance by 12.5% in just 210 s. The graphical representation of the improved routes for the 36-node version of example LR-104 is given in Fig. 7(a) where the total travel time is also reported.

To analyze the effect of the adopted objective function on the solution morphology, example LR-104 (36 nodes) was successively solved through the exact approach by using the following problem goals: (a) minimization of the overall travelled distance ( $\sum_v c_v$ ); (b) minimization of the total travelling time ( $\sum_v t_v$ ); (c) minimization of a weighted objective function ( $1/2 \sum_v t_v + 1/2 \sum_v c_v$ ). The optimal solutions for cases (b) and (c) are summarized in Fig. 7(b and c), including the node sequence on every tour, the total travel distance, the total travel time and the total idle time. By choosing the total travel time as the objective function, the idle time drops from 241.6 to 103.5 and at the same time the total travel distance rises from 605.1 to 660. The weighted objective function provides

a good trade-off between the two operational goals as shown in Fig. 7(c). The total travel distance is kept at 631.7 while the idle time is limited to 143.5. It can be seen that the change in the objective function does not significantly modify the somewhat distorted “petal” shaped pattern of the optimal routes. Consequently, PDPTW algorithmic approaches that basically assume such a kind of route configuration to explore the solution space can still be applied even if other objective functions different from the total travel distance are adopted. It is worth noting that the use of the total travel distance frequently leads to non-realistic solutions with large vehicle idle times. In other words, vehicles arrived early at some customer locations and should wait for the opening of the time window. However, the choice of the total traveling time as the objective function may substantially increase the traveled distance. To get a good compromise between the two problem goals, it is better using a mixed objective function like the one applied in case (c) of example LR-104 (36 nodes). The improvement algorithm can also handle the three objective functions and its computational performance does not seem to be affected by the change in the problem objective. In each case, the same set of search parameters were adopted



Routes [Routing time; Travelled distance]		Routes [Routing time; Travelled distance]		Routes [Routing time; Travelled distance]	
D-69-53-26-13-D [ $t_v = 180.2$ ; $c_v = 63.3$ ]		D-90-10-53-26-D [ $t_v = 134.3$ ; $c_v = 88.4$ ]		D-70-30-69-53-26-13-D [ $t_v = 180.2$ ; $c_v = 90.5$ ]	
D-43-15-86-16-83-5-D [ $t_v = 191.6$ ; $c_v = 122.1$ ]		D-43-15-86-16-D [ $t_v = 155.6$ ; $c_v = 106.1$ ]		D-43-15-86-16-D [ $t_v = 155.6$ ; $c_v = 106.1$ ]	
D-70-30-66-9-78-24-29-3-D [ $t_v = 213.5$ ; $c_v = 122.7$ ]		D-88-32-66-9-78-24-29-3-D [ $t_v = 213.8$ ; $c_v = 133.8$ ]		D-88-32-66-9-78-24-29-3-D [ $t_v = 213.8$ ; $c_v = 133.8$ ]	
D-56-23-4-25-D [ $t_v = 215.5$ ; $c_v = 95.0$ ]		D-56-23-4-25-D [ $t_v = 215.5$ ; $c_v = 95.0$ ]		D-56-23-4-25-D [ $t_v = 215.5$ ; $c_v = 95.0$ ]	
D-18-8-46-49-19-7-D [ $t_v = 219.2$ ; $c_v = 100.4$ ]		D-28-1-69-18-8-83-5-13-D [ $t_v = 180.2$ ; $c_v = 92.9$ ]		D-18-83-5-8-46-49-19-7-D [ $t_v = 219.2$ ; $c_v = 117.7$ ]	
D-28-88-11-90-33-19-31-1-D [ $t_v = 186.7$ ; $c_v = 101.5$ ]		D-70-30-11-46-49-19-7-31-D [ $t_v = 224.1$ ; $c_v = 143.8$ ]		D-28-1-11-90-10-31-D [ $t_v = 150.9$ ; $c_v = 88.6$ ]	
Total travelled distance	605.1	Total travelled distance	660.0	Total travelled distance	631.7
Total travelling time	1206.7	Total travelling time	1123.5	Total travelling time	1135.2
Total idle time	241.6	Total idle time	103.5	Total idle time	143.5

Fig. 7. Optimal solutions to the 36-node version of problem LR-104 using different objective functions.

