Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 176 (2007) 1478–1507
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows

Rodolfo Dondo, Jaime Cerdá *

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

Received 28 December 2003; accepted 27 July 2004
Available online 23 January 2006

Abstract

This paper presents a novel three-phase heuristic/algorithmic approach for the multi-depot routing problem with time windows and heterogeneous vehicles. It has been derived from embedding a heuristic-based clustering algorithm within a VRPTW optimization framework. To this purpose, a rigorous MILP mathematical model for the VRPTW problem is first introduced. Likewise other optimization approaches, the new formulation can efficiently solve case studies involving at most 25 nodes to optimality. To overcome this limitation, a preprocessing stage clustering nodes together is initially performed to yield a more compact cluster-based MILP problem formulation. In this way, a hierarchical hybrid procedure involving one heuristic and two algorithmic phases was developed. Phase I aims to identifying a set of cost-effective feasible clusters while Phase II assigns clusters to vehicles and sequences them on each tour by using the cluster-based MILP formulation. Ordering nodes within clusters and scheduling vehicle arrival times at customer locations for each tour through solving a small MILP model is finally performed at Phase III. Numerous benchmark problems featuring different sizes, clustered/random customer locations and time window distributions have been solved at acceptable CPU times.

Keywords: Logistics; Routing; Scheduling; Transportation

1. Introduction

A key issue in transportation is the cost-efficient management of a heterogeneous vehicle fleet providing pick-up/delivery service to a given set of customers with known demands. The collection/distribution system manager should not only

* Corresponding author. Tel.: +54 42 559174 77; fax: +54 42 532965.
E-mail address: jcerda@intec.unl.edu.ar (J. Cerdá).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2004.07.077

Nomenclature

Sets

I	nodes
K	clusters
P	depots
V	vehicles

Parameters

Δ	maximum allowed waiting time between intra-cluster nodes
ρ_i	penalty cost for unit-time violations of the specified time window for node i
ρ_v	penalty cost for unit-time violations of the maximum working time for vehicle v
a_i	earliest service time at node i
aC_k	earliest service time at cluster C_k
b_i	latest service time at node i
bC_k	latest service time at cluster C_k
c_{ij}^v	vehicle-dependent distance cost matrix
cf_v	fixed cost for using vehicle v
$d^{\max}$	maximum allowed distance between intra-cluster nodes
l	depot set cardinality
m	vehicle set cardinality
n	node set cardinality
nC	cluster set cardinality
q_v	capacity of vehicle v
st_i^v	service time at node i by vehicle v
$\mathrm{st}C_k$	effective service time at cluster C_k
t_{ij}^v	least-cost travel time from node i to node j
$\mathrm{tv}_v^{\max}$	maximum working time for vehicle v
w_i	demand at node i
wC_k	demand at cluster C_k

Variables

S_{ij}	binary variable denoting that node i is visited before or after node j
X_{pv}	binary variable denoting assignment of vehicle v to depot p
Y_{iv}	binary variable denoting assignment of vehicle v to node i
Δa_i	ith-time window violation due to early service
Δb_i	ith-time window violation due to late service
ΔT_v	working time violation for vehicle v
C_i	accumulated distance cost up to node i
CV_v	total distance cost for vehicle v
T_i	vehicle arrival time at node i
TV_v	tour duration for vehicle v

decide on the number and types of vehicles to be used but also he/she must specify which customers are serviced by which vehicle and what sequence to follow so as to minimize the transportation cost. Products to be delivered are loaded at the depot and picked-up products are transported back to the depot. Then, every vehicle route must start and finish at the assigned terminal and both vehicle capacity and working time constraints are to be satisfied. Moreover, each customer must be serviced by exactly one vehicle since split demand is not allowed. This class of logistic problems is usually known as the vehicle routing problem (VRP) and its objective usually is the minimization of the overall distance traveled by the vehicles while servicing all the customers. The interest in VRP problems comes from its practical relevance as well as from the considerable difficulty to solve them exactly. In the field of combinatorial optimization, the VRP is regarded as one of the most challenging problems. It is indeed *NP*-hard, so that the task of finding the best set of vehicle tours by solving optimization models is computationally prohibitive for real-world applications. As a result, different types of heuristic methodologies are usually applied.

Several classes of vehicle routing problems have been studied in the literature. Though addressing different practical situations, they all focus on the common issue of efficiently managing a vehicle fleet for the purpose of serving a given set of customers. The most basic VRP is the capacitated vehicle routing problem (CVRP) that assumes a fixed fleet of vehicles of uniform capacity housed in a central depot. It is intrinsically a spatial problem with some capacity constraints. In addition to the geographic component, more realistic routing problems include a scheduling part by incorporating travel times between every pair of nodes, customer service times and the maximum tour duration as additional problem data. The vehicle routing problem with time windows (VRPTW) is a generalization of the CVRP with the further complexity of time windows and other time data. Because the VRP is NP-hard, the VRPTW is NP-hard too (Savelsbergh, 1985). In the VRPTW problem, each customer has an associated time window defined by the earliest and the latest time to start the customer service. The depot may also have a time window defining the scheduling horizon. Time windows can be hard or soft. In the hard time window case, a vehicle arriving too early at the customer site is permitted to wait until the customer window is open. However, a vehicle is not permitted at all to arrive at the node after the latest service start time. In contrast, the soft time window case permits time window violations at the expense of a penalty cost.

In the past four decades, a tremendous amount of work in the field of vehicle routing and scheduling problems has been published. They are summarized in recent books and surveys (see Laporte, 1992; Desrosiers et al., 1995; Fisher, 1995; Bramel and Simchi-Levi, 1997 and Crainic and Laporte, 1998). Some research efforts were oriented towards the development and analysis of approximate heuristic techniques capable of solving real-size VRP problems. Bowerman et al. (1994) classified the heuristic approaches to the VRP into five classes: (1) cluster-first/route-second, (2) route-first/cluster-second, (3) savings/insertion, (4) improvement/exchange and (5) simpler mathematical programming representations through relaxing some constraints. From the two clustering procedures, the cluster-first/route-second looks more effective. This algorithm first groups the nodes into clusters, assigns each cluster to a different vehicle and, finally, finds the vehicle tour by solving the corresponding traveling salesman problem (TSP). Heuristics 3 and 4 permit to construct an initial solution or improve the current set of tours by either inserting customers or exchanging arcs. Some approximate approaches called metaheuristics, including simulated annealing, tabu search and genetic algorithms, have recently become very popular (Gendrau et al., 1997).

On the other hand, effective optimal approaches for VRPTW problems of smaller size have also been reported. Exact approaches can be categorized according to the underlying methodology into: (a) dynamic programming techniques (Kolen et al., 1987), which are extensions of the state-space relaxation method of Christofides et al. (1981); (b) Lagrangian relaxation methods which are currently capable of optimally solving some 100-customer VRPTW problems (Jornsten et al., 1986; Desrosiers et al., 1988; Halse, 1992); (c) column generation algorithms that are based on a combination of linear programming relaxed set covering and column generation (Desrochers et al., 1992), and (d) K-tree approaches that extended the classical 1-tree method for the TSP to the case with vehicle capacity and time window constraints (Fisher, 1994; Fisher et al., 1997). The first three exact approaches rely on the solution of a shortest path problem with time windows and vehicle capacity constraints either as part of a Lagrangian relaxation or to generate new columns.

Current VRPTW optimization models are useful for a variety of practical applications. However, some practical issues have not yet been addressed. While most solution methods have assumed a single depot and a homogeneous fleet, real-world problems usually include multiple terminals and a finite set of vehicles with non-uniform capacity. Moreover, the best solution for the VRPTW that minimizes the total distance traveled by the vehicles to visit all the customers often includes undesirable high waiting times. A more suitable VRPTW cost function should be a combination of fixed vehicle utilization costs and variable operational costs with the latter ones including distance and travel times, waiting time and service time costs.

In the first part, this paper presents a general mixed-integer linear (MILP) mathematical programming formulation for the VRPTW problem with several terminals and multiple vehicle types (see Section 2). Time window-based exact rules are subsequently introduced to cut the problem size and to reduce the solution time. However, the proposed exact approach remains computationally efficient for VRPTW problems of at most 25 nodes and a single depot. To overcome such an usual limitation of optimization methods, the second part describes a systematic three-phase heuristic/algorithmic hybrid approach that is capable of solving 100-customer VRPTW problems with different topologies to optimality (see Section 3). The new approach includes an initial preprocessing phase during which a heuristic-based clustering algorithm described in the paper is applied to group the customers into a small number of clusters. In this way, the general VRPTW model can be written in terms of clusters rather than nodes to generate a much smaller problem formulation. A commercial MILP branch-and-bound solver is then applied to efficiently find the optimal set of tours at the level of clusters at Phase II. Finally, the clusters on every tour are disaggregated into the original nodes through solving a small MILP model derived from the general VRPTW formulation at Phase III. Section 4 shows the numerical results found by solving a significant number of well-known Solomon's homogeneous VRPTW benchmark problems and some new multi-depot heterogeneous fleet VRPTW examples introduced in this paper.

2. The multi-depot heterogeneous VRPTW problem

2.1. Problem definition

Let us consider a routing network, represented by the directed graph $\mathrm{G}\{I, P, A\}$, connecting customer nodes $I = \{i_1, i_2, \ldots, i_n\}$ and depot nodes $P = \{p_1, p_2, \ldots, p_l\}$ through a set of directed edges $A = \{(i,j)/i,j \in (I \cup P)\}$. The edge $(i,j) \in A$ is supposed to be the lowest cost route connecting node i to node j. At each customer location $i \in I$, a fixed load w_i is to be picked up (delivered) within a time window $[a_i, b_i]$, where a_i is the earliest time and b_i is the latest time at which the service can start. A fleet of heterogeneous vehicles $V = \{v_1, v_2, \ldots, v_m\}$ with different cargo-capacities (q_v) and housed in multiple depots $p \in P$ is available to accomplish the required pick-up/delivery tasks. Each vehicle v must leave from the assigned depot $p \in P$, pick up the full load from several supply points and then return to the same terminal p. Then, the route for vehicle v is a tour of nodes $r = (p, \ldots, i, (i+1), \ldots, p)$ connected by directed edges belonging to A that starts and ends at depot p assigned to vehicle v. Associated to the set of edges $a_{ij} \in A$, there is a pair of vehicle-dependent matrices $C = \{c_{ij}^v\}$ and $\Gamma = \{t_{ij}^v\}$ denoting the travel cost and the travel time from node i to node j using vehicle v, respectively. It is assumed that the triangle inequality is satisfied by the c_{ij}'s and the t_{ij}'s, i.e. $c_{ik} + c_{kj} \geqslant c_{ij}$ and $t_{ik} + t_{kj} \geqslant t_{ij}$. The demand ($w_i$) and the service time by vehicle $v(\mathrm{st}_i^v)$ at node i are also given. Therefore, a feasible solution to the VRPTW problem must satisfy the following constraints: (i) every route must start and end at the same depot; (ii) each node must be serviced by just a single vehicle; (iii) the total load assigned to vehicle v must never exceed its cargo-capacity q_v; (iv) the length of time during which a vehicle v can be in-service should be shorter than the maximum allowed working time $\mathrm{tv}_v^{\max}$; (v) the pick-up/delivery service at every customer site i must start within the time window $[a_i, b_i]$ since otherwise a penalty cost should be charged. The problem goal is to minimize the total cost of performing the pick-up (or delivery) services at all customer nodes. Four types of costs are considered in the objective function: fixed costs for used vehicles, distance costs and travel time costs along the selected routes, waiting time costs and penalty costs due to time-window and working time constraints violations.

