Computers and Chemical Engineering 33 (2009) 513–530

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A hybrid local improvement algorithm for large-scale multi-depot vehicle routing problems with time windows

Rodolfo G. Dondo, Jaime Cerdá*

INTEC (Universidad Nacional del Litoral – CONICET) Güemes 3450, (3000) Santa Fe, Argentina

ARTICLE INFO

Article history:
Received 6 March 2006
Received in revised form
10 September 2008
Accepted 5 October 2008
Available online 18 October 2008

Keywords:
Supply chain management
Vehicle routing
Improvement algorithm
MILP-formulations

ABSTRACT

One of the major research topics in the supply chain management field is the multi-depot vehicle routing problem with time windows (m-VRPTW). It aims to designing a set of minimum-cost routes for a vehicle fleet servicing many customers with known demands and predefined time windows. This paper presents an m-VRPTW local search improvement algorithm that explores a large neighborhood of the current solution to discover a cheaper set of feasible routes. The neighborhood structure comprises all solutions that can be generated by iteratively performing node exchanges among nearby trips followed by a node reordering on every route. Manageable mixed-integer linear programming (MILP) formulations for both algorithmic steps were developed. To further reduce the problem size, a spatial decomposition scheme has also been applied. A significant number of large-scale benchmark problems, some of them including up to 200 customers, multiple depots and different vehicle-types, were solved in quite reasonable CPU times.

1. Introduction

The supply chain management problem covers the entire process of moving raw materials and input requirements from suppliers to plants, converting them into products at processing plants, shipping the finished products to several warehouses, and subsequently delivering them to the final customers. Effective supply chain management (SCM) should deal with a variety of decisions at the strategic, tactical and operational levels. Structural decisions concerning the location of facilities, like production plants or warehouses, may be viewed as strategic, while the problem of determining the fleet size and the vehicle-type mix could be regarded as tactical. At the operational level, the routing and scheduling of vehicles/vessels and the crew assignment are to be selected on a daily basis. Logistic costs for the chemical industry are often considerable and sometimes as large as 20% of the purchasing costs (Jetlund & Karimi, 2004). Moreover, logistic expenses directly related to transport operations and crew salaries constitute a large portion of the total distribution cost. Consequently, better routes and schedules for vehicles/vessels transporting products from production sites to depots and from depots to clients may result in substantial cost savings over a number of years. As a result, logistics has become an issue of increasing importance to the chemical industry in reducing costs. Typical SCM-industrial cases at the operational level reported in the literature involve the delivery of refined products from depots to gasoline stations through using a fleet of multi-parcel trucks (Van der Bruggen, Gruson, & Salomon, 1995), the scheduling of multi-compartment chemical carriers transporting chemicals between upstream and downstream refineries and manufacturers (Jetlund & Karimi, 2004), the dispatching of either crude oil to refineries or refined products to industrial users through tankers and barges (Fagerholt, 2004), and the fresh milk transportation from hundreds of dairy farms to processing plants and the delivery of dairy products from the central warehouse to retail outlets (Tarantilis & Kiranoudis, 2007).

Practical approaches to the vehicle routing problem (VRP) mostly apply heuristic approximate algorithms providing good solutions within a reasonable computer time. Heuristic methods can be classified into three broad categories: tour-construction procedures, tour-improvement procedures and composite procedures (Bodin, Golden, Assad, & Ball, 1983). Tour construction procedures generate a good set of vehicle routes from the node-to-node distance matrix. Route improvement procedures generally start with a non-optimal set of feasible tours and seek out a better solution by performing local perturbations on the initial routes. Almost all VRP improvement algorithms iteratively use some version of a local search method to obtain a new set of lower-cost, feasible vehicle routes. They range from simple improvement heuristics to modern metaheuristics like simulated annealing, tabu search and genetic algorithms (Papadimitrou & Steiglitz, 1982; Gendreau, Laporte, & Potvin, 1994). Routing improvement procedures generally assume symmetric traveling costs ($c_{ij} = c_{ji}$). On the other hand, composite

* Corresponding author. Tel.: +54 342 4559174/77; fax: +54 342 4550944.
E-mail address: jcerda@intec.unl.edu.ar (J. Cerdá).

0098-1354/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2008.10.003

or two-phase algorithms first construct an initial solution and then attempt to improve it using one or several tour improvement techniques. The other alternative to deal with vehicle routing problems is the use of exact approaches based on mixed-integer linear (MILP) or mixed-integer non-linear (MINLP) mathematical formulations (Ball, Magnanti, Monma, & Nemhauser, 1995). Most of the exact approaches can be regarded as route construction methods.

One of the most important features of a VRP improvement algorithm is the local search technique that it applies. A local search algorithm is basically a neighborhood search procedure that analyzes a set of feasible points within the neighborhood of the current solution and finds a better neighbor. Though the definition of the neighborhood structure varies with the local search approach, the set of neighbors is generally obtained from the current solution by doing a limited number of moves. Different types of improvement moves can be applied like string exchange, string relocation and string cross. String exchange means the exchange of two strings with at most k nodes within the same route or between neighboring routes. String relocation consists of moving a string of at most k nodes (with $k = 1–2$) from one to another route. String cross is performed when two strings of vertices on two different routes are exchanged by crossing two edges of such routes. Therefore, the improvement procedure may operate on either an individual route or several routes at a time. A critical issue is the choice of the neighborhood structure, i.e. the way in which the neighborhood around the current solution is defined. Generally, the larger the neighborhood, the better is the quality of the best neighbor and the higher is the likelihood of converging to the truly optimal solution. At the same time, the larger the neighborhood, the longer the time it takes to search the neighborhood at each iteration. Therefore, a large neighborhood is not always the best option unless it is explored in a very efficient manner (Ahuja, Ergun, Orlin, & Punnen, 2002). Sometimes, the neighborhood is generated by weakly modifying the current solution in only two or three edges per iteration. As a result, a local optimum that is very dependent on the initial solution is found and a little improvement is generally achieved. On the contrary, this paper is especially interested on local search algorithms that efficiently explore a large neighborhood so as to provide a set of feasible tours much cheaper than the starting routes.