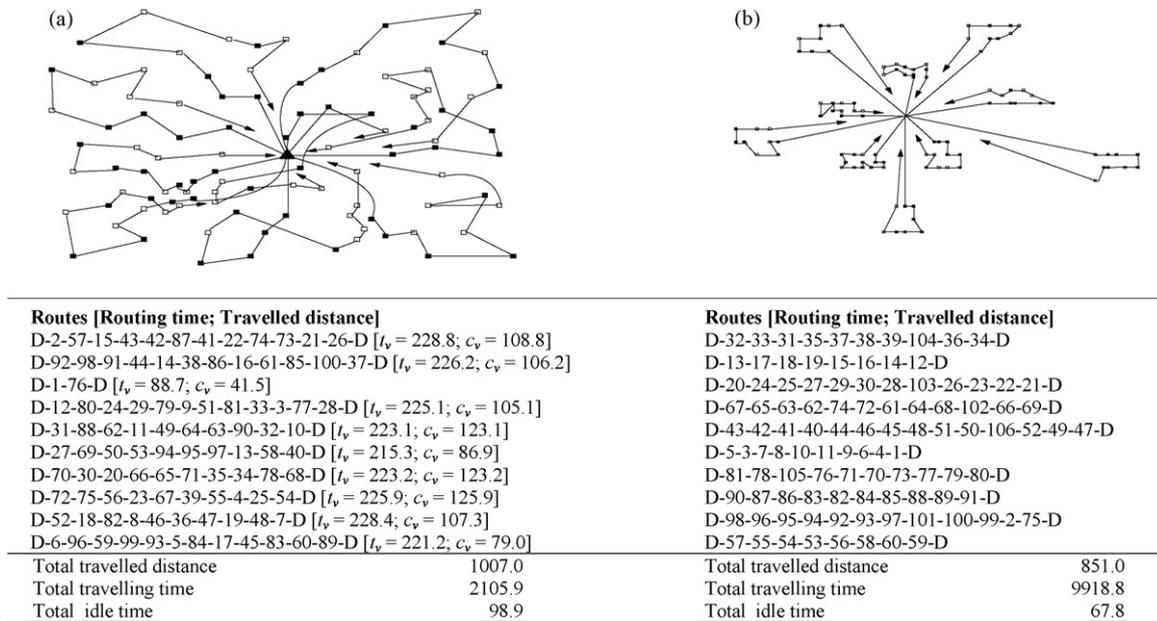


Fig. 8. Best solutions found for 100-node benchmark problems: (a) LR-108, (b) LC-102.

Table 5

Depot coordinates for the multi-depot PDPTW examples

D2-problem instances: Two depots
Depot coordinates: P1 { $X=35, Y=35$ }, P2 { $X=55, Y=55$ }
D3-problem instances: Three depots
Depot coordinates: P1 { $X=35, Y=35$ }, P2 { $X=55, Y=55$ }, P3 { $X=35, Y=75$ }

(see Table 3). The near-optimal solutions were obtained for the three cases (a–c) of example LR-104 (36 nodes) in 210, 327 and 286 s, respectively. Furthermore, the related sub-optimality levels were 1.3, 5.1 and 2.4%, respectively, with regards to the optimal solution found through the MILP exact approach.

### 5.2. Solving some Li and Lim’s 100-node benchmark problems

Results to some original Li and Lim’s problem instances involving nearly 50 requests and 100 pickup/delivery nodes are all presented in Table 4. They are compared with the best available solutions reported in the literature also shown in Table 4 (Ropke & Pisinger, 2006). The adopted values for the model

parameters are relatively similar to those selected for the validation examples. As before, near-optimal solutions for LC and LR-benchmark problems were found in a rather short CPU time (see Table 4). Moreover, the optimality gap in terms of the objective value for most 100-node benchmark problems remains below 7% and the CPU time averages 846 s. In each case, the best solution provided by the local search improvement approach uses one or two vehicles more than the one specified at the original example so as to strictly satisfy hard time-window constraints. Except for problem LR-202, we can claim that optimality gaps are relatively small and good feasible solutions are discovered at reasonable computational cost. Fig. 8 shows a graphical representation of the best solutions found for benchmark problems LR-108 and LC-102. It also includes: (i) the node sequence and the total distance and travel time for every tour and (ii) the total distance, travel time and idle time for the whole set of routes. Though additional pairing constraints are to be considered, the vehicle tours for such PDPTW benchmark problems still feature the typical petal-shaped pattern. Moreover, the total idle time has been reduced to a small fraction of the total traveling time. The large sub-optimality of the solution discovered for the benchmark problem LR-202 can be explained by the

Table 6

Best solutions found for multi-depot PDPTW examples through the improvement algorithm