2.2. Problem decision variables

The proposed mathematical formulation requires to define three different sets of 0–1 variables: (a) the assignment variable Y_{iv} to allocate vehicle $v \in V$ to customer site $i \in I$; (b) the assignment variable X_{pv} to allocate vehicle $v \in V$ to depot $p \in P$; and (c) the sequencing variable S_{ij} to denote that customer site $i \in I$ is visited before ($S_{ij} = 1$) or after ($S_{ij} = 0$) node j just in case they are both serviced by the same vehicle v ($Y_{iv} = Y_{jv} = 1$). Otherwise, the value of S_{ij} will be meaningless. It is defined just a single variable S_{ij} for each pair of nodes (i, j) that can share the same tour. Therefore, the relative ordering of nodes (i, j) is established by the variable S_{ij}, such that ord(i) < ord(j), where ord(i) indicates the relative position of the element i in the customer set I. In this way, the number of sequencing variables is cut by half. It should be emphasized that the approach uses the notion of generalized predecessor rather than direct predecessor. Then, a node j on a particular tour may have one or several predecessors; i.e. $S_{ij} = 1$ for every node i visited before node j by the same vehicle.

2.3. Problem mathematical formulation

The proposed MILP mathematical formulation for the multi-depot heterogeneous fleet VRPTW problem is shown in Table 1. A thorough model explanation is subsequently made.

2.3.1. Objective function

The problem objective (1) aims to minimize the overall service expenses, including fixed vehicle utilization costs, traveling distance and time costs, waiting and service time costs and penalty costs. The parameter cf_v stands for the fixed cost of using vehicle v and the binary variable X_{pv} becomes equal to one only if vehicle v is employed, i.e.

Table 1
The MILP mathematical model

$$\text{Min} \sum_{v\in V}\left(cf_v \sum_{p\in P} X_{pv} + c_t \text{TV}_v + \text{CV}_v\right) + \rho_v \Delta T_v + \sum_{i\in I} \rho_i(\Delta a_i + \Delta b_i) \quad (1)$$

subject to

$$\sum_{v\in V} Y_{iv} = 1 \quad \forall i \in I \quad (2)$$

$$\sum_{p\in P} X_{pv} \leqslant 1 \quad \forall v \in V \quad (3)$$

$$C_i \geqslant c_{pi}^v (X_{pv} + Y_{iv} - 1) \quad \forall i \in I, p \in P, v \in V \quad (4)$$

$$C_j \geqslant C_i + c_{ij}^v - M_C(1 - S_{ij}) - M_C(2 - Y_{iv} - Y_{jv}) \quad (5a)$$

$$C_i \geqslant C_j + c_{ji}^v - M_C S_{ij} - M_C(2 - Y_{iv} - Y_{jv}) \quad \forall i, j \in I, v \in V : i < j \quad (5b)$$

$$\text{CV}_v \geqslant C_i + c_{ip}^v - M_C(2 - X_{pv} - Y_{iv}) \quad \forall i \in I, p \in P, v \in V \quad (6)$$

$$T_i \geqslant t_{pi}^v (X_{pv} + Y_{iv} - 1) \quad \forall i \in I, p \in P, v \in V \quad (7)$$

$$T_j \geqslant T_i + st_i + t_{ij}^v - M_T(1 - S_{ij}) - M(2 - Y_{iv} - Y_{jv}) \quad (8a)$$

$$T_i \geqslant T_j + st_j + t_{ji}^v - M_T S_{ij} - M(2 - Y_{iv} - Y_{jv}) \quad \forall i, j \in I, v \in V : i < j \quad (8b)$$

$$\text{TV}_v \geqslant T_i + st_i + t_{ip}^v - M_T(2 - X_{pv} - Y_{iv}) \quad \forall i \in I, p \in P, v \in V \quad (9)$$

$$\Delta a_i \geqslant a_i - T_i \quad \forall i \in I \quad (10)$$

$$\Delta b_i \geqslant T_i - b_i \quad \forall i \in I \quad (11)$$

$$\Delta T_v \geqslant \text{TV}_v - tv_v^{\max} \quad \forall v \in V \quad (12)$$

$$\sum_{i\in I} w_i Y_{iv} \leqslant q_v \sum_{p\in P} X_{pv} \quad \forall v \in V \quad (13)$$

whenever it was assigned to a depot p. Since the tour duration includes all travel times, waiting times and service times, then a single term is included in the objective function to charge time-based costs like labor expenses. Such type of cost is assumed to be a linear function of time and the parameter c_t denotes the labor cost per unit time. In addition, CV_v represents the total distance-based travel cost and the last two terms penalize violations on either the maximum allowed working time or the node time windows.

2.3.2. Problem constraints

- Assignment of nodes to vehicles
  Eq. (2) states that every customer node $i \in I$ must be serviced by a single vehicle $v \in V$. Splitting the load to be picked-up from a customer site is a forbidden option.
- Assignment of vehicles to depots
  Constraint (3) states that every used vehicle v should be allocated to a single depot p to which it returns after visiting all the assigned customers. The required fleet size is a problem variable to be determined simultaneously with the best set of routes and schedules.
- Least traveling cost for vehicle v to arrive at node i
  Constraint (4) states that the cost of traveling from depot p to node i (C_i) must be greater than or equal to c^v_{pi} only if the node $i \in I$ is serviced by vehicle v ($Y_{iv} = 1$) housed in depot p ($X_{pv} = 1$). This is so because, by definition, c^v_{pi} denotes the least travel cost from depot p to node i. Constraint (4) can become binding just in case customer i is first visited by vehicle v.
- Relationship between traveling costs up to nodes $i, j \in I$ on the same tour
  Let c^v_{ij} stand for the least travel cost from node i to node j on the vehicle v. If both nodes (i, j) are on the same tour ($Y_{iv} = Y_{jv} = 1$, for some vehicle v) and node i is visited before ($S_{ij} = 1$), then Eq. (5a) states that the distance-based travel cost from the depot to node j (C_j) must always be greater than C_i by at least c^v_{ij}. In case node j is visited earlier ($S_{ij} = 0$), the reverse statement holds. Constraints (5a) and (5b) both become redundant whenever nodes $i, j \in I$ are serviced by different vehicles ($Y_{iv} + Y_{jv} < 2$, for any v). By definition, M_C is a large positive number.
- Overall traveling cost along the tour assigned to vehicle v
  Constraint (6) states that the overall traveling cost incurred by vehicle v (CV_v) to complete the assigned pick-up/delivery tasks must always be greater than the traveling expenses from the depot to any node i (C_i) along the tour by at least the amount c^v_{ip}. Indeed, the last node visited by vehicle v is the one finally defining the value of CV_v. Therefore, the constraint (6) related to such a node and the assigned vehicle v is just the one binding at the optimum. If nodes $i, i' \in I$ are both on the same tour and i' is the last visited, then constraint (6) for node i will usually become redundant because the travel cost c^v_{ip}, 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. $c^v_{ip} \leqslant c^v_{ii'} + c^v_{i'p}$.
- Earliest service starting time at node i
  Constraint (7) states that the vehicle v will never begin the service at the assigned node i before time t^v_{pi}, where t^v_{pi} is the least travel time from depot p to node i. Constraint (7) assumes that vehicle v is ready at $t = 0$. Otherwise, the v-vehicle ready time should be added to t^v_{pi}. M_T is a positive large number.
- Relationship between the service starting times at the pair of nodes (i, j) on the same tour
  Let us assume that nodes i and j are both serviced by the same vehicle v. If node i is visited before ($S_{ij} = 1$), then the constraint (8a) states that the service starting time at node j (T_j) should be greater than T_i by at least the sum of both the traveling time t^v_{ij} and the service time (st_i) at node i. If not ($S_{ij} = 0$), the reverse statement holds and constraint (8b) will become active. If one of the nodes is not on the tour, then $Y_{iv} + Y_{jv} < 2$ and constraints 8a,8b both become redundant.
- Overall traveling time for vehicle v
  Constraint (9) indicates that the total time required by vehicle v to complete the tour is found by adding the sum of both the service time st_i at node i and the travel time t^v_{ip} along

the edge (i, p) to the service initial time at the node last visited i (i.e., the largest service initial time). Since the node last visited by vehicle v is not known beforehand, then Eq. (9) should be written for every node i.

- Time constraint violations due to early/late services at customer sites
  Time windows can be hard or soft. When the time windows are regarded as hard constraints, constraint (10) states that a vehicle cannot start the service at the assigned node i before the earliest time a_i by simply making $\Delta a_i = 0$. In turn, constraint (11) prohibits to start the service at node i after the allowed latest time b_i by setting $\Delta b_i = 0$. If the vehicle arrives too early at the customer site, it must wait until the customer is ready to start the service. This is an option allowed by the model. In the soft time window case, time window constraints can be violated at a finite cost and the vehicle can start the service at node i before time a_i. In such a case, the variables Δa_i and Δb_i stand for the size of TW-constraint violations caused by early or late service at node i, respectively.
- Time constraint violation for vehicle v
  Constraint (12) applies just in case the maximum allowed working time $\mathrm{tv}_v^{\max}$ is regarded as a soft constraint that can be violated at some penalty cost. Otherwise, TV_v should not be greater than $\mathrm{tv}_v^{\max}$.
- Capacity constraints
  Constraint (13) states that the overall load to pick up from/deliver to customer sites serviced by a used vehicle v should never exceed its cargo-capacity q_v. Any used vehicle v is assigned to a depot p and, therefore, $\Sigma_p X_{pv} = 1$.
  The continuous problem variables, T_i, TV_v, C_i, CV_v, Δa_i, Δb_i, and ΔT_v, are all non-negative. Assignment constraints 2 and 3 together with traveling cost constraints (4)–(6) and vehicle capacity constraints (13) all define the feasible solution space for the traditional VRP. Timing constraints include visiting time constraints (7)–(9) and soft time constraint violations (10)–(12). No sub-tour breaking constraints are necessary. Time-window and working time constraints can be treated as hard constraints by simply setting all variables Δa_i, Δb_i and ΔT_v equal to zero and, in addition, the penalty cost terms are removed from the objective function. In short, the proposed general mathematical model can account for a fleet of fixed or variable size composed by either homogeneous or heterogeneous vehicles and housed at single or multiple depots. The new MILP formulation can still be applied if open tours not starting and ending at the same location are considered.

2.4. Exact elimination rules

The information on customer time windows can be used in order to reduce the VRPTW problem size, thus enhancing the efficiency of the solution algorithm. The narrower the time windows the larger the number of sequencing variables and constraints that can be deleted from the problem formulation. The time window-based elimination rules that consider the time windows as hard constraints can be stated as follows:

Rule 1. If no vehicle $v \in V$ can service a pair of nodes $i, j \in I$ without violating the related time windows constraints, then two different vehicles must be used to visit them. Whatever is the node first visited, the service at the other node can never begin before the latest start time unless another vehicle is used. Mathematically, this condition can be expressed as follows:

$$\forall i, j \in I, v \in V: \quad i < j \wedge \left(a_i + \mathrm{st}_i^v + t_{ij}^v\right) \geqslant b_j \wedge \left(a_j + \mathrm{st}_j^v + t_{ji}^v\right) \geqslant b_i \Rightarrow Y_{jv} + Y_{iv} \leqslant 1. \tag{14}$$

When Rule 1 applies to a pair of nodes $i, j \in I$, then the constraint $Y_{iv} + Y_{jv} \leqslant 1$ must be included in the problem formulation. Constraint (14) indicates that vehicle v can service either node i or node j but not both. Thus, constraints (5a)–(5b) and (8a)–(8b) can be eliminated for any pair $\{i, j\}$ satisfying condition (14). Moreover, the sequencing variable S_{ij} is no longer needed since both nodes cannot belong to the same tour.