Interesting surveys on classical and modern heuristic-based improvement methods for the vehicle routing problem with time windows (VRPTW) can be found in Laporte, Gendreau, Potvin, and Semet (2000) and Ahuja et al. (2002). The neighborhood structure for the classical interchange procedure proposed for the traveling salesman problem (TSP) can be built by randomly breaking a trip at k points into k paths and reconnecting them in all possible ways. This is a variable-depth search method known as the k-OPT procedure (Lin & Kernighan, 1973). For $k = 3$, eight different trips can be generated. If a better route among them is identified, it is adopted. If not, the value of k may be increased. The procedure stops at a local minimum when no further improvement can be obtained. Van der Bruggen, Lenstra, and Schuur (1993) developed a local search improvement procedure exploring larger neighborhoods. The neighborhood structure is built by making arc-exchanges on individual trips, i.e. the interchange of k arcs in a route by some other k arcs on the same route. If a route cannot be improved by k-exchange is said to be k-optimal. The most used are 2-exchange, 3-exchange and Or-exchange operations with the later being a special 3-exchange move in which only strings of one, two or three consecutive nodes are relocated on the same route (Or, 1976). On the other hand, cyclic transfer is a neighborhood search technique developed for multi-vehicle routing and scheduling problems (Thompson & Psaraftis, 1993). It improves the set of vehicle routes by exchanging some k nodes among b routes according to a cyclic scheme, i.e. a b-cyclic k-transfer exchange. Usually, it is adopted $b = 2$ and $k = 1–2$. Given some routes ($r1$, $r2$, $r3$), a cyclic transfer consists of moving a certain number of requests k from route $r1$ to route $r2$, a similar one from route $r2$ to $r3$ and, finally, k customers will be moved from route $r3$ to route $r1$. Therefore, the cyclic-transfer neighborhood of a given solution s is the set of feasible routing networks that are reachable from solution s via cyclic transfer. It is said that s is a cyclic-transfer optimum if no member of the cyclic-transfer neighborhood has a better objective function value. Xu and Kelly (1995) used three different types of moves to define the neighborhood structure: swaps of nodes between a pair of routes, a global repositioning of some nodes into other routes and, finally, local route improvements. The node global repositioning is achieved by solving a network flow model to optimally relocate a given number of customers into different routes. Local route reoptimizations are performed by means of 3-opt exchanges and a TSP improvement strategy. Van Breedam (1994) identified a set of four parameters that substantially influence the computational behavior of local improvement procedures: (i) the initial solution, (ii) the type of string moves and the string length k, (iii) the improvement strategy for choosing the next starting solution, i.e. the first-improvement (FI) or the best-improvement (BI) policies, and (iv) the extent of the search procedure within the neighborhood of the current solution. The FI-strategy consists in adopting the first neighboring solution that improves the objective function as the next starting solution while the best-improvement strategy (BI) chooses the best neighbor. Van Breedam concludes that better results are found in terms of both solution quality and computing time by: (1) initiating the search from a good solution, (2) performing string exchanges with a string length k equals 2.

Metaheuristics were proposed in the 1980s in order to tackle combinatorial optimization problems arising in many practical areas. Metaheuristic is an iterative process driven by some subordinate heuristic. Metaheuristic techniques have been able to discover the best solutions for the VRPTW and other difficult routing problems (Osman & Laporte, 1996). However, they usually consume a significant amount of time when compared with other approaches. Metaheuristics include but are not limited to simulated annealing, tabu search, threshold algorithms, neural networks and genetic algorithms. Tabu search algorithms start from an initial solution and move at each iteration from the current solution $\mathbf{x}^k$ to its best neighbor $\mathbf{x}^{k+1}$ until a stopping criterion is satisfied. The value of the objective function at $\mathbf{x}^{k+1}$ is not necessarily less than $f(\mathbf{x}^k)$. To avoid cycling, tabu solutions recently generated are prohibited for a number of iterations. Gendreau, Hertz, and Laporte (1994) introduced the Taburoute algorithm with several innovative features. Initially, several solutions are generated and a limited search within the neighborhood of each one is carried out. The best solution found is then selected as the starting point for the main search. The neighborhood structure is defined by all solutions that can be reached from the starting one by removing a vertex from a particular route, and inserting it into a close route containing one of its m nearest neighbors (Gendreau, Hertz, & Laporte, 1992). Solutions may be infeasible with regards to capacity or maximum route length constraints. To allow this option, the objective function includes two additional terms penalizing overcapacity and tour overduration, respectively. In this way, the likelihood of being trapped in a local minimum is diminished. The basic tabu search procedure can be enhanced by diversification and intensification strategies (Glover & Laguna, 1997). The intensification step consists of reoptimizing the route in which a vertex has just been inserted. In turn, the diversification strategy penalizes vertices that frequently move so as to increase the probability of considering slow moving vertices. In the Tabu Threshold (TT) method, the neighborhood of the current solution is subdivided by grouping the admissible moves into a number of subsets. At each time, one of them is chosen and the best move for the current active subset is performed. As a consequence, the number of moves being considered is reduced and the overall computational effort is decreased. The TT method is based on the alternation of two steps called *improve* and *mixed* phases. The improve phase finds a good local optimum, and the *mixed phase* is aimed at escaping from a local optimum by even moving to either a non-improving or an infeasible solution. The mixed phase is applied when the improve phase fails to get a better solution. Finally, the Taillard's search algorithm (Taillard, 1993) decomposes the main problem into several subproblems by dividing the service area into sectors centered at the central depot and defining concentric regions within each sector. Each subproblem involving the vertices in each mini-sector can be solved independently, but moves of vertices to adjacent sectors are periodically considered. Other metaheuristics for VRPTW can be found in Rochat and Taillard (1995), Taillard, Badeau, Gendreau, Guertin, and Potvin (1997), Gambardella, Taillard, and Agazzi (1999), and Bräysy, Dullaert, & Gendreau, 2004.

Dondo and Cerdá (2007) introduced a new incomplete optimization algorithm that embeds a heuristic procedure into an exact optimization method so as to discovering the best solution within a more manageable solution space. It is the so-called three-phase cluster-based hybrid approach for the multi-depot vehicle routing problem with time windows (the m-VRPTW problem). By taking into account the customer time windows and the node distance matrix, a preprocessing stage is accomplished to first group the nodes into a few clusters so as to define an efficient, compact cluster-based VRPTW mathematical representation. The clusters are the new problem nodes and several of them can be visited by the same vehicle at the optimum. Each optimal cluster-based trip is subsequently disaggregated into the original requests through solving a small MILP model. In this way, detailed vehicle routes and schedules are found. This route construction method proved to be very robust to tackle not only a wide variety of Solomon's single-depot homogeneous fleet benchmark problems (Solomon, 1987) but also new multi-depot heterogeneous fleet VRPTW examples. Optimal or near-optimal solutions were obtained for a significant number of C-class problems with up to 100 clustered nodes, while the suboptimal gap for low-size RC and R-class problems with some node random distribution was larger but still acceptable. However, severe difficulties arise when the number of nodes is above 50 for RC-examples or higher than 25 for R-problems.