Example	Plants	Vehicles	Vehicles assignment			Traveled distance		CPU	Search parameters				
			P1	P2	P3	Initial	Final		Normal			Perturbation	
									$\varphi_1$	$\varphi_0$	$\Omega$	$\varphi_0$	$\Omega$
D2-LR-103	2	15	9	6	–	1435.1	1295.5	909	0.6	1.8	$\pi/6$	1.2	$\pi/6$
D2-LR-105	2	16	8	8	–	1710.0	1568.5	644	0.6	1.8	$\pi/2$	1.2	$\pi/2$
D2-LR-202	2	5	4	1	–	1735.9	1612.3	613	0.4	1.2	$\pi/2$	0.8	$\pi/2$
D3-LR-103	3	15	8	2	4	1612.2	1387.7	3667	0.6	1.8	$\pi/3$	1.2	$\pi/3$
D3-LR-105	3	15	6	5	4	1548.3	1476.9	1002	0.6	1.8	$\pi/2$	1.2	$\pi/2$
D3-LR-202	3	5	3	1	1	1746.4	1649.6	617	0.4	1.2	$\pi/2$	0.8	$\pi/2$

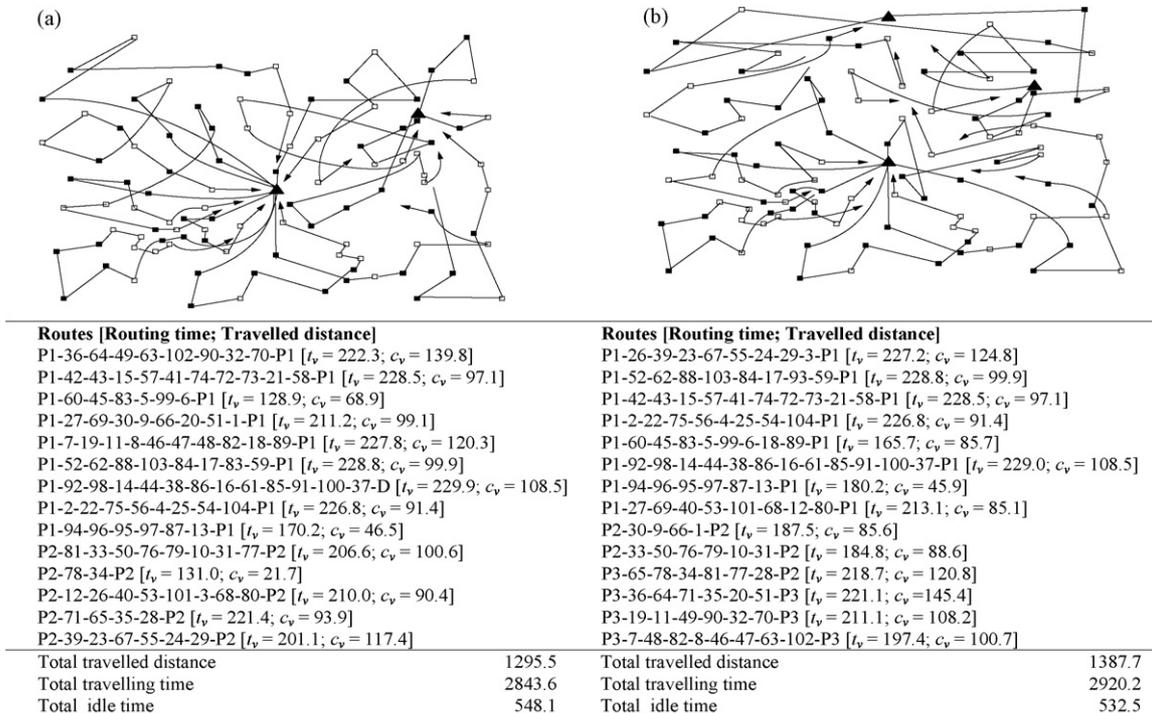


Fig. 9. Best solutions found for multi-depot  $m$ -PDPTW problems D2-LR-103 and D3-LR-103.

fact that problems of type “2” involves longer optimal tours and many requests per vehicle. In such cases, we will need to adopt larger angular parameters  $\Omega$  and  $\varphi_0$  to get a better performance of the proposed approach.