To illustrate this rule, let us suppose that the service times at nodes $i, j \in I$, are: $\mathrm{st}_i = st_j = 10$. In addition, the node time windows are given by $[a_i, b_i] = [10, 20]$, $[a_j, b_j] = [20, 30]$ and the travel time from either i to j or j to i is equal to 30. If node i is first visited at the earliest time $t = a_i = 10$, then the earliest service start time at node j will be $a_i + \mathrm{st}_i + t_{ij} = 10 + 10 + 30 = 50 > b_j$ (see Fig. 1a). Therefore, node j cannot be serviced within its related time window by a vehicle departing from node i. Inversely, if node j is first visited at the earliest time $t = a_j = 20$, then the earliest service start time at node i will be given by $a_j + \mathrm{st}_j + t_{ji} = 20 + 10 + 30 = 60$. Since $60 > b_i = 20$, node i cannot be visited by a vehicle departing from node j without violating its time window (see Fig. 1a). Consequently, if both nodes are on the same tour, a time window constraint will be violated. Therefore, it can be imposed the condition: $Y_{iv} + Y_{jv} \leqslant 1$ for any $v \in V$.

Rule 2. Let us suppose that nodes i and j are visited by the same vehicle v. Moreover, the sum of the earliest service start time at node i (a_i) and the travel time between both nodes, including the service time at node i, is higher than the latest service start time at node j (b_j). Then, node i cannot be visited before node j and $S_{ij} = 0$. Therefore,

$$\forall i, j \in I, v \in V: \quad i < j \wedge \left(a_i + \mathrm{st}_i^v + t_{ij}^v\right) > b_j \wedge \left(a_j + \mathrm{st}_j^v + t_{ji}^v\right) \leqslant b_i \Rightarrow S_{ij} = 0. \tag{15}$$

If Rule 2 applies, then node i cannot be visited before node j. As a result, the sequencing variable S_{ij} and the constraints (5a) and (8a) can all be dropped from the problem formulation. More-

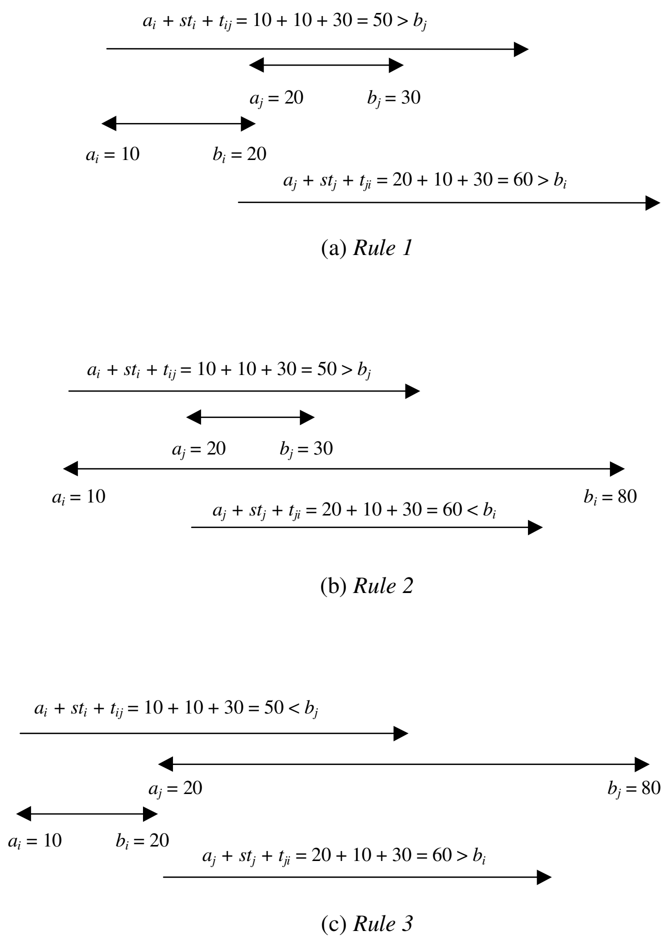

Fig. 1. Illustrating the exact elimination rules.

over, constraints (5b) and (8b) will reduce to the simpler Eqs. (16) and (17), respectively.

$$C_i \geqslant C_j + c^v_{ji} - M_C(2 - Y_{iv} - Y_{jv}), \tag{16}$$

$$T_i \geqslant T_j + \mathrm{st}^v_j + t^v_{ji} - M_T(2 - Y_{iv} - Y_{jv}). \tag{17}$$

Constraints (16) and (17) really apply just in case both nodes are on the same tour ($Y_{iv} + Y_{jv} = 2$). Otherwise, they become redundant.

Let us suppose that the service times at nodes $i, j \in I$, are: $\mathrm{st}_i = \mathrm{st}_j = 10$ and the time windows are given by $[a_i, b_i] = [10, 80]$ and $[a_j, b_j] = [20, 30]$. Moreover, the travel times from i to j and from j to i are: $t_{ij} = t_{ji} = 30$. If node i is first visited at the earliest time $t = a_i = 10$, then the earliest service start time at node j will be given by $a_i + \mathrm{st}_i + t_{ij} = 10 + 10 + 30 = 50 > b_j$. Therefore, node j cannot be visited within its related time window by a vehicle departing from node i (see Fig. 1b). Inversely, if node j is first visited at the earliest time $t = a_j = 20$, then the earliest service start time at node i will be: $a_j + \mathrm{st}_j + t_{ji} = 20 + 10 + 30 = 60$. Since $60 < b_i = 80$, node i can be visited by a vehicle departing from node j without violating its time window. Consequently, if both nodes are on the same tour, $S_{ij} = 0$.

Rule 3. Let now suppose that nodes i and j have been assigned to the same tour. Furthermore, the sum of the earliest service start time at node j (a_j) and the travel time along the route (j, i), including the service time at node j, is higher than the latest service start time at node i (b_i). Then, node j cannot precede node i on the tour and $S_{ij} = 1$:

$$\forall i, j \in I, v \in V: \quad i < j \wedge \left(a_i + \mathrm{st}^v_i + t^v_{ij}\right) \leqslant b_j \wedge \left(a_j + \mathrm{st}^v_j + t^v_{ji}\right) > b_i \Rightarrow S_{ij} = 1. \tag{18}$$

Therefore, constraints (5b) and (8b) can be eliminated from the model and inequalities (5a) and (8a) reduce themselves to

$$C_j \geqslant C_i + c^v_{ij} - M_C(2 - Y_{iv} - Y_{jv}), \tag{19}$$

$$T_j \geqslant T_i + \mathrm{st}^v_i + t^v_{ij} - M_T(2 - Y_{iv} - Y_{jv}). \tag{20}$$

To illustrate Rule 3, let us consider a simple example where the service times at nodes $i, j \in I$, are: $\mathrm{st}_i = \mathrm{st}_j = 10$, their related time windows are given by $[a_i, b_i] = [10, 20]$, and $[a_j, b_j] = [20, 80]$ and the travel times from i to j and vice versa is equal to $t_{ij} = t_{ji} = 30$. If node i is first visited at the earliest time $t = a_i = 10$, then the earliest service start time at node j will be $a_i + \mathrm{st}_i + t_{ij} = 10 + 10 + 30 = 50$. Since $50 < b_j = 80$, service at node j can be started within the specified time window by a vehicle departing from node i (see Fig. 1c). Inversely, a vehicle first visiting node j at the earliest start time $t = a_j = 20$, cannot start servicing node i before its time window has closed because $a_j + \mathrm{st}_j + t_{ji} = 20 + 10 + 30 = 60 > b_i$. Consequently, $S_{ij} = 1$ and node i must be a predecessor of node j.

3. A three-phase hierarchical hybrid approach for large-scale VRPTW problems

There is no doubt that the multi-depot heterogeneous fleet VRPTW is very difficult to solve through a pure optimization approach. In fact, even simpler vehicle routing problems are among the most difficult class of combinatorial optimization problems. Current fastest algorithms can discover the optimal solution for single-depot homogeneous fleet VRPTW problem instances featuring tight time windows and up to 100 customers (Desrochers et al., 1992; Fisher et al., 1997). Problems with wider time windows and a number of nodes over 100 are still considered computationally challenging to solve to optimality. In general, heuristics can approximately solve problems of larger sizes in less computational time. For example, meta-heuristics such as tabu-search, simulated annealing and genetic algorithms (Gendrau et al., 1997; Golden et al., 1998) are able to solve vehicle routing problems with wide time windows and nearly 500 customers.

However, heuristics usually lack robustness and their performance is problem dependent. Instead, optimization algorithms offer the best promise for robustness (Fisher, 1995). Given the enormous complexity of large VRPTW problems, however, it does not seem realistic to apply pure optimization methods. Instead, we can focus on hybrid solution algorithms that can be as robust as the optimization methods and capable of discovering good solutions for large problems within acceptable CPU times. In this work, it has been developed a hierarchical hybrid solution approach that integrates a heuristic clustering procedure into an optimization framework (see Fig. 2). It is based on the traditional *cluster first-route second* philosophy. Clusters of nodes are first defined, then such clusters are assigned to vehicles and sequenced on the related tours and finally the routing and scheduling for each individual tour in terms of the original nodes is separately found. In this way, a three-phase VRPTW hierarchical hybrid approach has been defined.

Finding a good set of clusters, each one comprising several customer sites, without relying on routing information is a quite difficult task. Consequently, this paper introduces a time-window based heuristic algorithm that efficiently assemblies customer nodes into a rather low number of feasible clusters. Such a heuristic clustering procedure leads to a compact version of the VRPTW formulation presented in Section 2.2 by just replacing nodes by clusters. Both the cluster procedure and the compact VRPTW model constitute the basic building blocks of the proposed hybrid approach. After grouping customer nodes into a few clusters during Phase I, the solution of a low-size multi-depot heterogeneous fleet VRPTW formulation, posed in terms of clusters rather than nodes, permits to simultaneously allocate clusters to vehicles and construct routes by linking clusters on the same tour (Phase II). In the last phase, the detailed routing and scheduling for each tour found in Phase II is determined. Therefore, as many scheduling problems are to be tackled in Phase III as the number of tours required to visit all the customer nodes. In each case, a more compact form of the basic VRPTW formulation is to be solved. This is so because the model just account for the nodes contained in the clusters over the tour under analysis. Since the sequence of clusters has already been defined in Phase II, then the relative location of nodes belonging to different clusters is already established and the values of many sequencing variables S_{ij} are known beforehand. Such a fact produces a sharp decrease

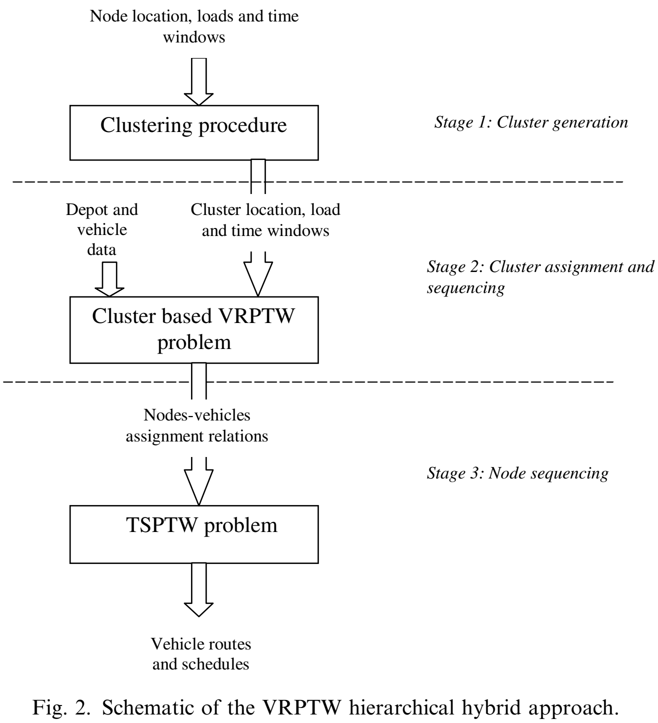

Fig. 2. Schematic of the VRPTW hierarchical hybrid approach.

in the number of binary variables and constraints in the MILP model solved in Phase III.