This paper presents a model-based large-scale neighborhood (LSN) search method that steadily improves an initial solution provided by the three-phase cluster-based hybrid approach of Dondo and Cerdá (2007). At each iteration, a sequence of two evolutionary steps is normally executed (the normal mode). First, a neighborhood around the starting solution is implicitly generated by developing a mixed-integer linear problem (MILP) formulation that allows multiple nodal exchanges between neighboring trips. By solving the MILP model, the best neighbor is found (the improvement step). Next, a new different neighborhood is defined by just allowing relocations of nodes on the same tour (the local route reoptimization step). A well-defined neighborhood structure at each step permits to identify not only the feasible moves (the problem variables) but also the solution space (the problem constraints) to be explored. In this way, manageable mathematical formulations for both subproblems can be developed and sequentially solved through an efficient branch-and-bound algorithm. At each step, the best neighbor minimizing the overall routing cost, including fixed and variable traveling expenses, is sought. Moreover, a mixed or perturbation mode is activated whenever no better neighbor is found through the normal procedure. The perturbation-mode explores a larger neighborhood that is generated by simultaneously making nodal exchanges among close trips and node reordering on every route. To further reduce the size of the MILP-formulations to solve under normal or perturbation modes, a spatial decomposition scheme has additionally been applied. In order to get a better evaluation of the algorithm performance, a sizable set of large-scale VRPTW benchmark problems with different nodal distributions has been tackled.

2. Model assumptions

1. Problem data are known with certainty and remain invariant with time; i.e. a deterministic, static VRPTW problem is studied.
2. Either pick-up or delivery services are provided to customers but not both.
3. Each pick-up or delivery node must be visited within the specified time window just once.
4. Though the problem can involve several depots, each route should start and end at the same depot.
5. The total load transported by a vehicle must never exceed its capacity.
6. Time-window and maximum trip duration constraints can be relaxed by including penalty cost terms in the objective function that linearly increases with the time-window violation or the trip overduration.
7. A feasible problem solution is available to start the improvement algorithm.
8. Assignment of depots to vehicles at the starting solution should remain unchanged during the whole improvement process, i.e. they are no longer problem variables.
9. The initial allocation of vehicles to trips constitutes a set of fixed decisions that remain valid throughout the entire process. However, the improved solution may include a number of tours lower than the starting value because some vehicles may no longer be required.
10. A local search approach that just analyzes the set of feasible points within the neighborhood of the current solution is applied.
11. The solution space (the neighborhood) to be explored is implicitly generated by two types of moves: (a) balanced/unbalanced exchanges of node strings among routes and (b) local repositioning of nodes on every route. However, just feasible moves satisfying capacity and time constraints are considered by the problem formulation.
12. The best feasible neighbor is chosen as the next base solution to restart the search, i.e. the best-improvement strategy has been adopted.

3. Major problem variables

Let the set I be the transportation requests to be serviced and the set V denote the available vehicle fleet. Let us identify each trip by the vehicle v assigned to it at the starting solution. Therefore, the subset I_v^o includes all nodes being serviced by vehicle v at the initial solution. At any iteration k of the route improvement process, two types of structural changes can be performed:

(1) transferring nodes $i \in I_v^{(k-1)}$ from trip v to a neighboring trip $v' \neq v$ and vice versa;
(2) changing the relative order of any pair of nodes (i,i') on every trip v so that vehicle v may service node i' before visiting node i instead of stopping earlier at node i.

Therefore, a mathematical representation of the VRPTW improvement problem should include two different types of 0–1 decision variables. Assignment variables $Y_{iv}^{(k)}$ that control the transfer of nodes i currently serviced by vehicle v' to a nearby trip v. If $Y_{iv}^{(k)} = 1$, then node i is transferred to trip v at iteration k. Otherwise, it remains on the same trip $v'(Y_{iv'}^{(k)} = 1)$ or moves to another neighboring route v''. Just one of such options will be permitted. In addition, a set of sequencing variables $X_{ii'}$ is also included in the problem formulation to reverse the order of nodes (i,i') located on the same trip v (i.e. $Y_{iv} + Y_{i'v} = 2$) if such a move produces traveling cost savings. Nodes (i,i') are usually located at non-adjacent positions and $X_{ii'} = 1$ just indicates that node i is visited before node i'. Only one variable $X_{ii'}$ controls the order of a pair of nodes (i,i') on the same trip. In this paper, the variable $X_{ii'}$ with $i < i'$ (i.e. node i appears before i' in the set I) is only defined.

Moreover, some important non-negative continuous variables are also defined to get a complete problem representation. They are: (a) the vehicle arrival time at node i (AT_i), $i \in I$; (b) the overall duration of trip $v(OT_v)$, $v \in V$; (c) the accumulated variable traveling cost from the starting depot to node $i \in I_v$ along the trip $v(C_i)$; (d) the total traveling cost of tour $v(OC_v)$, $v \in V$; (e) the size of time-window violations for early or late vehicle arrivals at node i (E_i or L_i). In turn, the major problem data are

- Asymmetrical traveling times (t_{ij}^v, t_{ji}^v) between a pair of nodes (i, j) or between a node i and the depot (t_{iv}, t_{vi}) assigned to vehicle trip v.
- Asymmetrical variable traveling costs ($c_{ij'}^v$, c_{ji}^v) between a pair of nodes (i,j) or between a node i visited by vehicle v and the designated depot (c_{iv}, c_{vi}).
- Duration of the pickup/delivery task at node i (st_i).
- The load to be delivered or picked-up at node i (w_i).
- The maximum capacity of vehicle v (q_v).
- The service time window within which the service must start at node i $[a_i,b_i]$.
- The maximum allowed service time for vehicle $v(tv_v^{\max})$.
- The unit-time penalty costs for early/tardy vehicle arrivals at node i (β_i,γ_i), and for vth-tour overduration (α_v).

4. The neighborhood structure

Problem nodes can be classified into two types: fixed and moving nodes. A fixed node $i \in I_v^F \subseteq I_v$ should stay on the current trip v because the chance of reducing the overall traveling costs by moving node i to a neighboring route is rather low. However, it can be repositioned within the same route. Some simple criterion to estimate the likelihood of getting savings in traveling costs from moving node i to another trip v' is then required. Obviously, I_v^F may change from one iteration k to the next $(k+1)$ since the set of nodes on the trip v generally varies. On the contrary, it is expected that the transfer of a moving node $i \in I_v^M = (I_v - I_v^F)$ to closer trips probably leads to a lower-cost solution. A route can have one or several neighboring trips but the set of candidate routes for a particular moving node i at iteration k, $V_i^{(k)}$, will comprise only some of them plus the one to which belongs at the start of iteration k. In short, the neighborhood structure will comprise vehicle routes generated by (a) transferring mobile nodes, for instance node $i \in I_v^M$, from the current route v to some arbitrary position on another trip $v' \in V_i (v' \neq v)$ and, (b) reordering fixed nodes $i \in I_v^F$ on the current trip v. Since the neighborhood also contains the current solution, it will be never be an empty set.