### 5.3. Solving new examples involving multiple depots

To test the  $m$ -PDPTW improvement algorithm performance when tackling multi-depot problems, some of Li and Lim’s benchmark problems have been modified by incorporating one or two new depots. Therefore, the modified problem instances will include two or three depots but the vehicle data remain unchanged. They will belong to a new category of benchmark problems identified with the prefix Dn, where n is the number of depots. Depot coordinates are given in Table 5 while the solutions to six modified multi-depot problems of classes LR1 and LR2 are reported in Table 6. The required CPU times were roughly similar to the ones needed to solve the corresponding single-depot problems instances. Allocation of vehicles to depots is made during the construction of the initial solution, and such decisions are “frozen” throughout the improvement process. Fig. 9(a and b) depicts the graphical representations of the best vehicle routes found for examples D2-LR-103 and D3-LR-103, respectively. Since nodes located far away from a depot can hardly be visited by vehicles using that depot as their base, it becomes necessary to make use of a limiting Euclidean distance  $d^{\max}$  to define the set of requests to be serviced by vehicles coming from that terminal. A desirable option to reduce the objective function is the re-assignment of bases to vehicles during the improvement process. However, the proposed local search improvement approach assumes that vehicles have pre-assigned bases and their re-allocation to other depots are

forbidden. This is a usual practice in supply-scenarios where vehicles are pre-allocated to bases at the tactical level and such decisions cannot be changed at the operation level.

## 6. Conclusions

The transportation costs for logistic activities heavily depend on operational decisions like the routing and scheduling of a vehicle fleet providing pickup and delivery services to a large number of customers. A new exact precedence-based MILP formulation for the multiple vehicle pickup and delivery problem with time windows ( $m$ -PDPTW) has been presented. The proposed approach allows to separately treating depot-vehicle allocation, vehicle-request assignment and node sequencing decisions through different sets of two-index binary variables. In this way, a two-index mathematical model capable of dealing with heterogeneous vehicles, multiple depots, many-to-many requests and pure pickup/delivery nodes has been derived. The model utilizes a continuous representation of the time domain, the cost domain and the load domain. Different types of objective functions like vehicle fixed costs, distance-based traveling cost, time-based traveling cost, customer inconvenience or a weighted combination of them can be alternatively used. To tackle medium-size problems, several variable and constraint elimination rules disallowing non-compatible requests to stay on the same tour have also been presented. Several examples involving up to 50 nodes and 14 vehicles were solved to optimality in quite acceptable CPU times.

However, the exact approach can efficiently handle  $m$ -PDPTW problem instances with at most 50 nodes. To deal with large-scale problems, a novel MILP improvement framework for

time-window-constrained pickup and delivery problems involving a heterogeneous fleet and multiple depots has also been developed. It is a local search improvement approach that fully explores a rather large neighborhood around the current solution so as to provide a better set of routes in an efficient manner. To achieve this goal, the approach relies on two key elements: a spatial decomposition scheme and a new MILP mathematical representation for the *m*-PDPTW improvement problem. A significant reduction in the problem size is obtained by properly adopting a single parameter  $\Omega$  to divide the geographical area to be serviced into *N* smaller angular sectors. By definition, routes which are partially or completely inside a particular sector belong to it. In addition, a pair of parameters  $(\varphi_0, \varphi_1)$  permits to define the set of feasible improvement moves on every zone, i.e. the neighborhood to be explored. The proposed MILP improvement problem formulation allows to efficiently explore a rather large solution space around the starting point on every zone by accounting for all kinds of improvement moves. To get MILP-models of moderate size, two types of improvement moves are considered one at a time: (1) request exchanges between neighboring tours and (2) node relocations on every individual tour. In this way, the *m*-PDPTW improvement problem for each zone can be decomposed into a pair of lower-size MILP sub-problems that are sequentially solved, using as initial point the best solution found for the other sub-problem (the normal mode). However, this sequential search scheme can be trapped on a local optimum. If so, the complete MILP problem formulation simultaneously considering both types of improvement moves should be solved (the perturbation or mixed-mode). Several benchmark problems involving 100 nodes, and in some cases multiple depots, were successfully tackled. Moreover, the algorithm usually converges to a low-cost solution at reasonable computational cost.

**Acknowledgments**

Financial support received from FONCYT-ANPCyT under Grant PICT 11-14717, from CONICET under Grant PIP-5729 and from Universidad Nacional del Litoral under CAI+D 003-13 is fully appreciated.