3.1. Heuristic clustering algorithm (Phase I)

Phase I is intended to massively reduce the computational burden of the subsequent solution phases. This goal is achieved by cleverly defining a small set of feasible clusters or "hyper-nodes", each one enclosing several customer sites, and then establishing approximate travel distances and times between any pair of them. By expressing the mathematical model in terms of few clusters rather than a huge number of customers, the VRPTW problem size can be sharply decreased. To rapidly find a good set of clusters for large VRPTW, the heuristic procedure shown in Fig. 3 is applied.

The procedure inputs are the set of customer nodes I, the set of vehicles V, travel distances and times among nodes, service times and pick-up/delivery loads as well as time-window data. The aim of the procedure is to identify a set of feasible clusters that includes all customer nodes, where "feasible" cluster means that (a) the cluster cargo can be assigned to a single vehicle and, in

1. (a) Open a list of nodes L and sort them by increasing values of the earliest arrival times a_i. If several nodes have the same a_i, arrange them by increasing values of the latest arrival times b_i.
   (b) Open a list of available vehicles V and sort them by decreasing values of the ratio (q_v/cf_v).
   (c) Choose the maximum allowed distance between any pair of nodes in the same cluster (d^{max}) and the maximum allowed waiting time Δ.
2. (nth-major iteration) Open an empty list K_n linked to the next cluster C_n to be created. Assign the top entry of list V to cluster C_n and delete it from V.
3. (a) Pick up the top node i on the list L and place it at the bottom of list K_n. Initialize the parameters of cluster C_n:
   $$aC_n \leftarrow a_i \qquad bC_n \leftarrow b_i$$
   $$wC_n \leftarrow w_i \qquad stC_n \leftarrow st_i$$
   (b) Delete node i from list L and make a copy of the current list L and call it L'.
4. Pick up the top node j from list L', and verify that the current load to pick up from cluster C_n plus w_j does not exceed the cargo-capacity q_v of the assigned vehicle v. If the vehicle capacity is exceeded, delete node j from list L' and repeat step (4). Otherwise, proceed to step (5).
5. (a) Compute the distance d_{ji} between node j and its nearest node i on the list K_n.
   (b) Verify that d_{ij} is smaller than the maximum allowed distance d^{max}. If not, delete node j from the temporary list L' and return to step (4). Otherwise, proceed to step (6).
6. Verify that the following constraint is satisfied:
   $$aC_n + stC_{ni} + t_{ij}^v \le \max(bC_n, b_j)$$
   If not, delete node j from the temporary list L' and return to step (4). Otherwise, proceed to step (7).
7. Verify that the following constraint is satisfied:
   $$aC_n + stC_n + t_{ij}^v + \Delta \ge a_j$$
   If not, close the cluster C_n by deleting the temporary list L' and saving the list K_n defining C_n and return to step (4). Otherwise, proceed to step (8).
8. (a) Place node j at the bottom of list K_n and update the parameters for cluster C_n as follows:
   $$wC_n \leftarrow wC_n + w_j \qquad stC_n \leftarrow \max\left(stC_n + t_{ij}^v + st_j, a_j + st_j - a_i\right)$$
   (b) If $bC_n > b_j$, then:
   $$bC_n \leftarrow b_j$$
   Otherwise, the latest time arrival bC_n remains unchanged. Delete node j from lists L and L' and go to step (9).
9. If list L' is empty, save the list K_n defining the cluster C_n and proceed to step (10). Otherwise, return to step (4).
10. Repeat steps 2-9 until the list L is empty.
11. Compute the cluster centroids as well as the time and distance between any pair of clusters defined by the algorithm.

Fig. 3. The heuristic clustering procedure.

addition, (b) there exists a route connecting the nodes on the cluster that satisfies all the time window constraints. The set of clusters to be synthesized should also be cost-effective, in the sense that (c) the vehicle waiting time because of early arrivals at pick-up/delivery points must be kept as small as possible and (d) the average length per node traveled by the assigned vehicle throughout the cluster should remain low.

To reach such goals, the node list L is properly arranged in step (1) to facilitate the generation of feasible, cost-effective clusters. Before adding another node to the cluster being generated, its closeness to the other nodes in the cluster and the fulfillment of time window and vehicle capacity constraints are tested (steps 4–7). Moreover, a maximum idle time Δ for early arrival at the customer site (i.e. before the earliest service start time) is just permitted. If exceeded, the incorporation of the node in the cluster is rejected (step 7).

To define the mathematical formulation in terms of clusters rather than nodes, the notions of "cluster time window" and "cluster service time" have been introduced. The earliest service start time at cluster C_n (aC_n) is given by $\min_{i\in C_n}(a_i)$, while the latest service start time for C_n (bC_n) is updated by taking the minimum of (bC_n, b_i), with bC_n denoting the current C_n-latest service start time and b_i representing the corresponding value for the new entry i in the cluster C_n. In this manner, the time windows for the current nodes in C_n are all satisfied and the cluster time windows become much tighter. Each time a new node is added to a cluster, the related time window parameters (aC_n, bC_n) are updated. On the other hand, the "cluster service time" stC_n is a good approximation to the overall time expended by the assigned vehicle while visiting the cluster C_n. Therefore, it includes not only the service time at the enclosed nodes but also the travel and idle times throughout the cluster.

Since the fleet size is a problem variable, the procedure must choose the most efficient vehicles to accomplish the pick-up tasks. This can be achieved by sorting the vehicle list in step (2) by increasing cost-efficiency values. Finally, step (11) determines the cluster "locations" as well as the travel distance and the travel time between any pair of clusters.

3.2. Cluster-based multi-depot heterogeneous fleet VRPTW problem (Phase II)

The aim of Phase II is to assign clusters to vehicles and then sequence those ones on the same tour by solving a compact version of the MILP model introduced in Section 2.2. Such shrinkage in the model size is achieved through replacing the customer nodes by the clusters generated in Phase I. In the worst case, the number of binary variables (BV) drops from:

$$\mathrm{BV} = lm + mn + \frac{n(n-1)}{2} \quad \text{to}$$

$$\mathrm{BV} = lm + mnC + \frac{nC(nC-1)}{2}$$

where nC denotes the total number of clusters for the VPRTW problem. If nodes(n) $= 200$, depots(l) $= 1$, vehicles(m) $= 10$ and clusters(nC) $=$ 15, the value of BV drops from 21,910 to 265, i.e. almost a two-order-of magnitude reduction. Because of the way the parameter bC_n is updated (step 8b), the clusters generated in Phase I can often feature narrow time-windows. Therefore, the application of the exact elimination rules at the level of clusters may lead to a further decrease of the VRPTW model size. After accomplishing Phase II, the following tasks have been completed: (i) assignment of customer nodes via clusters to vehicles; (ii) allocation of used vehicles to depots; (iii) discovery of a near-optimal set of cluster-based tours and (iv) the sequence of clusters on the same tour that indirectly provides a partial arrangement of the customer sites visited by the same vehicle.

3.3. The single tour scheduling problem (Phase III)

Ordering nodes within clusters and scheduling the service start times at the customer locations for every tour is the aim of Phase III. To reach that goal, a low-size version of the VRPTW mathematical formulation introduced in Section 2.2 must be solved as many times as the number of tours found in Phase II. By including just the nodes in the clusters linked to the same tour, a TSP-type formulation for each single-tour scheduling problem can be derived. Moreover, the relative ordering of the clusters along the tour found in Phase II will permit to further reduce the number of sequencing variables S_{ij}. To illustrate this statement, let us consider a pair of nodes $i \in C_l$ and $j \in C_{l'}$ and assume that clusters $(C_l, C_{l'})$ are on the same tour and C_l is visited earlier by the assigned vehicle. Then, node i will surely precede node j and, consequently, $S_{ij} = 1$ and the exact elimination rule # 3 can be applied. By so doing, the variable S_{ij} and the related constraints (4) and (7) can all be eliminated from the problem formulation. For the reverse situation, S_{ij} will be equal to zero and the exact elimination rule # 2 can be enforced. In this way, the formulation avoids including cost/time sequencing constraints for every pair of nodes belonging to different clusters. Often, such a saving in binary variables allows one to solve the whole set of individual tour scheduling problems all at once in a short CPU time.

4. Results and discussion

To show the computational performance of the proposed three-phase hierarchical hybrid VRPTW approach, several examples from the classical collection of 56 Solomon's homogeneous VRPTW benchmark problems have been solved. A full description of Solomon's problems can be found in Solomon (1987). In addition, new test examples involving multiple depots and heterogeneous fleets have also been tackled. In this way, we aim to: (i) show that the new hybrid methodology is able to provide good solutions to moderate size instances of the multi-depot heterogeneous fleet VRPTW problem in a reasonable CPU time, and (ii) assess the effectiveness of the three-phase hybrid approach by solving several Solomon's homogeneous VRPTW instances and subsequently comparing the solutions found and the CPU times required with known optimal solutions and computational needs reported in the literature. This will allow us to conclude that the proposed strategy is able to often find optimal solutions to instances with clustered customers and good solutions to instances with randomly distributed customers.

Solomon's benchmark problems have been grouped into three different categories: C, R and RC. The data set for every category comprises from 8 to 12 examples all involving 100 nodes with the same geographical distribution, a central depot, similar vehicle capacities but different time windows. Problems of class C have clustered customers whose time windows have been generated based on known solutions. R-class problems have customer locations uniformly randomly generated over a square. In turn, RC-class problems have a combination of clustered and randomly generated customers and time windows. Problems of each class are further classified into two types "1" and "2", like C1 and C2. Type-1 problems have narrow time windows and a small vehicle capacity, while type-2 problems have wider time windows and a larger vehicle capacity. Therefore, solutions to type-2 problems include fewer tours and longer scheduling horizons. Problem data also 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. Furthermore, time windows are regarded as hard constraints, service times are assumed to be independent of the customer requirements and the tour duration cannot exceed a maximum value $t_v^{\max}$. The selected problem objective is the minimization of the total distance cost.

Since test examples for the multi-depot heterogeneous fleet VRPTW problem are barely reported in the literature, some new benchmark problems have been introduced. They result from generating a modified version of Solomon's instance C-101 that includes both a heterogeneous fleet and a pair of depots.

Though one can expect a much better performance with problems of types "C" and "RC", the proposed approach has been used to tackle problems of any category. Results for a significant number of examples involving up to 100 nodes are next reported. They have been found by using ILOG OPL Studio 3.5 (script mode) on a 733 MHz Pentium III PC. The effectiveness of the strategy is first studied by thoroughly analyzing the solution generated at each phase for two small examples: C-101(25) and C-102(25) problems, each involving 25 nodes and 3 vehicles with a uniform capacity of 200 units. In Phase I, the

Table 2
Results for problem C-101(25) and some variants

Problem type	Binary variables	Continuous variables	Constraints	Objective function	Solution time[a]
Single depot—homogeneous fleet					
Exact approach	93	56	2274	191.8	43.94
Hybrid approach (Phase II/Phase III)	21/0	18/56	171/2274	191.8	0.16/0.16
Single depot—heterogeneous fleet					
Exact approach	93	56	2274	193.23	71.5
Hybrid approach (Phase II/Phase III)	21/0	18/56	171/2274	193.23	0.06/0.16
Two depots—homogeneous fleet					
Exact approach	96	56	2350	162.68	1451.41
Hybrid approach (Phase II/Phase III)	24/0	18/56	171/2274	162.68	0.39/0.28
Two depots—heterogeneous fleet					
Exact approach	96	56	2350	172.93	2924.4
Hybrid approach (Phase II/Phase III)	24/0	18/56	171/2274	172.93	0.28/0.28

[a] Seconds ILOG on a 733 MHz Pentium III PC by using OPL Studio 3.5 (script mode).