To classify the nodes on a trip into fixed or moving nodes, easy-to-compute criteria closely related to the widely known sweep heuristic have been developed (Gillet & Miller, 1974). Such a sweep heuristic groups together the nodes based on their angular coordinate with regards to some line radiating from the central depot. As the line moves clockwise or counterclockwise, trips are constructed by allocating nodes with similar angular coordinates to the same vehicle while its capacity is not exceeded. In this way, a set of routes with an optimal topology pattern (i.e., non-overlapping petal-shaped tours) usually observed on a wide range of VRP problems is generated. In a similar manner, this paper introduces the angular distances φ_0 and φ_1, with $\varphi_1 < \varphi_0$, to define a pair of characteristic cones surrounding each trip v. Let us assume that the whole vehicle fleet is housed in a central depot. Both characteristic cones for a trip v has its origin at the central depot and its axis goes from the origin through the centre of gravity of the trip. Besides, the two delimiting rays form alternatively an angle φ_0 or φ_1 with the trip axis (see Fig. 1). Such characteristic cones are defined to categorize the nodes on a trip v as fixed or mobile. Values for the model parameters (φ_0, φ_1) are set by the user following the guidelines given in Section 9.

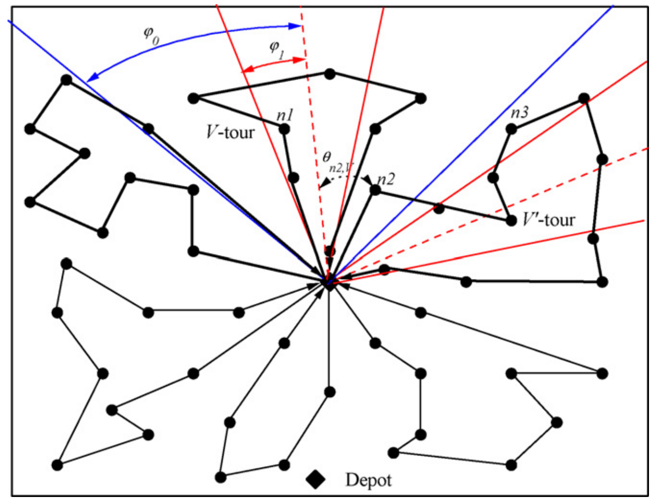

Fig. 1. Model parameters (φ_0, φ_1) defining fixed and mobile nodes.

The φ_1-slim cone defines a narrow zone around the trip axis containing the nodes that 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 $\theta_{iv} \le \varphi_1$, where θ_{iv} is measured from the vth-trip axis to the ray connecting node i to the central depot (see Fig. 1). Any node $j \in I_v$ outside the φ_1-cone ($\theta_{jv} > \varphi_1$) will be regarded as a potential mobile node on trip v that may be transferred to nearby trips. The angle θ_{jv} is determined from the polar coordinates of both the node j and the centre of gravity of route v to which currently belongs. Such polar coordinates are in turn computed from the Cartesian coordinates for the node j (x_j, y_j) and the depot ρ (x_p, y_p), respectively.

On the other hand, the φ_0-expanded cone defines a geographical area beyond the φ_1-cone that may contain nodes from neighboring routes $v' \neq v$ (see Fig. 1). In this way, the φ_0-cone around trip v permits to identify mobile nodes from neighboring routes that can be transferred to trip v. They are those lying inside the φ_0-cone around trip v. Therefore, a mobile node j from a neighboring trip v' featuring an angle $\theta_{jv} \le \varphi_0$ measured from the vth-trip axis to the ray connecting node j to the central depot has the route v as one of the candidate trips to which it can move, i.e. $v \in V_j^{(k)}$. In other words, the φ_0-cone aims to define the candidate routes for mobile nodes. As a result, the set of nodes that can move to trip v on the next iteration $k(I_v^k)$ becomes available. In addition to such mobile nodes from nearby tours, the set I_v^k will also include the fixed and moving nodes on the route v at the start of iteration k, i.e. $I_v^k \supseteq Iv^{(k-1)}$.

Fixed nodes in $Iv^{(k-1)}$ will surely stay in route v at the end of iteration $Iv^{(k-1)}$. Besides, the trip v for some mobile nodes in $Iv^{(k-1)}$ may still be the best choice. The angle φ_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_{jv} is just defined for a mobile node j only if $j \in I_v^k$. From Fig. 1, it can be observed that node $n1$ is a fixed node on tour v while $n2$ is a mobile node from route v' that can relocated to trip v. However, the mobile node $n3$ on route v cannot move to any other route.

Finally, it is defined a third model parameter d^{max} representing the maximum allowed Euclidean distance between a trip v and a mobile node $I \in I_v^k$ of a neighboring route. Despite $\theta_{iv} \leq \varphi_0$ and, therefore, the node i lies inside the φ_0-cone, every node i farther than d^{max} from the center of gravity of tour v cannot be transferred to v. If so, Y_{iv} can be deleted from the problem formulation. In all examples solved in the paper, the value of d^{max} was large enough to never exclude a potential nodal exchange between neighboring trips.

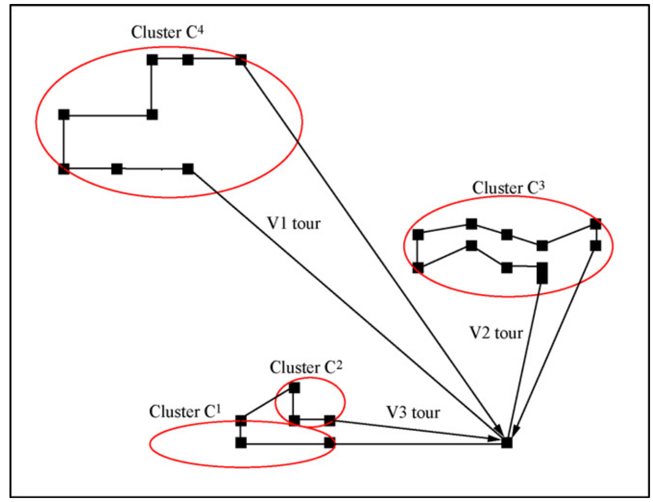

Fig. 2. Best solution for example C-101.25 through the initialization procedure.