**Appendix A. The mathematical formulation for sub-problem I**

*A.1. Model constraints*

- Re-allocation of mobile requests

$$\sum_{v \in V_r} Y_{rv} = 1 \quad \forall r \in R^M$$

- Lower bound on the traveling cost/arrival time at the first visited node

$$C_i \geq c_{pi}^v Y_{rv}, \quad T_i \geq t_{pi}^v Y_{rv} \quad \forall i \in I_r, r \in R_v^M, v \in V_p, p \in P$$

$$C_i \geq c_{pi}^v, \quad T_i \geq t_{pi}^v \quad \forall i \in I_r, r \in R_v^F, v \in V_p, p \in P$$

- Precedence relationships to determine vehicle traveling costs/arrival times at nodes  $(i, i')$ 
  - For every pair of nodes  $(i, i')$  belonging to mobile requests  $(r, r') \in R_v^M$

$$\left\{ \begin{array}{l} C_{i'} \geq C_i + c_{ii'}^v - M_C(1 - S_{ii'}) - M_C(2 - Y_{rv} - Y_{r'v}) \\ C_i \geq C_{i'} + c_{i'i}^v - M_C S_{ii'} - M_C(2 - Y_{rv} - Y_{r'v}) \\ T_{i'} \geq T_i + st_i + t_{ii'}^v - M_T(1 - S_{ii'}) - M_T(2 - Y_{rv} - Y_{r'v}) \\ T_i \geq T_{i'} + st_{i'} + t_{i'i}^v - M_T S_{ii'} - M_T(2 - Y_{rv} - Y_{r'v}) \end{array} \right.$$

$$\forall i \in I_r, i' \in I_{r'}(i < i'), (r, r') \in R_v^M, v \in V$$

- For every pair of nodes  $(i, i')$  belonging to a pair of mixed requests  $(r \in R_v^F, r' \in R_v^M)$

$$\left\{ \begin{array}{l} C_{i'} \geq C_i + c_{ii'}^v - M_C(1 - S_{ii'}) - M_C(1 - Y_{r'v}) \\ C_i \geq C_{i'} + c_{i'i}^v - M_C S_{ii'} - M_C(1 - Y_{r'v}) \\ T_{i'} \geq T_i + st_i + t_{ii'}^v - M_T(1 - S_{ii'}) - M_T(1 - Y_{r'v}) \\ T_i \geq T_{i'} + st_{i'} + t_{i'i}^v - M_T S_{ii'} - M_T(1 - Y_{r'v}) \end{array} \right.$$

$$\forall i \in I_r, i' \in I_{r'}(i < i'), r \in R_v^F, r' \in R_v^M, v \in V$$

- For every pair of nodes  $(i, i')$  belonging to fixed requests  $(r, r') \in R_v^F$  (assuming  $S_{ii'} = 1$ )

$$\left\{ \begin{array}{l} C_{i'} \geq C_i + c_{ii'}^v \\ T_{i'} \geq T_i + st_i + t_{ii'}^v \end{array} \right.$$

$$\forall i \in I_r, i' \in I_{r'}(i < i'), (r, r') \in R_v^F, v \in V : i \text{ precedes } i'$$

- Overall travelling time/cost for vehicle *v*

$$\left\{ \begin{array}{l} OC_v \geq C_i + c_{ip}^v - M_C(1 - Y_{rv}) \\ OT_v \geq T_i + st_i + t_{ip}^v - M_T(1 - Y_{rv}) \end{array} \right.$$

$$\forall i \in I_r, r \in R_v^M, v \in V_p, p \in P$$

$$\left\{ \begin{array}{l} OC_v \geq C_i + c_{ip}^v \\ OT_v \geq T_i + st_i + t_{ip}^v \end{array} \right. \quad \forall i \in I_r, r \in R_v^F, v \in V_p, p \in P$$

- Early/late arrival time at node *i* and over-duration of routing time of vehicle *v*

$$\left\{ \begin{array}{l} E_i \geq a_i - T_i \\ B_i \geq T_i - b_i \end{array} \right. \quad \forall i \in I_r, r \in R$$

$$OD_v \geq OT_v - tv_v^{\max} \quad \forall v \in V$$

- Minimum loaded/unloaded cargo at node *i*

$$L_i \geq \sum_{r' \in R_v^F} q_{r'}^0 + \sum_{r' \in R_v^M} q_{r'}^0 Y_{r'v} + \alpha_{ir} - M_L(1 - Y_{rv}) \quad \forall i \in I_r, r \in R_v^M, v \in V$$

$$L_i \geq \sum_{r' \in R_v^F} q_{r'}^0 + \sum_{r' \in R_v^M} q_{r'}^0 Y_{r'v} + \alpha_{ir} \quad \forall i \in I_r, r \in R_v^F, v \in V$$

$$U_i \geq \beta_{ir} \quad \forall i \in I_r, r \in R$$

- Precedence relationships to determine the accumulated loaded/unloaded cargo at nodes  $(i, i')$