Table 3a
Optimal tour schedules for example C-101(25 nodes)

Vehicle	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	T13	0.00	30.81	120.81	30			
	T17	0.00	124.81	214.81	20			
	T18	0.00	217.81	307.81	20			
	T19	0.00	312.81	402.81	10			
	T15	0.00	407.81	497.81	40			
	T16	0.00	502.81	592.81	40			
	T14	0.00	594.81	684.81	10			
	T12	0.00	687.81	777.81	20	190	95.88	815.88
V2	T5	0.00	15.13	105.13	10			
	T3	0.00	106.13	196.13	10			
	T7	0.00	198.13	288.13	20			
	T8	0.00	290.96	380.96	20			
	T10	0.00	384.57	474.57	10			
	T11	0.00	477.57	567.57	10			
	T9	0.00	570.73	660.73	10			
	T6	0.00	662.96	752.96	20			
	T4	0.00	755.20	845.20	10			
	T2	0.00	848.81	938.81	30			
	T1	0.00	967.00	1057.00	10	160	59.48	1075.68
V3	T20	0.00	10.00	100.00	10			
	T24	0.00	105.00	195.00	10			
	T25	0.00	197.00	287.00	40			
	T23	440.76	732.00	822.00	10			
	T22	0.00	825.00	915.00	20			
	T21	0.00	917.00	1007.00	20	110	36.44	1017.20

Objective function: 191.80.

maximum allowed distance between intra-cluster nodes $d^{\max}$ has been chosen equal to 10, almost one-third of the average distance among nodes for both benchmark problems. The procedure starts by setting $d^{\max}$ equal to the average inter-node distance in order to generate few clusters. Afterwards, $d^{\max}$ is gradually reduced by a factor 0.8 until either no solution improvement is achieved or the CPU time required for solving the cluster-based MILP formulation in Phase II sharply increases. In turn, the maximum allowed waiting time Δ has been adopted equal to $\max(\mathrm{st}_i, 0.05t_v^{\max})$. Results indicate that the morphology of the synthesized clusters is not very sensitive to minor changes in $d^{\max}$ and Δ.

4.1. Example C101(25)

Example C-101(25) has been derived from the original benchmark problem C-101 comprising 100 nodes (Solomon, 1987) by just considering the first 25 customers. Three variants of Example C-101(25) involving (a) single terminal/heterogeneous fleet; (b) two depots/homogenous fleet and (c) two terminals/heterogeneous fleet were also tackled. The optimization method based on the VRPTW mathematical formulation introduced in Section 2 was first used to solve Example C-101(25) and their three variants. Model sizes, solution times and the optimal objective value for the four cases are all reported in Table 2. Euclidean coordinates for the new depot are: $X = 30$ and $Y = 55$, while the heterogeneous fleet is composed by 2 vehicles of 150-unit capacity and a third one of 250-unit capacity. The true optimal solution to the original Example C101(25) was found in 43.94 seconds. When a second depot was considered, without any further restriction, the solution time increased to 1451 seconds. Moreover, the solution time for the heterogeneous-fleet case was 71.5 seconds for one depot and 2924.4 seconds for two depots, respectively. Optimal node-based

Table 3b
Optimal tour schedule for the heterogeneous-fleet C-101(25) variant

Vehicle	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	T20	0.00	10.00	100.00	10			
	T24	0.00	105.00	195.00	10			
	T25	0.00	224.00	314.00	40			
	T23	440.76	732.00	822.00	10			
	T22	0.00	825.00	915.00	20			
	T21	0.00	917.00	1007.00	20	110	36.44	1017.20
V2	T5	0.00	15.13	105.13	10			
	T3	0.00	106.13	196.13	10			
	T7	0.00	198.13	288.13	20			
	T8	0.00	290.96	380.96	20			
	T10	0.00	384.57	474.57	10			
	T11	0.00	477.57	567.57	10			
	T9	0.00	570.73	660.73	10			
	T6	0.00	662.96	752.96	20			
	T4	0.00	755.20	845.20	10	120	53.30	863.31
V3	T13	0.00	30.80	120.80	30			
	T17	0.00	124.80	214.80	20			
	T18	0.00	217.80	307.80	20			
	T19	0.00	312.80	402.80	10			
	T14	0.00	594.80	684.80	10			
	T15	0.00	407.80	497.80	40			
	T16	0.00	502.80	592.80	40			
	T12	0.00	687.80	777.80	20			
	T2	22.20	825.00	915.00	30			
	T1	0.00	917.00	1007.00	10	230	103.49	1025.68

Objective function: 193.23.

tour schedules are summarized in Tables 3a–3d and Figs. 4a–4d.

Next, the three-phase hierarchical hybrid VRPTW approach was applied to the original Example C101(25) involving an homogeneous fleet and a single depot. The 25 original nodes have been merged into four customer clusters. The nodes in each cluster are shown in Table 4 in the same order they were added to the cluster entry list. In Phase II, the three available vehicles are used to service the clusters. Two of them visit just a single cluster while the remaining one successively provides service to the sequence of clusters C^1 and C^2 (see Table 5). In Phase III, clusters are disaggregated into the original nodes to find the three tour schedules. The solution depicted in Fig. 4a and Table 3a is the truly problem optimum already found through the exact approach. The same vehicle schedules are still found even if the cluster sequencing on every tour provided by Phase II is explicitly considered to decrease the model size solved in Phase III. Moreover, the nodes in each cluster are visited in the same order they appear in the cluster entry list. Note that the total travel-times for vehicles V1 and V2 are quite similar to those estimated in Phase II (see Table 3a). In addition, the total travel time for vehicle V3 is quite similar to the sum of both the service time for clusters C^1 and C^2 and the waiting time before servicing cluster C^2. A similar conclusion can be drawn when estimated and real travel distances are compared. Note that the large waiting time before V3 starts servicing cluster C2 arises because the selected problem objective ignores time-related costs. In summary, the clustering procedure leads to a cluster-based problem formulation of smaller size that still is a good representation of the original VRPTW problem.

Table 3c
Optimal tour schedule for the two-depot C-101(25) variant

Vehicle	Depot	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	P1	T5	0.00	15.13	105.13	10			
		T3	0.00	106.13	196.13	10			
		T7	0.00	198.13	288.13	20			
		T8	0.00	290.96	380.96	20			
		T10	0.00	384.57	474.57	10			
		T11	0.00	477.57	567.57	10			
		T9	0.00	570.73	660.73	10			
		T6	0.00	662.96	752.96	20			
		T4	0.00	755.20	845.20	10			
		T2	0.00	848.80	938.80	30			
		T1	0.00	940.80	1030.80	10	160	59.98	1049.49
V2	P2	T20	5.00	10.00	100.00	10			
		T24	0.00	105.00	195.00	10			
		T25	0.00	224.00	314.00	40			
		T23	0.00	732.00	822.00	10			
		T22	0.00	825.00	915.00	20			
		T21	0.00	917.00	1007.00	20	110	29.24	1010.00
V3	P2	T13	8.46	30.00	120.00	30			
		T17	0.00	124.00	214.00	20			
		T18	0.00	217.00	307.00	20			
		T19	0.00	312.00	402.00	10			
		T15	0.00	407.00	497.00	40			
		T16	0.00	502.00	592.00	40			
		T14	0.00	594.00	684.00	10			
		T12	0.00	687.00	777.00	20	190	88.22	807.41

Objective function: 177.44.

Table 3d
Optimal tour schedule for the two-depot heterogeneous-fleet C-101(25) variant

Vehicle	Depot	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	P2	T5	0.00	15.62	105.62	10			
		T3	0.00	106.62	196.62	10			
		T7	0.00	198.62	288.62	20			
		T8	0.00	291.45	381.45	20			
		T10	0.00	385.05	475.05	10			
		T11	0.00	478.05	568.05	10			
		T9	0.00	571.21	661.21	10			
		T6	0.00	663.45	753.45	20			
		T4	0.00	755.69	845.69	10	120	53.38	863.38
V2	P2	T20	5.00	10.00	100.00	10			
		T24	0.00	105.00	195.00	10			
		T25	0.00	197.00	287.00	40			
		T23	0.00	732.00	822.00	10			
		T22	0.00	825.00	915.00	20			
		T21	0.00	917.00	1007.00	20	110	29.24	1010.00
V3	P1	T13	0.00	30.80	120.80	30			
		T17	0.00	124.80	214.80	20			
		T18	0.00	217.80	307.80	20			
		T19	0.00	312.80	402.80	10			
		T15	0.00	407.80	497.80	40			
		T16	0.00	502.80	592.80	40			
		T14	0.00	594.80	684.80	10			
		T12	0.00	687.80	777.80	20			
		T2	22.20	825.00	915.00	30			
		T1	0.00	917.00	1007.00	10	230	103.48	1025.68

Objective function: 186.11.

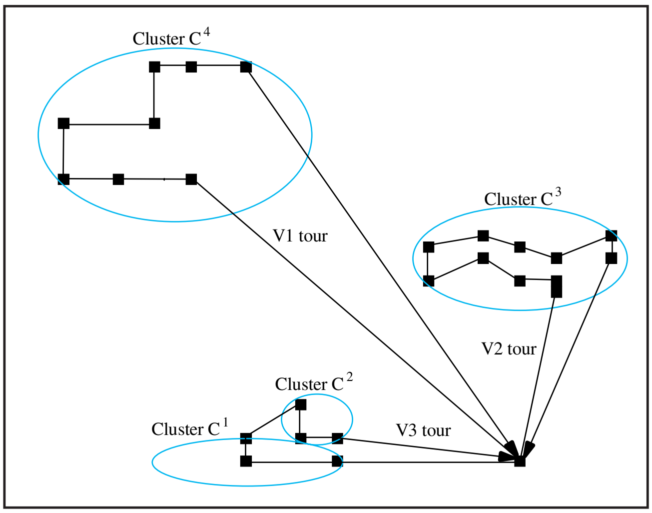

Fig. 4a. Best solution found for example C-101(25 nodes).

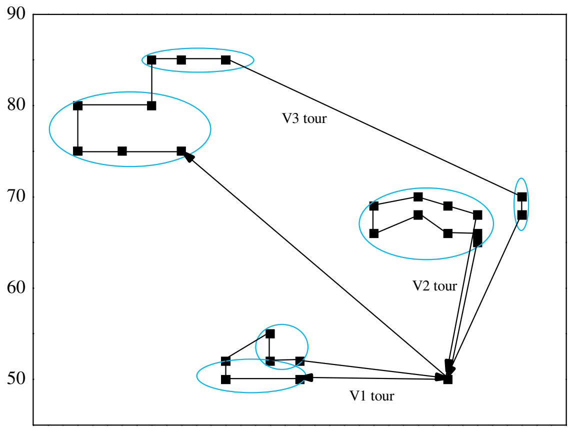

Fig. 4b. Best solution found for the heterogeneous-fleet C-101(25) variant.

Model sizes and CPU times for the problems solved at Phases II and III are also reported in Table 2. The three vehicle scheduling problems at Phase III were all solved at once. Compared with the rigorous approach, the number of binary variables has been cut by three and the CPU time sharply drops from 43.94 seconds to less than one second. Time requirements from the clustering procedure have been neglected. Next, the proposed hybrid approach was applied to the three variants of Example C101(25). The optimal solutions depicted in Figs. 4b, 4c and 4d, respectively, were again found but at much lower computational cost. Table 2 shows that the solution time for the two-depot/homogeneous fleet C101(25) variant decreases from 1451.41 seconds (required by the exact approach) to merely 0.67 seconds. Saving in CPU time is much larger for the two-depot/heterogeneous fleet variant since it drops from 2924.4 seconds to 0.56 seconds. As mentioned before, vehicle-scheduling problems at Phase III were all solved at once since the values of variables Y_{iv} and X_{vp} have already been set in Phase II and, in addition, the exact elimination rules fixed the values of all S_{ij}-variables. Consequently, the MILP formulation tackled in Phase III becomes an LP.