5. The initialization procedure

To get a good starting solution, the three-phase cluster-based construction method for m-VRPTW problems proposed by Dondo and Cerdá (2007) has been applied. Phase I is intended to massively reduce the computational burden of the subsequent phases by cleverly grouping all customer locations into a rather small set of feasible, cost-effective clusters or "hyper-nodes". A "feasible" cluster means that (a) the cluster cargo can be assigned to a single vehicle and, in addition, (b) there exists at least a route connecting the nodes on the cluster that satisfies all the related time window constraints. By developing the mathematical model in terms of few clusters rather than a huge number of customers, the VRPTW problem size can be sharply decreased. The aim of phase II is to simultaneously assigning clusters to vehicles and sequencing clusters visited by the same vehicle through a compact MILP model. Ordering nodes within clusters and scheduling customer services on every tour are the goals of phase III. To reach those objectives, a low-size MILP mathematical formulation must be solved as many times as the number of tours found in phase II. Accounting for the relative ordering of clusters on the same tour found in phase II, the number of sequencing variables S_{ij} to be considered in phase III can be sharply diminished. This three-phase construction method can efficienty find near-optimal solutions for 100-node VRPTW examples with clustered nodal distributions. Though a much better performance for problems with clustered (C-class) and randomized clustered (RC-class) nodal distributions is expected, the initialization approach has also been used to tackle rather small R-class VRPTW examples. In addition, the 3-phase construction algorithm is able to optimize vehicle-depot assignment decisions for multi-depot VRPTW problems.

To illustrate the initialization procedure, a small example involving 25 customer locations has been tackled. It was derived from the original benchmark problem C-101 proposed by Solomon (1987) that originally involves 100 nodes by just considering the first 25 customers. The example assumes a 3-vehicle homogeneous fleet and a single depot (see Fig. 2). Through phase I, the 25 original nodes have been merged into four customer clusters C^1–C^4. In phase II, the three available vehicles are allocated to such four clusters. Two of them visit just a single cluster while the remaining one visits the sequence of clusters C^1–C^2. In phase III, clusters are disaggregated into the original nodes to find the three tour schedules. The solution depicted in Fig. 2 is the truly problem optimum (Dondo & Cerdá, 2007).

6. The problem solution strategy

To improve the starting set of routes, a model-based iterative approach has been adopted. At each iteration k, the sets of fixed $(I_v^F)^k$ and mobile nodes $(I_v^M)^k$ on every route $v \in V$ as well as the candidate neighboring routes for any mobile node $i(V_i^k)$ are all updated. In this way, the neighborhood structure around the new current solution is found. To get a better set of feasible routes from the one available at the start of iteration k, a MILP mathematical formulation accounting for nodal exchanges between neighboring trips and nodal repositioning on individual routes is to be solved. Since the solution space just accounts for alternative routes within the proposed neighboring structure, a local optimum is just found. By repeating this procedure at every iteration k, the proposed approach generates a sequence of solutions featuring an overall routing cost z^k that steadily diminishes. Hopefully, it will converge to the problem optimum. The three-step cluster-based VRPTW algorithm introduced by Dondo and Cerdá (2007) was used to provide a good starting point.

Though the number of node string exchanges among trips can be limited by a proper choice of the model parameters (φ_0, φ_1), the simultaneous handling of both types of improving actions (node exchange, local node repositioning) leads to a large, somewhat intractable problem formulation. To overcome such a difficulty, a mathematical decomposition strategy based on gathering the improvement moves into two sets has been adopted. Each set just comprises a single type of improvement moves, either node exchange between close tours or node reordering on the same tour. At each iteration k, a pair of subproblems rather than a single one is to be sequentially solved with each one allowing just moves of a single type. Subproblem I hopefully producing traveling cost savings by exchanging customers among neighboring trips is first tackled. Mobile nodes $i \in I^M$ can be relocated to another candidate route $v \in V_i$ by setting $Y_{iv} = 1$. Next, the new set of routes provided by subproblem I is further improved by optimally reordering all nodes on every individual trip through subproblem II. The optimal values for the sequencing variables X_{ij} controlling the relative locations of nodes (i,j) on the same tour are to be encountered. Therefore, there will be as many instances of subproblem II as the number of routes at the optimum of subproblem I. The iterative procedure is stopped when the decrease in overall traveling expenses, including penalty costs, from iteration $(k-1)$ to k given by $\Delta z = (z^{k-1} - z^k)$ is lower than some prespecified small number

$\varepsilon > 0$. In short, the neighborhood structure has been divided into two parts, one involving neighboring trips generated by exchange of mobile nodes exclusively while the other includes routes produced by the complete repositioning of nodes on individual routes. In each case, the neighborhood is fully explored and the best option becomes the new incumbent solution. After updating the neighborhood around the new current solution provided by subproblem I, subproblem II is solved again or vice versa. The procedure is repeated until no further improvement on the objective function is achieved.

Since this iterative, sequential improvement procedure can be trapped into a local optimum, a mixed-mode phase must be activated whenever the routing cost reduction per iteration becomes lower than a sufficiently small value $\varepsilon > 0$. The perturbation mode is performed by solving subproblem III that allows not only the reordering of nodes on every individual route but also the transfer of mobile nodes to candidate neighboring routes, i.e. both types of improvement moves. To this end, φ_1 is driving to zero and every customer on any route υ is a mobile node. In particular, a node j is permitted to move from route υ to υ' only if it features an angle $\theta_{j\upsilon'} \le \varphi_0$ from the υ'-trip axis to the ray connecting node j to the depot. The model parameter φ_0 should be carefully adjusted to define neighborhoods of proper size that lead to a solvable, computationally efficient mathematical formulation for subproblem III. Therefore, the value of φ_0 defining the set of candidate tours for any node i at subproblem III should be lower than the one chosen for subproblem I. Usually, two-third of the φ_0-value for subproblem I. Since subproblem III is more costly and aims to just moving away from a local optimum, it is applied only once every time the normal mode gets stuck in a local optimum. In some large VRPTW problem instances, however, almost 50% of the trip improvements come from subproblem III. When the routing cost reduction provided by solving subproblem III is less than a small parameter $\varepsilon > 0$, the improvement procedure should be stopped and the current solution is the best one found for the problem.

7. Mathematical formulations for subproblems I–III

In this section, mixed-integer linear problem formulations for subproblems I and II are presented. For sake of simplicity, it has been omitted the supraindex k indicating the iteration to which the sets I^F, I^M and V_i correspond. The mathematical formulation for subproblem III can be derived from subproblem I by simply making: $I^F = \emptyset$ and, consequently, $I = I^M$. Therefore, every problem constraint on fixed nodes must be deleted. The fact that each node i is a mobile one at subproblem III does not mean that it could be transferred to any neighboring trip. Often, V_i denoting the set of candidate tours for node i may only include its current route.