- For every pair of nodes  $(i, i')$  belonging to mobile requests  $r, r' \in R_v^M$

$$\left\{ \begin{array}{l} L_{i'} \geq L_i + \alpha_{i'r'} - M_L(1 - S_{ii'}) - M_L(2 - Y_{rv} - Y_{r'v}) \\ L_i \geq L_{i'} + \alpha_{ir} - M_L S_{ii'} - M_L(2 - Y_{rv} - Y_{r'v}) \\ U_{i'} \geq U_i + \beta_{i'r'} - M_L(1 - S_{ii'}) - M_L(2 - Y_{rv} - Y_{r'v}) \\ U_i \geq U_{i'} + \beta_{ir} - M_L S_{ii'} - M_L(2 - Y_{rv} - Y_{r'v}) \end{array} \right\}$$

$$\forall i \in I_r, i' \in I_{r'} (i < i'), (r, r') \in R_v^M, v \in V$$

- For every pair of nodes belonging to a pair of mixed requests  $(r \in R_v^F, r' \in R_v^M)$

$$\left\{ \begin{array}{l} L_{i'} \geq L_i + \alpha_{i'r'} - M_L(1 - S_{ii'}) - M_L(1 - Y_{r'v}) \\ L_i \geq L_{i'} + \alpha_{ir} - M_L S_{ii'} - M_L(1 - Y_{r'v}) \\ U_{i'} \geq U_i + \beta_{i'r'} - M_L(1 - S_{ii'}) - M_L(1 - Y_{r'v}) \\ U_i \geq U_{i'} + \beta_{ir} - M_L S_{ii'} - M_L(1 - Y_{r'v}) \end{array} \right\}$$

$$\forall i \in I_r, i' \in I_{r'} (i < i'), r \in R_v^F, r' \in R_v^M, v \in V$$

- For every pair of nodes  $(i, i')$  belonging to fixed requests  $r, r' \in R_v^F$  (assuming  $S_{ii'} = 1$ )

$$\left\{ \begin{array}{l} L_{i'} \geq L_i + \alpha_{i'r'} \\ U_{i'} \geq U_i + \beta_{i'r'} \end{array} \right\}$$

$$\forall i \in I_r, i' \in I_{r'}, (r, r') \in R_v^F, v \in V : i \text{ precedes } i'$$

- Bounds on the net cargo  $(L_i - U_i)$  loaded/unloaded at every node  $i$

$$L_i - U_i \leq q_v + Q(1 - Y_{rv}) \quad \forall i \in I_r, r \in R_v^M, v \in V \quad (Q = \max(q_v))$$

$$L_i - U_i \leq q_v \quad \forall i \in I_r, r \in R_v^F, v \in V$$

$$L_i - U_i \geq 0 \quad \forall i \in I_r, r \in R, v \in V$$

- Upper bounds on the cargo loaded/unloaded at every node  $i$

$$L_i - \sum_{r' \in R_v^F} \left( q_{r'}^0 + \sum_{i' \in I_{r'}^+} \alpha_{i'r'} \right) - \sum_{r' \in R_v^M} \left( q_{r'}^0 + \sum_{i' \in I_{r'}^+} \alpha_{i'r'} \right) Y_{r'v} \leq M_L(1 - Y_{rv}) \quad \forall i \in I_r, r \in R_v^M, v \in V$$

$$L_i - \sum_{r' \in R_v^F} \left( q_{r'}^0 + \sum_{i' \in I_{r'}^+} \alpha_{i'r'} \right) - \sum_{r' \in R_v^M} \left( q_{r'}^0 + \sum_{i' \in I_{r'}^+} \alpha_{i'r'} \right) Y_{r'v} \leq 0 \quad \forall i \in I_r, r \in R_v^F, v \in V$$

$$U_i - \sum_{r' \in R_v^F} \sum_{i' \in I_{r'}^-} \beta_{i'r'} - \sum_{r' \in R_v^M} \sum_{i' \in I_{r'}^-} \beta_{i'r'} Y_{r'v} \leq M_L(1 - Y_{rv}) \quad \forall i \in I_r, r \in R_v^M, v \in V$$

$$U_i - \sum_{r' \in R_v^F} \sum_{i' \in I_{r'}^-} \beta_{i'r'} - \sum_{r' \in R_v^M} \sum_{i' \in I_{r'}^-} \beta_{i'r'} Y_{r'v} \leq 0 \quad \forall i \in I_r, r \in R_v^F, v \in V$$