4.2. Example C102(25)

Benchmark problems C101(25) and C102(25) both comprise the same set of nodes and locations but a different time-window distribution. Application of the clustering procedure to problem C-102(25) yielded 5 clusters (see Table 6). However, it is observed that the merging of clusters $C^4 \cup C^5$ generated for Example C102(25) is equivalent to the single cluster C^4 of problem C-101(25). The other three clusters (C^1, C^2, C^3) remain

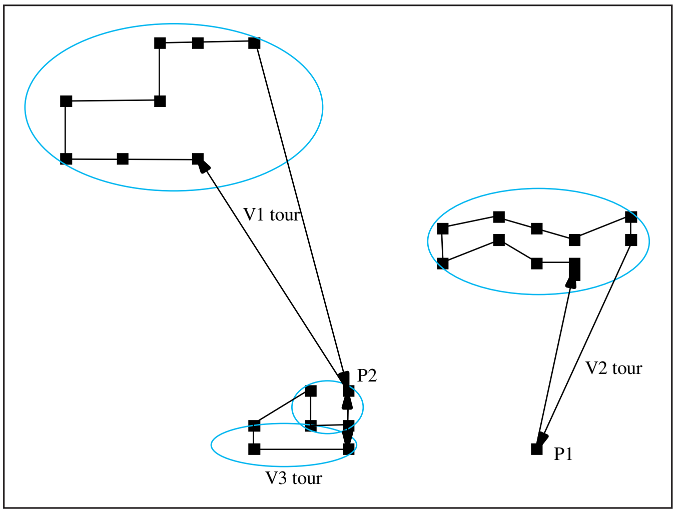

Fig. 4c. Best solution found for example C-101(25 nodes) with 2 depots and a homogeneous fleet.

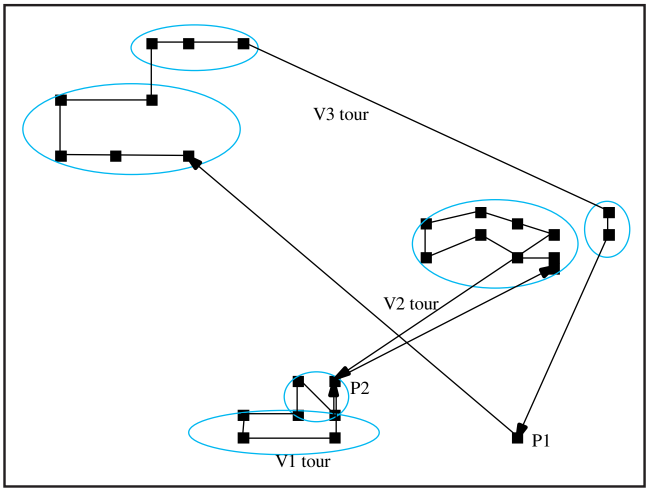

Fig. 4d. Best solution found for the two-depot heterogeneous fleet C-101(25) variant.

unchanged but not necessarily with the same ordering of nodes in the corresponding entry list. In short, the change in the time-window distribution led to the split of a cluster into a pair of new ones and a different ordering of nodes in the entry list of cluster C^1.

In Phase II, the three available vehicles are used to service the five clusters, with vehicle V1 successively visiting clusters C^5 and C^4 in this order. As a result, a cluster-based tour schedule similar to that found for Example C101(25) was discovered (see Table 7). Finally, Phase III provides the optimal node-based tour schedules that are summarized in Table 8 and Fig. 5. The only difference between the optimal tour schedules for Examples C101(25) and C102(25) is the reordering of nodes on the tour assigned to vehicle V2 visiting cluster C^3 only. The analysis of results indicates that: (a) Phase II again provides an accurate estimation of the travel time required by each vehicle to complete the tour, and (b) the optimal solution is still found even if the ordering of clusters on each tour found in Phase II is enforced during the execution of Phase III. In this way, sizable savings in binary variables, constraints and CPU time are simultaneously achieved at no additional traveling expenses. In other words, aggregated tour images are generated during Phases I and II. Though somewhat simpler than the real route topology, they still capture their fundamental structures.

Table 4
Clusters for example C-101(25) generated at Phase I

	C^1	C^2	C^3	C^4
Assigned nodes	T20 T24 T25	T23 T22 T21	T5 T3 T7 T8 T10 T11 T9 T6 T4 T2 T1	T13 T17 T18 T19 T15 T16 T14 T12
Cluster load	60	50	160	190
Coordinates				
X	27	29	40	20
Y	51	53	68	80
Time window				
aC	10	732	15	30
bC	73	777	67	92
Cluster service time	278	276	1029	764
Traveled-distance	8	6	39	44

Table 5
Cluster-based tour schedules for example C-101(25 nodes)

Vehicle	Cluster	Waiting time	Arrival time	Departure time	Cluster load	Vehicle load	Traveled distance	Travel time
V1	C^4	0.00	30.81	794.81	190	190	105.62	825.62
V2	C^3	0.00	15.13	982.13	160	160	69.26	997.26
V3	C^1	0.00	10.00	288.00	60			
	C^2	441.14	732.00	1008.00	50	110	41.26	1019.40

Objective function: 216.14 (includes intra-cluster traveled distances).

4.2.1. Numerical results for large homogeneous VRPTW examples

Numerical results for a large number of Solomon test problems of different sizes; node geographical densities and time-window distributions are presented in Tables 9–15. Optimal values for them reported in Tables 9 and 12 were extracted from Kallehauge et al. (2001). The maximum allowed distance between intra-cluster nodes was $d^{\max} = 10$ for "C" and "RC" problems and $d^{\max} = 12$ for "R" problems. Since the geographical locations of nodes in "R" problems are randomly generated based on a uniform distribution, it was necessary a larger $d^{\max}$ to still get a low number of clusters during Phase I. From the numerical experiments, it can be drawn the following conclusions:

(1) As expected, the method was especially successful for clustered examples solving many of them to optimality. Table 9 shows the number of clusters generated during Phase I for nine benchmark problems of class C when the first 25, 50 or the whole set of

Table 6
Clusters generated for example C-102(25 nodes)

	C^1	C^2	C^3	C^4	C^5
Assigned nodes	T20 T24 T25	T23 T22 T21	T1 T2 T3 T5 T7 T8 T10 T11 T9 T6 T4	T12 T15 T16 T14	T13 T17 T18 T19
Cluster Load	60	50	160	110	80
Coordinates					
X	27	29	40	22	18
Y	51	53	68	84	76
Time window					
aC	10	732	0	0	30
bC	73	777	324	429	92
Cluster service time	278	276	1029	659	375
Traveled distance	8	6	39	24	15

Table 7
Cluster-based tour schedules for example C-102(25 nodes)

Vehicle	Cluster	Waiting time	Arrival time	Departure time	Cluster load	Vehicle load	Traveled distance	Travel time
V1	C^5	0.00	30.81	405.81	80			
	C^4	0.00	414.75	1073.75	110	190	117.20	1112.20
V2	C^3	0.00	16.00	1045.00	160	160	71.00	1061.00
V3	C^1	0.00	10	288.00	60			
	C^2	441.14	732.00	1008	50	110	41.26	1019.40

Objective function: 229.46 (includes intra-cluster traveled distances).

Table 8
Optimal tour schedules for example C-102(25 nodes)

Vehicle	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	T13	0.00	30.81	120.81	30			
	T17	0.00	124.81	214.81	20			
	T18	0.00	217.81	307.81	20			
	T19	0.00	312.81	402.81	10			
	T15	0.00	407.81	497.81	40			
	T16	0.00	502.81	592.81	40			
	T14	0.00	594.81	684.81	10			
	T12	0.00	687.81	777.81	20	190	95.88	815.88
V2	T7	0.00	16.00	106.00	20			
	T8	0.00	108.83	198.83	20			
	T10	6.39	210.83	300.83	10			
	T11	0.00	303.83	393.83	10			
	T9	32.16	429.15	519.15	10			
	T6	0.00	521.38	611.38	20			
	T4	0.00	613.62	703.62	10			
	T2	0.00	707.22	797.22	30			
	T1	0.00	799.22	889.22	10			
	T3	0.00	892.83	982.83	10			
	T5	0.00	983.83	1073.83	10	160	58.38	1048.38
V3	T20	0.00	10.00	100.00	10			
	T24	0.00	105.00	195.00	10			
	T25	0.00	197.00	314.00	40			
	T23	440.76	732.00	822.00	10			
	T22	0.00	825.00	973.00	20			
	T21	0.00	917.00	1225.00	20	110	36.44	1017.20

Objective function: 190.70.

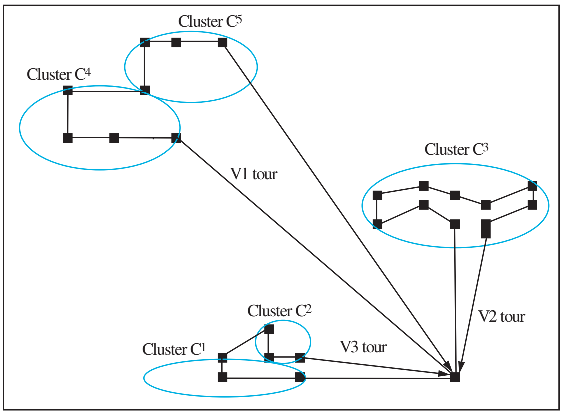

Fig. 5. Best solution found for example C-102(25 nodes).

100 nodes were considered. In all cases except one the optimal solution was found at low computation cost. Moreover, an improved solution for Problem C205(50 nodes) has been discovered (see Table 10 and Fig. 6). Fig. 7 presents the best solution found for Problem C-102 involving 100 nodes.

(2) Phase II is usually the "bottleneck" step of the proposed approach mostly determining the CPU time requirements. Model sizes and solution times for C-class problems solved in the paper are shown in Table 11. It can be seen that Phase-II solution time ranges from 21.64 seconds to 117.2 seconds for 100-node problems of type C1, rising to 1960 seconds for Solomon's problem C-201(100 nodes).
(3) The clustering strategy was also very effective in dealing with RC class problems but its efficacy decreases when R-class problems are tackled. Numerical results for several RC and R benchmark problems involving up to

Table 9
Results found for different C-class benchmark problems

Problem	Nodes	Clusters	Vehicles	This approach	Best solution reported	Subopt. gap (%)
C-101	25	4	3	191.80	191.80	0.00
	50	7	5	363.23	363.23	0.00
	100	10	10	828.90	828.90	0.00
C-101m[a]	100	10	10	784.90	–	–
C-102	25	7	3	190.70	190.70	0.00
	50	12	5	373.68	362.17	3.17
	100	20	10	828.90	828.90	0.00
C-103	25	7	3	191.80	191.80	0.00
C-105	25	4	3	191.80	191.80	0.00
	50	6	5	363.10	363.10	0.00
	100	10	10	828.90	828.90	0.00
C-106	25	4	3	191.80	191.80	0.00
	50	5	5	363.23	363.23	0.00
	100	10	10	828.90	828.90	0.00
C-107	25	4	3	191.80	191.80	0.00
	50	6	5	363.23	363.23	0.00
	100	10	10	828.90	828.90	0.00
C-108	25	4	3	191.80	191.80	0.00
	50	5	5	363.23	363.23	0.00
	100	10	10	828.90	828.90	0.00
C-109	25	4	3	191.80	191.80	0.00
	50	5	5	363.23	363.23	0.00
	100	10	10	828.90	828.90	0.00
C-201	25	6	2	215.55	215.55	0.00
	50	9	3	391.55	361.80	0.00
	100	7	3	591.10	591.10	0.00
C-202	25	29	2	215.55	215.55	0.00
C-203	25	9	2	215.55	215.55	0.00
C-205	25	5	2	215.55	215.55	0.00
	50	8	3	361.80	363.50[b]	0.00
	100	7	3	588.40	588.40	0.00
C206	25	5	2	215.55	215.55	0.00
C-207	25	8	2	215.55	215.55	0.00
C-208	25	5	2	215.55	215.55	0.00

[a] Two-depot heterogeneous-fleet C101(100) variant.
[b] A new best solution for problem C205(50 nodes).