7.1. Subproblem I: Exchanging customers among neighboring routes

7.1.1. Problem constraints

Re-assignment of vehicles to customers:

$$\sum_{\upsilon \in V_i} Y_{i\upsilon} = 1 \quad \forall i \in I^M \tag{1}$$

where I^M stands for the mobile node set and V_i represents the subset of tours to which mobile node i can move, including the one where it is currently located. Constraint (1) states that every customer location must be serviced by a single vehicle. Therefore, the load to be picked up at any node i cannot be allocated to multiple vehicles.

Vehicle capacity constraints:

$$\sum_{i \in I_\upsilon^F} l_i + \sum_{i \in I_\upsilon^M} l_i Y_{i\upsilon} \le q_\upsilon \quad \forall \upsilon \in V \tag{2}$$

where l_i is the amount of load to be pick up at node i, I_υ^F is the set of fixed nodes currently visited by vehicle υ, I_υ^M is the set of mobile nodes that can be allocated to vehicle υ and q_υ is the υth-maximum capacity.

Relationships between arrival times/traveling costs at nodes (i, j) on the same tour:

- For a pair of mobile nodes ($i, j \in I^M$):

$$\left\{\begin{array}{l} C_j \ge C_i + c_{ij}^\upsilon - M_C(1 - X_{ij}) - M_C(2 - Y_{i\upsilon} - Y_{j\upsilon}) \\ AT_j \ge AT_i + st_i + t_{ij}^\upsilon - M_T(1 - X_{ij}) - M_T(2 - Y_{i\upsilon} - Y_{j\upsilon}) \\ C_i \ge C_j + c_{ji}^\upsilon - M_C X_{ij} - M_C(2 - Y_{i\upsilon} - Y_{j\upsilon}) \\ AT_i \ge AT_j + st_j + t_{ji}^\upsilon - M_T X_{ij} - M_T(2 - Y_{i\upsilon} - Y_{j\upsilon}) \end{array}\right\}$$

$$\forall i, j \in I^M (i < j), \quad \upsilon \in V_i \cap V_j \tag{3}$$

where the adopted values for the upper bounds M_C and M_T are given by

$$M_C = max(c_{ij}/t_{ij}) * t\upsilon_\upsilon^{max} \quad \text{and} \quad M_T = t\upsilon_\upsilon^{max}$$

- For a pair of fixed nodes ($i, j \in I^F$):

Let us assume that fixed nodes (i, j) are both currently allocated to vehicle υ and node i is earlier visited. Then, node i is either a direct or a non-direct predecessor of node j on the υth tour.

$$\left\{\begin{array}{l} C_j \ge C_i + c_{ij}^\upsilon \\ AT_j \ge AT_i + st_i + t_{ij}^\upsilon \end{array}\right\} \quad \forall i, j \in I_\upsilon^F, \quad i \in PR_j \tag{4}$$

- For a pair of mixed nodes ($i \in I^F, j \in I^M$):

Let us suppose that node i is fixed and currently assigned to vehicle υ while j is a mobile node that can be allocated to vehicle $\upsilon (j \in I_\upsilon^M)$. Assuming that $i < j$, then:

$$\left\{\begin{array}{l} C_j \ge C_i + c_{ij}^\upsilon - M_C(1 - X_{ij}) - M_C(1 - Y_{j\upsilon}) \\ AT_j \ge AT_i + st_i + t_{ij}^\upsilon - M_T(1 - X_{ij}) - M(1 - Y_{j\upsilon}) \\ C_i \ge C_j + c_{ji}^\upsilon - M_C X_{ij} - M_C(1 - Y_{j\upsilon}) \\ AT_i \ge AT_j + st_j + t_{ji}^\upsilon - M_T X_{ij} - M_T(1 - Y_{j\upsilon}) \end{array}\right\}$$

$$\forall i \in I^F, \quad j \in I^M : i < j, \quad \upsilon \in V_j \tag{5}$$

In case that $j < i$, X_{ij} should be replaced by $(1 - X_{ij})$ in the above inequalities.

Arrival time/traveling cost at the node first visited: If node i is first visited, then constraints (3)–(5) become redundant and additional constraints should be included.

- For a mobile node ($i \in I^M$):

$$C_i \ge \sum_{\upsilon \in V_i} c_{\upsilon i} Y_{i\upsilon} \quad \forall i \in I^M \tag{6}$$

$$AT_i \ge \sum_{\upsilon \in V_i} t_{\upsilon i} Y_{i\upsilon} \quad \forall i \in I^M \tag{7}$$

where t_{vi} is the travel time from the vth-depot to mobile node i, and c_{vi} is the travel cost from that depot to node i.

- For a fixed node ($i \in I^F$):

$$C_i \geq c_{vi} \quad \forall i \in I^F \tag{8}$$

$$AT_i \geq t_{vi} \quad \forall i \in I^F \tag{9}$$

Overall traveling time/cost for a tour v: Let us define the overall variable traveling cost OC_v and the total duration OT_v for the trip v. In addition to the OC_v-term, the objective function will include vehicle fixed costs and penalty costs for early and/or late vehicle arrivals at customer locations and tour overdurations.

- For a mobile node i that can be allocated to trip v:

$$OC_v \geq C_i + c_{iv} - M_C(1 - Y_{iv}) \tag{10}$$

$$OT_v \geq T_i + st_i + t_{iv} - M_C(1 - Y_{iv}) \quad \forall i \in I_v^M, \quad v \in V_i \tag{11}$$

- For a fixed node currently visited by vehicle v:

$$OC_v \geq C_i + c_{iv} \tag{12}$$

$$OT_v \geq T_i + st_i + t_{iv} \quad \forall i \in I_v^F, \quad v \in V \tag{13}$$

where c_{iv} and t_{iv} stand for the travel cost/time along the shortest route from node i to the depot.

Earliness or tardiness on the vehicle arrival at node i:

$$\left\{ \begin{array}{l} E_i \geq a_i - T_i \\ L_i \geq T_i \ - b_i \end{array} \right\} \quad \forall i \in I \tag{14}$$

Overduration of trip v:

$$OD_v \geq OT_v - tv_v^{max} \quad \forall v \in V \tag{15}$$

tv_v^{max} is the maximum allowed duration of tour v and OD_v is the overduration of tour v.

7.1.2. Objective function

The problem goal is to minimize the total service cost, including the penalties for early/late arrivals and tour overdurations.

$$\text{Min} \sum_{v \in V} (OC_v + \alpha_v OD_v) + \sum_{i \in I} (\beta E_i + \gamma L_i) \tag{16}$$

7.2. Subproblem II: repositioning nodes on every individual tour v

In subproblem II, the exchange of nodes among tours is prohibited. To reduce service costs, only full reordering of nodes on every trip can be made, i.e. a traveling-salesman problem (TSP). Therefore, assignment variables Y_{iv} are no longer needed and sequencing variables X_{ij} are to be defined for every pair of nodes on the same tour. As a result, constraints (1)–(2) and (10)–(11) are deleted from the problem formulation and, at the same time, constraints (3)–(9) become much simpler because variables Y_{iv} must be omitted. The arrival time at node i can be changed by repositioning node i on the current tour. Effective heuristics for TSP can alternatively be used to solve subproblem II.