- Objective function

$$\min \sum_{v \in V} (OC_v + \Gamma OT_v + \Lambda OD_v) + \sum_{i \in I} (\lambda E_i + \omega L_i)$$

## Appendix B. The mathematical formulation for sub-problem II

Sub-problem II has to be applied to every individual tour  $v \in V$  provided by sub-problem I. This is so because every request should stay on the starting route. No exchange of nodes between neighboring tours is allowed. To derive the formulation of sub-problem II, it is also assumed that every vehicle  $v$  has its pre-assigned depot  $p$ , i.e. the one housing vehicle  $v$  at the starting solution. Moreover, the set  $R_v^F$  comprises the fixed requests serviced by vehicle  $v$  in the optimal solution of sub-problem I.

### B.1. Model constraints

- Lower bound on the traveling cost/arrival time at the first visited node

$$C_i \geq c_{pi}^v, \quad T_i \geq t_{pi}^v \quad \forall i \in I_r, r \in R_v^F, p \in P$$

- Precedence relationships to determine vehicle traveling costs/arrival times at nodes  $(i, i')$

$$\left\{ \begin{array}{l} C_{i'} \geq C_i + c_{ii'}^v - M_C(1 - S_{ii'}) \\ C_i \geq C_{i'} + c_{i'i}^v - M_C S_{ii'} \\ T_{i'} \geq T_i + st_i + t_{ii'}^v - M_T(1 - S_{ii'}) \\ T_i \geq T_{i'} + st_{i'} + t_{i'i}^v - M_T S_{ii'} \end{array} \right\}$$

$$\forall i \in I_r, i' \in I_{r'}, (r, r') \in R_v^F$$

- Overall travelling time/cost for vehicle  $v$

$$\left\{ \begin{array}{l} OC_v \geq C_i + c_{ip}^v \\ OT_v \geq T_i + st_i + t_{ip}^v \end{array} \right\} \quad \forall i \in I_r, r \in R_v^F$$

- Early/late arrival time at node  $i$  and over-duration of routing time of vehicle  $v$

$$\begin{cases} E_i \geq a_i - T_i \\ B_i \geq T_i - b_i \end{cases} \quad \forall i \in I_r, \quad r \in R$$

$$OD_v \geq OT_v - tv_v^{\max} \quad \forall v \in V$$

- Minimum loaded/unloaded cargo at node  $i$

$$L_i \geq \sum_{r' \in R_v^F} q_{r'}^0 + \alpha_{i r'} \quad \forall i \in I_r, \quad r \in R_v^F$$

$$U_i \geq \beta_{i r} \quad \forall i \in I_r, \quad r \in R$$

- Precedence relationships to determine the accumulated loaded/unloaded cargo at nodes  $(i, i')$

$$\begin{cases} L_{i'} \geq L_i + \alpha_{i' r'} - M_L(1 - S_{i' r'}) \\ L_i \geq L_{i'} + \alpha_{i r} - M_L S_{i' r'} \\ U_{i'} \geq U_i + \beta_{i' r'} - M_L(1 - S_{i' r'}) \\ U_i \geq U_{i'} + \beta_{i r} - M_L S_{i' r'} \end{cases}$$

$$\forall i \in I_r, \quad i' \in I_{r'}, \quad (r, r') \in R_v^F$$

- Bounds on the net cargo  $(L_i - U_i)$  loaded/unloaded at every node  $i$ .

$$L_i - U_i \leq q_v \quad \forall i \in I_r, \quad r \in R_v^F$$

$$L_i - U_i \geq 0 \quad \forall i \in I_r, \quad r \in R, \quad v \in V$$

- Upper bounds on the cargo loaded/unloaded at every node  $i$

$$L_i - \sum_{r' \in R_v^F} \left( q_{r'}^0 + \sum_{i' \in I_{r'}^+} \alpha_{i' r'} \right) \leq 0 \quad \forall i \in I_r, \quad r \in R_v^F$$

$$U_i - \sum_{r' \in R_v^F} \sum_{i' \in I_{r'}^-} \beta_{i' r'} \leq 0 \quad \forall i \in I_r, \quad r \in R_v^F$$

- Objective function

$$\min(OC_v + \Gamma OT_v + \Lambda OD_v) + \sum_{r \in R_v} \sum_{i \in I_r} (\lambda E_i + \omega L_i)$$