50 nodes are shown in Table 12. Optimal and near-optimal solutions were often discovered when tackling RC-problems. Model sizes and computational requirements are listed in Table 13. Moreover, a new best solution was found for problem RC-101(25 nodes) that is shown in Table 14 and Fig. 8. Note that the CPU time required to solve the cluster-based MILP formulation shows a sizable increase for RC-problems of 50 nodes.

(4) Despite the good performance of the clustering algorithm for RC-problems, some difficulty arises when the number of nodes is 50 or 100. Combination of tight and randomly distributed time-windows together with approximated intra-cluster travel time forces using more than one vehicle to service some clusters. This is the case for benchmark problems RC-102 and RC-103. To avoid such infeasibilities during Phase III, $d^{\max}$ is further reduced and, consequently, the number of clusters rises. As a result, the computational efficiency and the solution quality both become worse. Difficulties with RC-problems with 50 or more nodes can be overcome by allowing small time-window violations at

Table 10
New best solution for example C-205(50 nodes)

Vehicle	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	T5	0.00	15.13	105.13	10			
	T2	22.04	133.00	223.00	30			
	T1	0.00	231.60	321.60	10			
	T7	663.59	993.00	1083.00	20			
	T3	0.00	1087.12	1177.12	10			
	T4	0.28	1181.00	1271.00	10	90	70.71	1296.62
V2	T49	855.95	887.00	977.00	10			
	T40	749.84	1739.00	1829.00	10			
	T44	0.00	1833.47	1923.47	10			
	T46	0.15	1929.00	2019.00	30			
	T45	0.00	2025.40	2115.40	10			
	T50	92.13	2212.00	2302.00	10			
	T47	93.00	2400.00	2490.00	10			
	T43	0.00	2493.00	2583.00	10			
	T42	0.00	2586.00	2676.00	20			
	T41	0.00	2678.00	2768.00	10			
	T48	0.00	2776.06	2866.06	10	140	95.78	2876.84
V3	T20	0.00	10.00	100.00	10			
	T22	0.00	102.82	192.82	20			
	T24	159.02	355.44	445.44	10			
	T27	0.00	448.27	538.27	10			
	T30	0.00	542.51	632.51	10			
	T29	0.00	637.51	727.51	10			
	T6	0.00	736.45	826.45	20			
	T32	0.00	832.78	922.78	30			
	T33	0.00	924.78	1014.78	40			
	T31	0.00	1020.16	1110.16	20			
	T35	0.00	1115.16	1205.16	10			
	T37	0.00	1211.00	1301.00	20			
	T38	0.00	1303.00	1393.00	30			
	T39	0.00	1398.00	1488.00	20			
	T36	0.00	1493.00	1583.00	10			
	T34	0.00	1586.00	1676.00	20			
	T28	0.00	1686.77	1776.77	20			
	T26	0.00	1784.83	1874.83	10			
	T23	0.00	1882.04	1972.04	10			
	T18	0.00	1981.09	2071.09	20			
	T19	0.00	2076.09	2166.09	10			
	T16	0.00	2173.17	2263.17	40			
	T14	0.00	2265.17	2355.17	10			
	T12	0.00	2358.17	2448.17	20			
	T15	0.00	2455.24	2545.24	40			
	T17	0.00	2550.63	2640.63	20			
	T13	0.00	2644.63	2734.63	30			
	T25	0.00	2743.63	2833.63	40			
	T9	0.00	2840.84	2930.84	10			
	T11	0.00	2937.91	3027.91	10			
	T10	0.00	3030.91	3120.91	10			
	T8	0.00	3127.00	3217.00	20			
	T21	0.00	3222.65	3312.65	20	630	195.30	3324.32

Objective function: 361.80.

Table 11
Model sizes and solution times for C-class benchmark problems

Problem	Nodes	Phase II				Phase III			
		Binary variables	Continuous variables	Constraints	CPU time (seconds)	Binary variables	Continuous variables	Constraints	CPU time (seconds)
C-101	25	18	14	63	0.60	40	23	257	3.79
	50	36	24	236	1.76	1	23	177	1.38
	100	100	40	610	74.15	45	27	246	1.42
C-101m[a]	100	103	40	610	2271.00	45	27	246	1.42
C-102	25	29	20	264	0.71	40	23	257	3.79
	50	77	34	1001	2.69	40	23	257	3.79
	100	263	60	5730	21.64	48	27	252	2.25
C-103	25	42	20	339	12.15	52	24	270	0.60
C-105	25	14	14	90	0.54	1	23	179	0.22
	50	35	22	273	1.75	30	23	237	0.44
	100	106	40	1000	100.13	14	26	221	0.33
C-106	25	12	14	69	0.12	12	24	190	0.10
	50	26	20	140	0.60	12	24	190	0.10
	100	103	40	840	81.00	19	28	260	0.60
C-107	25	14	14	90	0.45	13	24	192	0.12
	50	35	22	273	1.75	13	24	192	0.12
	100	115	40	1310	117.20	16	28	254	0.60
C-108	25	16	14	102	0.55	24	24	214	0.60
	50	28	20	180	0.69	24	24	214	0.60
	100	121	40	1480	91.30	31	28	284	0.55
C-109	25	17	14	114	0.60	34	24	234	2.53
	50	28	20	180	0.55	34	24	234	2.53
	100	115	40	1310	117.20	43	28	308	1.61
C-201	25	15	16	132	0.39	22	40	482	0.76
	50	43	24	119	17.47	28	56	894	2.53
	100	27	20	240	1960.00	145	68	1512	4.83
C-202	25	32	22	295	4.67	58	40	554	4.28
C-203	25	36	22	313	1.70	136	40	710	0.94
C-205	25	15	14	113	0.99	50	40	538	0.17
	50	41	22	375	6.37	100	68	1422	2.14
	100	27	20	240	23.18	100	70	1493	2.63
C-206	25	16	14	117	0.55	68	40	574	0.49
C-207	25	23	20	222	0.60	119	40	676	3.79
C-208	25	16	14	117	0.55	87	40	612	3.80

[a] Two-depot heterogeneous-fleet C101(100) variant.

some penalty costs during Phase II. In this way, the sub-optimal gap reported in Table 12 for benchmark problems RC-102(50 nodes) and RC-108(50 nodes) can be reduced by half and the solution times experience a large decrease.

(5) Despite using a larger $d^{\max}$, the number of clusters generated for R-class problems is considerably higher than that for C and RC-problems and so are the solution time requirements (see Tables 11 and 13). However, near optimal solutions were sometimes identified. The best solution found for problem R-112(25 nodes) is described in Table 15 and Fig. 9.

(6) The computational performance improves with tight time windows and high node geographical density. Due to the use of pruning rules of Section 2.3, the critical size of the cluster-based MILP formulation significantly decreases and the hybrid approach becomes much more efficient.

Table 12
Numerical results found for RC-class and R-class benchmark problems

Problem	Nodes	Clusters	Vehicles	This approach solution	Optimal solution	Subopt. gap (%)
RC-101	25	6	4	454.19	461.10[a]	0.00%
RC-102	25	12	3	348.13	348.13	0.00
	50	15	7	1004.92	822.50	22.17
RC-103	25	12	3	334.00	334.00	0.00
	50	17	6	928.39	710.90	30.60
RC-104	25	8	3	307.02	307.02	0.00
RC-105	25	11	4	416.88	412.30	1.11
RC-106	25	5	3	346.50	346.50	0.00
RC-107	25	6	3	298.95	298.30	0.00
RC-108	25	5	3	294.98	294.50	0.00
	50	9	6	716.55	598.10	19.80
RC-201	25	6	3	361.24	361.24	0.00
R-101	25	19	8	693.11	619.17	11.94
R-102	25	19	7	623.19	547.90	14.89
R-103	25	18	5	478.50	454.60	5.25
R-104	25	13	4	430.11	418.00	2.90
R-107	25	16	4	483.17	425.30	13.60
R-111	25	10	4	517.27	429.90	19.39
R-112	25	9	4	428.90	394.00	8.86

[a] A new best solution for problem RC-101(25 nodes).

Table 13
Problem sizes and CPU times for RC-class and R-class examples

Problem	Nodes	Phase II				Phase III			
		Binary variables	Continuous variables	Constraints	Solution time	Binary variables	Continuous variables	Constraints	Solution time
RC-101	25	29	20	222	0.38	6	19	129	0.06
RC-102	25	70	32	708	0.05	11	19	139	0.11
	50	143	44	2103	1977.86	47	22	235	6.87
RC-103	25	71	60	675	1.07	23	19	163	0.88
	50	173	46	2661	2976.96	51	22	243	38.15
RC-104	25	39	22	324	0.33	41	20	200	2.53
RC-105	25	60	60	629	2.91	31	18	159	0.11
RC-106	25	19	16	129	0.88	24	18	145	0.88
RC-107	25	20	18	150	0.17	46	20	210	0.22
RC-108	25	20	16	135	0.22	46	20	210	0.33
	50	72	30	729	41.69	56	22	253	20.43
RC-201	25	27	18	163	0.44	7	19	131	0.05
R-101	25	156	54	2553	751.00	2	9	45	0.16
R-102	25	178	52	2892	600.23	5	9	47	0.05
R-103	25	145	46	2014	180.29	7	13	81	0.06
R-104	25	92	34	1007	1800.40	21	15	127	0.11
R-107	25	114	40	1356	190.86	19	15	123	0.28
R-111	25	58	28	566	7.25	18	14	97	1.31
R-112	25	49	26	447	11.04	29	16	136	0.28

Table 14
New best solution found for example RC-101(25 nodes)

Vehicle	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	T23	19.96	65.00	75.00	30			
	T21	0.00	77.00	87.00	10			
	T19	0.00	92.38	102.38	40			
	T18	0.00	107.77	117.77	20			
	T20	0.00	127.97	137.97	10			
	T25	3.82	154.00	164.00	20			
	T24	0.00	174.44	184.44	10	140	125.72	219.50
V2	T14	0.00	35.35	45.35	10			
	T12	0.00	48.35	58.35	40			
	T11	0.00	63.35	73.35	20			
	T15	0.00	79.35	89.35	20			
	T16	0.00	91.35	101.35	20			
	T9	0.00	112.53	122.53	20			
	T10	0.00	127.53	137.53	30			
	T13	0.00	144.61	154.61	10			
	T17	0.00	165.79	175.79	20	190	126.10	216.10
V3	T22	57.00	92.00	102.00	40	40	70.00	137.00
V4	T5	0.69	41.00	51.00	20			
	T2	0.00	61.20	71.20	30			
	T7	0.80	79.00	89.00	20			
	T8	0.00	94.00	104.00	10			
	T6	0.00	109.83	119.83	20			
	T3	0.00	130.60	140.77	10			
	T1	0.00	143.77	153.77	40			
	T4	0.00	160.14	170.14	20	170	132.39	212.39

Objective function: 454.19.

4.2.2. *A large multi-depot heterogeneous fleet VRPTW example*

In order to illustrate the use of the cluster-based approach on a large multi-depot heterogeneous fleet VRPTW case study, the benchmark problem C-101 has been modified by including a new depot (Euclidean coordinates, $X = 30$, $Y = 55$) and a heterogeneous fleet composed by 3 vehicles of 240 units capacity (V1, V2 and V3), 3 vehicles of 200 units capacity (V4, V5 and V6) and 4 vehicles of 170 units capacity (V7–V10). However, the total number of vehicles and the overall fleet capacity remain unchanged. Phase I generated the clusters to be assigned to tours. Phase II assigns clusters to vehicles in such a way that "big" trucks service clusters with large loads and "small" trucks visit clusters with small loads. The reallocation of some vehicles to the "new" depot allows an additional reduction in the traveled distance from 828.9 to 784.9 units (see Table 9). This is so because vehicles depart from depots closer to the clusters being serviced. The solution found is depicted in Fig. 10. Three of the vehicles have been assigned to the new depot P2. Model sizes and solution times are reported in Table 11. Though just three more binary variables are to be considered, the CPU time grows from 74.15 seconds for the single-depot/homogeneous fleet C101(100) to 2271 seconds for the two-depot/heterogeneous fleet variant. In turn, the solution time for the largest tour scheduling problem tackled in Phase III is 1.42 seconds.