7.2.1. Problem constraints

Relationships between vehicle arrival times/traveling costs at nodes $(i, j) \in I_v$:

$$\left\{ \begin{array}{l} C_j \geq C_i + c_{ij}^v \ - M_C\,(1 - X_{ij}) \\ AT_j \geq T_i + st_i + t_{ij}^v \ - M_T\,(1 \ - X_{ij}) \\ C_i \geq C_j + c_{ji}^v \ - M_C\,X_{ij} \\ AT_i \geq T_j + st_j + t_{ji}^v \ - M_T\,X_{ij} \end{array} \right\} \quad \forall i, j \in I_V (i < j) \tag{17}$$

Arrival time/traveling cost at the node $i \in I_v$ first visited:

$$\begin{array}{l} C_i \geq c_{vi} \\ AT_i \geq t_{vi} \end{array} \quad \forall i \in I_V \tag{18}$$

Overall traveling time/cost for tour v:

$$\begin{array}{r} OC_v \geq C_i + c_{iv} \\ OT_v \geq T_i + st_i + t_{iv} \end{array} \quad \forall i \in I_V \tag{19}$$

Earliness and tardiness in starting the service at node i:

$$\left\{ \begin{array}{l} E_i \geq a_i - T_i \\ L_i \geq T_i \ - b_i \end{array} \right\} \quad \forall i \in I \tag{20}$$

Overduration of trip v:

$$OD_v \geq OT_v - tv_v^{\max} \quad \forall v \in V \tag{21}$$

7.2.2. Objective function

Similarly to subproblem I, the goal is to minimize the total service expenses. Therefore, the objective function is given by (16).

7.3. Subproblem III: simultaneous reordering and exchange of nodes

For subproblem III, the constraints (4)–(5), (8)–(9) and (12)–(13) written for subproblem I no longer arise since $I^F = \emptyset$ and, therefore, every constraint on fixed nodes or any pair of mixed nodes must be deleted. Moreover, a candidate set of tours V_i should be defined for every node $i \in I$ though it may occur that V_i for some nodes just include the current choice. Therefore, the mathematical formulation for subproblem III is given by Eqs. (1)–(3), (6)–(7), (10)–(11) and (14)–(16).The problem size grows because the number of assignment and sequencing variables both increase. A similar impact on the number of assignment and sequencing constraints is observed.

7.4. Exact elimination rules

In order to reduce the number of assignment and sequencing variables, especially for subproblem III, the information on the customer time windows can help develop some exact elimination rules (Dondo & Cerdá, 2007). Such rules that regard the time windows are hard constraints can be stated as follows:

- (Subproblems I and III) If none of the used vehicles $v \in V$ can service a pair of mobile nodes $(i,j) \in I$ without violating the relating time window constraints, then two different vehicles must be used. Then:

  $$Y_{iv} + Y_{jv} \leq 1, \qquad \text{for any} \quad v \in V (= V_i \cap V_j)$$

  and the variable X_{ij} plus the related sequencing constraints for the pair of mobile nodes (i,j) can be deleted from the problem formulation.
- (Subproblems II and III) Let suppose that nodes i and j are visited by the same vehicle v. If $(a_i + st_{iv} + t_{ij}) > b_j$, then vehicle v cannot stop first at node i and, consequently, the sequencing variable X_{ij} must be fixed to zero ($X_{ij} = 0$). Moreover, the expression of any problem constraint involving X_{ij} becomes simpler. By fixing $X_{ij} = 0$, it follows that node j will be serviced before node i only if the same vehicle v visits both nodes ($Y_{iv} + Y_{jv} = 2$). Similarly, if $(a_j + st_{jv} + t_{ji}) > b_i$ and both nodes (i,j) are on the same tour, then node i must be first visited and $X_{ij} = 1$.

8. Spatial decomposition of large VRPTW problems

As the neighborhood structure accounts for solutions generated from the best available set of routes by reordering nodes on each individual tour or relocating customers to neighboring trips, there is no sense in tackling the whole VRPTW problem at once. In a local search environment, each tour just exchanges nodes with a few other 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 (CD) and delimiting rays emanating from the CD with angular coordinates Ω_1 and Ω_2, respectively (see Fig. 3). The angle Ω (=$\Omega_2 - \Omega_1$) between the extreme rays remains fixed as the RAS rotates. In order to sweep the whole service region, the RAS will turn around the CD by equally increasing the extreme ray angular coordinates Ω_1 and Ω_2 by a fixed quantity Ω. In this way, the RAS will take a series of angular positions $\{\Omega^{(1)}, \Omega^{(2)}, \ldots\}$ before rotating 2π and start a new turn. If some but not all nodes of a trip v are inside the RAS$^{(m)}$ 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$^{(m)}$. In other words, a node i will pertain to RAS$^{(m)}$ if either (a) its angular coordinate Θ_i is between ${\Omega_1}^{(m)}$ and

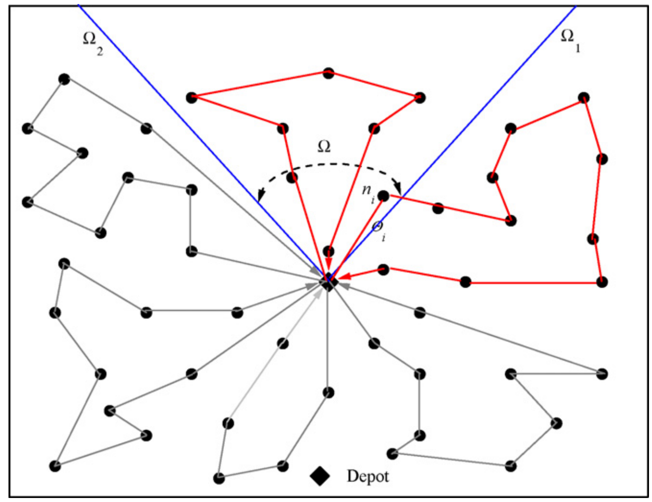

Fig. 3. The rotating angular sector (RAS) decomposing the service region into smaller zones.

Let $\Delta z^{(m)}$ be the transportation cost savings achieved on the last iteration for zone m while ΔZ denotes the decrease on the objective function over the whole service region during the last turn of the rotating angular sector (RAS).

- **Initialization Phase**
  - (i) Find an initial solution to start the improvement process (Dondo and Cerdá, 2007).
  - (ii) Choose a pair of values for each procedure parameter (φ_o, φ_l, Ω) to be applied under the normal mode and the perturbation mode, respectively. Compute: $N = 2\pi / \Omega$.
  - (iii) Set the values for the parameters T, d_{mx} and ε.