## References

- Altinel, I., & Öncan, T. (2005). A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *Journal of the Operational Research Society*, 56, 954–961.
- Bent, R., & Van Hentenryck, P. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers and Operations Research*, 33(4), 875–893.
- Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computers & Operations Research*, 10(2), 62–212.
- Cordeau, J. F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54, 573–586.
- Cyrus, J. (1988). *The vehicle scheduling problem: Models, complexity and algorithms*. Ph.D. Thesis. Technical University of Nova Scotia, Department of Engineering, Halifax, NS.
- Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M., & Soumis, F. (2002). VRP with pickup and delivery. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem, SIAM monographs on discrete mathematics and applications* (pp. 225–242). Philadelphia: SIAM.
- Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for vehicle routing problem with time windows. *Operations Research*, 40, 342–354.
- Desrosiers, J., Dumas, Y., Solomon, M., & Soumis, F. (1995). Time constrained routing and scheduling. In M. Ball, et al. (Eds.), *Handbook in OR & MS* (pp. 35–139). Elsevier Science.
- Dondo, R., Méndez, C. A., & Cerdá, J. (2003). An optimal approach to the multiple-depot heterogeneous vehicle routing problem with time window and capacity constraints. *Latin American Applied Research*, 33(2), 129–134.
- Dumas, Y., Desrosiers, J., & Soumis, F. (1989). *Large scale multi-vehicle dial-a-ride systems*. (Montreal Report G-89-30) Ecole des Hautes Commerciales.
- Dumas, Y., Desrosiers, J., & Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54, 7–22.
- Fisher, M. (1995). Vehicle routing. In M. Ball, et al. (Eds.), *Handbook in OR & MS* (pp. 1–33). Elsevier Science.
- Gillet, B., & Miller, L. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, 22, 340–349.
- Ioannou, G., Kritikos, M., & Prastacos, G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of the Operational Research Society*, 52, 523–537.
- Jaw, J., Odon, A., Psaraftis, H., & Wilson, N. (1986). A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research*, 20b, 243–257.
- Langevin, A. M., Desrochers, M., Desrosiers, J., Gelinas, S., & Soumis, F. (1993). A two-commodity flow formulation for the traveling salesman and makespan problems with time windows. *Networks*, 23, 631–640.
- Li, H., & Lim, A. (2003a). A meta-heuristic for the pickup and delivery problem with time windows. *Journal on Artificial Intelligent Tools*, 12(2), 173–186.
- Li, H., & Lim, A. (2003b). Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research*, 150(1), 115–127.
- Lu, Q., & Dessouky, M. M. (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science*, 38, 503–514.
- Lu, Q., & Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with hard time windows. *European Journal of Operational Research*, 175, 672–687.
- Méndez, C. A., Bonfill, A., Espuña, A., & Puigjaner, L. (2006). A rigorous approach to coordinate production and transport scheduling in a multi-site system. *Computer-Aided Chemical Engineering*, 21B, 2171–2176. ISBN:0-444-52970-5.
- Mitrovic-Minic, S. (1998). *Pickup and delivery problem with time windows: A survey*. (Technical Report 1998-12). Canada Research Chair in Distribution Management. CMPT.
- Nanry, W. P., & Wesley Barnes, J. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B*, 34, 107–121.
- Potvin, J., & Rousseau, J. (1995). An exchange heuristic for routing problems with time windows. *Journal of Operational Research Society*, 46, 1433–1446.
- Potvin, J., Shen, Y., & Rousseau, J. (1992). Neural networks for automated vehicle dispatching. *Computers & Operations Research*, 19(3–4), 267–276.
- Ropke, S., & Pisinger, D. (2006). An adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40, 455–472.
- Savelsbergh, M., & Sol, M. (1995). The general pick-up and delivery problem. *Transportation Science*, 29(1), 17–29.
- Savelsbergh, M., & Sol, M. (1998). Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4), 474–490.
- Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 32, 254–265.
- Taillard, E. D., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31, 170–186.

- Tam, V., & Kwan, M. C. (2004). Adapting the large neighborhood search to effectively solve pickup and delivery problems with time windows. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)* (pp. 519–523).
- Tang Montané, F., & Diéguez Galvao, R. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research*, 33(3), 595–619.
- Thompson, P., & Psaraftis, H. (1993). Cyclic transfer algorithms for multivehicle routing and scheduling problems with backhauls. *European Journal of Operational Research*, 113, 528–543.
- Toth, P., & Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31, 60–71.
- Van der Bruggen, L., Lenstra, J., & Schuur, P. (1993). Variable depth search for the single-vehicle pick-up and delivery problem with time windows. *Transportation Science*, 27, 298–311.