5. Conclusions

This work introduces a new cluster-based hierarchical hybrid approach for the multi-depot heterogeneous-fleet VRPTW problem. The approach

Table 15
Best solution found for example R-112 (25 nodes)

Vehicle	Node	Waiting time	Arrival time	Departure time	Node load	Vehicle load	Traveled distance	Travel time
V1	T7	14.39	36.00	46.00	5			
	T19	0.00	57.18	67.18	17			
	T11	0.00	74.25	84.25	12			
	T10	0.00	95.43	105.43	16			
	T20	0.00	121.24	131.24	9			
	T9	0.00	142.42	152.42	16			
	T1	0.00	170.22	180.22	10	85	110.67	195.46
V2	T12	0.00	15.00	25.00	19			
	T25	9.59	56.00	66.00	3			
	T24	0.00	81.00	91.00	3			
	T3	0.00	105.14	115.14	13	41	87.71	137.50
V3	T5	8.39	29.00	39.00	26			
	T6	0.00	49.00	59.00	3			
	T18	0.00	70.18	80.18	12			
	T8	0.00	90.62	100.62	9			
	T17	0.00	114.54	124.54	2			
	T16	3.28	139.00	149.00	19			
	T14	26.82	187.00	197.00	20	91	120.54	229.01
V4	T2	0.00	18.00	28.00	7			
	T15	0.00	41.00	51.00	8			
	T22	0.00	66.81	76.81	18			
	T23	0.00	87.99	97.99	29			
	T4	0.00	112.99	122.99	19			
	T21	0.00	132.99	142.99	11			
	T13	0.00	158.80	168.80	23	115	109.98	179.98

Objective function: 428.91.

aims to integrate a heuristic clustering algorithm into an optimization framework. To this purpose, an MILP mathematical formulation for this general VRPTW type was first developed. Likewise

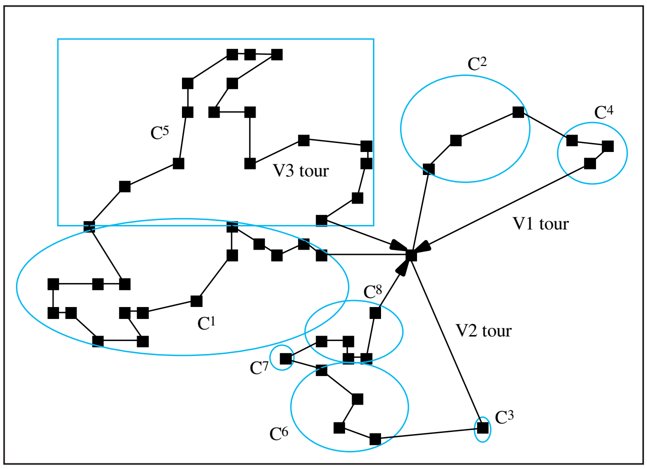

Fig. 6. Best solution found for problem C-205(50 nodes).

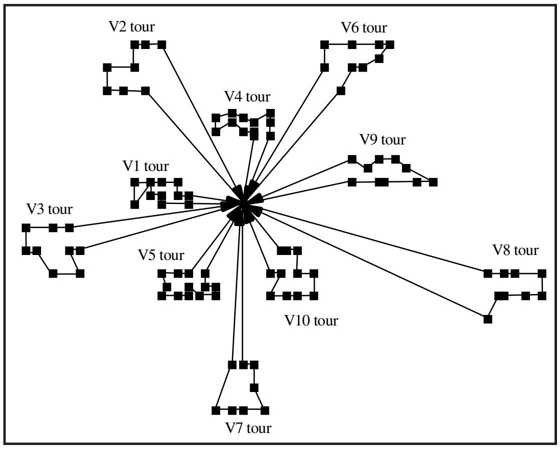

Fig. 7. Best solution found for example C-102(100 nodes).

pure optimization methods, however, the proposed exact approach is able to solve problems involving up to 25 nodes to optimality since the model size and the solution time both grow exponentially with the number of nodes and vehicles. On the other hand, heuristics can approximately solve problems of larger sizes in much less computational time but usually lack robustness and have a problem-dependent performance. A heuristic/algorithmic mixed strategy arises as another option often sharing the attributes of both techniques. By incorporating a preprocessing stage to gather nodes into a few clusters, an efficient, compact VRPTW mathematical model at the level of clusters can be derived. The cluster-based solution is subsequently processed by a small MILP model to provide detailed vehicle routes and schedules through disaggregating clusters into the original nodes. The proposed clustering algorithm that exploits time-window constraints to generate feasible clusters seems to be work well even for R-class problems. The three-phase hybrid approach is as robust as the optimization methods and capable of solving problems with 100 nodes at reasonable solution time. Numerical results indicate that the cluster-based optimization method proved to be quite successful on a variety of Solomon's single-depot homogeneous-fleet benchmark problems and new multi-depot heterogeneous fleet VRPTW instances introduced in this paper. Optimal or near optimal solutions were obtained for a significant number of C-class problems of different sizes. For RC and R-class problems, the sub-optimal gap increases but it remains within acceptable limits.

Future research lines include the development of a mathematical framework for further improving the solution provided by the three-phase hybrid approach and the extension of the strategy to more difficult problems such as the pick-up and delivery problem with time windows (PDPTW).

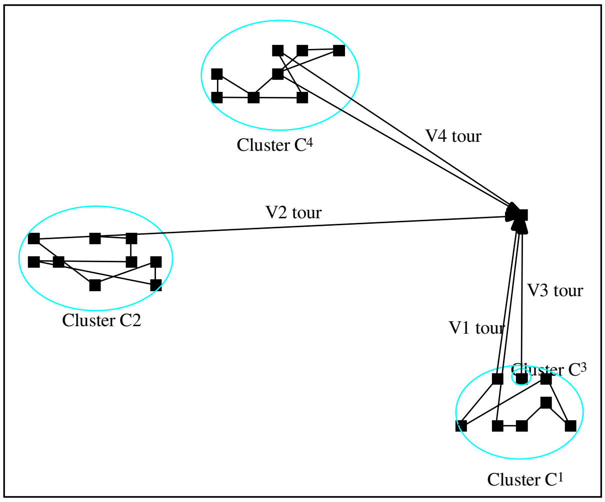

Fig. 8. New best solution found for problem RC-101(25 nodes).

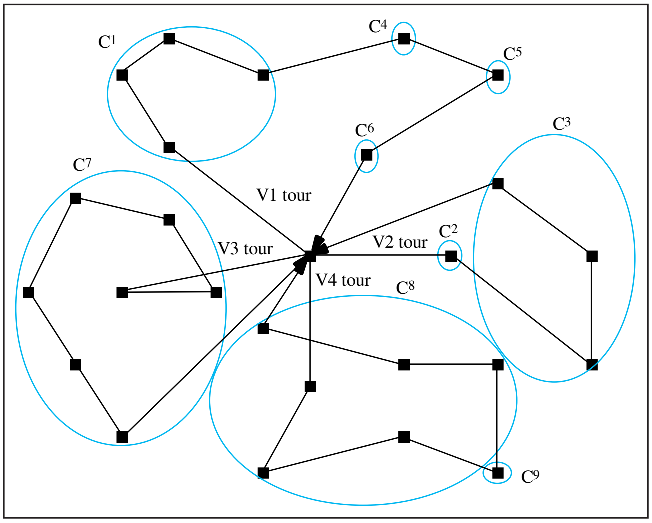

Fig. 9. Best solution found for problem R-112(25 nodes).

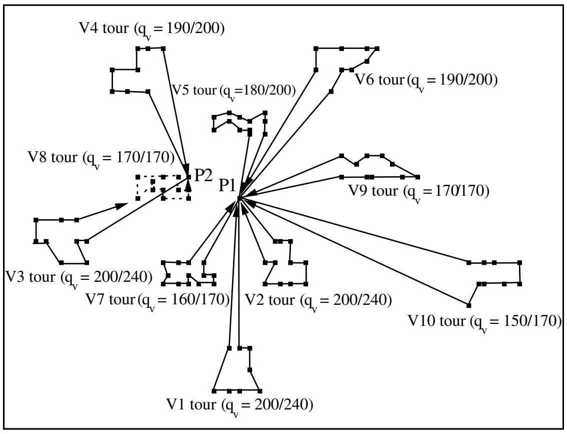

Fig. 10. Best solution found for the two-depot heterogeneous fleet C-101(100) variant.

References

Bowerman, R., Calamiand, P., Brent Hall, G., 1994. The spacefilling curve with optimal partitioning heuristic for the vehicle routing problem. European Journal of Operational Research 76, 128–142.

Bramel, J., Simchi-Levi, D., 1997. The logic of logistics. Theory, algorithms and applications for logistic management. Springer Series in Operations Research.

Christofides, N., Mingozzi, A., Toth, P., 1981. Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxations. Mathematical Programming 20, 255–282.

Crainic, T.G., Laporte, G., 1998. Fleet Management and Logistics. Kluwer Academic Publishers.

Desrochers, M., Desrosiers, J., Solomon, M.A., 1992. A new optimization algorithm for vehicle routing problem with time windows. Operations Research 40, 342–354.

Desrosiers, J., Sauvé, M., Soumis, F., 1988. Lagrangian relaxation methods for solving the minimum fleet size multiple traveling salesman problem with time windows. Management Science 34, 1005–1022.

Desrosiers, J., Dumas, Y., Solomon, M., Soumis, F., 1995. Time constrained routing and scheduling. In: Ball, M. et al. (Eds.), Handbook in OR & MS, vol. 8. Elsevier Science, pp. 35–139.

Fisher, M., 1994. Optimal solution of vehicle routing problems using minimum K-trees. Operations Research 42 (4), 626–642.

Fisher, M., 1995. Vehicle routing. In: Ball, M. et al. (Eds.), Handbook in OR & MS, vol. 8. Elsevier Science, pp. 1–33.

Fisher, M., Jornsten, K., Madsen, O., 1997. Vehicle routing with time windows: Two optimization algorithms. Operations Research 45, 488–495.

Gendrau, M., Laporte, G., Potvin, J.Y., 1997. Vehicle routing: Modern heuristics. In: Local Search in Combinatorial Optimization. John Wiley and Sons, pp. 311–336.

Halse, K., 1992. Modeling and solving complex vehicle routing problems. Ph.D. Dissertation no. 60, IMSOR, Technical University of Denmark.

Golden, B., Wasil, E.A., Kelly, J.P., Chao, I.M., 1998. Metaheuristics in vehicle routing, fleet management and logistics, pp. 33–56. Edited by T.G. Crainic and G. Laporte.

Jornsten, K., Madsen, O., Sorensen, B., 1986. Exact solution of the vehicle routing and scheduling problem with time windows by variable splitting. Research Report 5/86, Technical University of Denmark.

Kallehauge, B., Larsen, J., Madsen, O., 2001. Lagrangian duality applied on vehicle routing with time windows. Experimental results. Technical Report IMM-TR-2001-9, IMM.

Kolen, A., Rinnooy, A., Trienekens, H., 1987. Vehicle routing with time windows. Operations Research 35, 266–273.

Laporte, G., 1992. The vehicle routing problem: An overview of exact and approximated algorithms. European Journal of Operational Research 59, 345–358.

Savelsbergh, M., 1985. Local search in routing problems with time windows. Annals of Operations Research 4, 285–305.

Solomon, M., 1987. Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations Research 32, 254–265.