- **Improvement Phase**

  Set: $m = 1$, MODE = *normal*

  For the angular sector or service zone m:
  - a) Find the vehicle/node sets defining the neighborhood to be explored
    - Identify the tours lying on the angular sector m (V^m)
    - Identify the fixed and mobile nodes on each tour (I_v^F, I_v^M for $v \in V^m$)
    - Identify the candidate tours for every mobile node ($V_i^{(m)}$ for $i \in I_v^M$)
  - b) Solve the proper VRPTW improvement formulation

    MODE = *normal*

    Solve Subproblem I

    Eliminate any tour v featuring $I_v = \varnothing$

    Update the sets I_v^F, I_v^M and $V_i^{(m)}$ (for $i \in I_v^M$)

    Solve Subproblem II

    Go to Step (c)

    MODE = *perturbation*

    Solve Subproblem III

    Eliminate any tour v featuring $I_v = \varnothing$

    Update the sets I_v^F, I_v^M and V_i (for $i \in I_v^M$)
  - c) Decide what to do next

    Compute the cost savings ($\Delta z^{(m)}$) achieved on Step (b)

    If ($\Delta z^{(m)} > \varepsilon$) then Repeat Step (b)

    If ($\Delta z^{(m)} \leq \varepsilon$ and $m < N$) then $m = m+1$ and Go to Step (b)

- **Termination Phase**

  Compute the overall decrease on the total transportation cost (ΔZ) during the last turn of the rotating angular sector.

  If ($\Delta Z > \varepsilon$) then Restart the Improvement Phase

  If ($\Delta Z \leq \varepsilon$ and MODE = normal) then Set MODE= *perturbation* and restart the Improvement Phase

  Otherwise, stop the procedure. The algorithm has converged to a stationary point.

Fig. 4. The VRPTW neighborhood improvement algorithm.

$\Omega_2{}^{(m)}$ ($\Omega_1{}^{(m)} \leq \Theta_i \leq \Omega_2{}^{(m)}$) 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$^{(m)}$ will contain a limited number of complete tours. Moreover, a trip v may belong to two consecutive locations $\Omega^{(m)}$ and $\Omega^{(m+1)}$ of the rotating sector. 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 subproblems I and II for a particular location $\Omega^{(m)}$ are solved, only routes inside RAS$^{(m)}$ will be considered. Nodes and trips beyond RAS$^{(m)}$ are ignored. Mathematical formulations for subproblems 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, subproblems 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 node after sweeping the N locations, i.e. the whole service area. In order to avoid getting stuck on a local optimum, subproblem 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. 4 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 for the vehicles servicing the geograhical area covered by RAS$^{(m)}$. 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 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 of nodes I_v and the ordering of them 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 normal and perturbation modes both fail to provide a better solution or the improvement on the objective function is less than a small positive scalar ε, 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.

Since the construction algorithm of Dondo and Cerdá (2007) also optimizes the vehicle-depot assignment decisions, the proposed improvement algorithm can also be extended to multi-depot VRPTW problems by just freezing the initial vehicle-depot allocations. To do that, a higher-level outer loop should be included in order to sweep the whole service area as many times as the number of depots involved in the problem. At the major iteration for depot d, the RAS element turns around depot d and the improvement process will just consider the tours starting and ending at that depot.

9. Numerical results and discussion

The proposed VRPTW improvement algorithm has been applied to a significant number of Solomon's benchmark problems (Solomon, 1987). The classical collection of 56 Solomon's problems has been grouped into three different categories: C, R and RC. Problems of class C feature clustered customers whose time windows have been generated based on known solutions. Customer locations in R-class problems were randomly generated over a square while RC-class problems comprise a combination of clustered and randomly generated customers. The data set for every category comprises from 8 to 12 examples all comprising 100 nodes with the same type of nodal distribution, a central depot, similar vehicle capacities but different time-window distributions. Problem data also include the number of available vehicles, Euclidean distances among customers and normalized vehicle speeds making traveling times and Euclidean distances numerically identical. Furthermore, time windows are regarded as hard constraints, service times are independent of customer requirements and the tour duration cannot exceed a maximum value tv_v^{max}. The selected objective is the minimization of the total distance cost. Benchmark 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 small vehicle capacities, while type-2 problems feature wider time windows and larger vehicle capacities. Solutions to type-2 problems include fewer tours and longer scheduling horizons because vehicles have higher capacities. In addition, larger time windows make VRPTW problems very difficult to solve. This is so because the elimination rules based on customer time-windows can delete a fewer number of arcs and have a much less impact on the problem size. By using the cluster-based construction algorithm of Dondo and Cerdá (2007), the best solutions reported in the literature for a set of nine 100-node Solomon's benchmark problems with clustered distributions were found. Such a problem set includes the examples: C101-C102, C105-C109, C201 and C205. Solution times for C1-problems ranges from 70 s to 120 s, while problems C201 and C205 were solved in 23.2 s and 1960s, respectively (Dondo & Cerdá, 2007). Because of the good performance of the initialization procedure for class-C problems, most of the examples solved in the paper presents random and RC-nodal geographical distributions. Recommended values for the parameter φ_0 are: 0.9–1.2 for problems (C1, RC1), 0.5–0.6 for problems (R1, C2, RC2) and 0.3 for problems R2. In turn, the ratio (φ_0/φ_1) has been set equal to 3.

9.1. Illustrating the proposed VRPTW improvement method

To start with a simple example, a small version of Solomon's problem R-112 (Solomon, 1987), called R-112.25, that just accounts for the first 25 nodes was initially solved. The search begins from a feasible solution provided by the initialization procedure of Dondo and Cerdá (2007). This procedure generates near-optimal solutions for C-class problems at low CPU time but both solution quality and computational efficieny badly deteriorate when applied to R-class problems. The starting solution for problem R-112, shown in Fig. 5a, involves four petal-shaped tours and features a total cost of 428.9 units, i.e. 8.86% above the best reported value (Kallehauge et al., 2001). In order to get a better solution through the proposed VRPTW improvement methodology, the geographical area to be serviced was partitioned into five angular sectors of equal-size. In other words, it was swept by a rotating angular sector (RAS) with a fixed angle $\Omega = \pi/2.5$ between its delimiting rays, and five different positions $N = 5$ per turn. For the perturbation mode, N was increased to 10 by choosing $\Omega = \pi/5$ to decrease the size of the MILP model. Moreover, the following values for the model parameters: $\varphi_1 = 0.30$, $\varphi_0 = 0.90$ (normal mode) and $\varphi_0 = 0.60$ (perturbation mode) were adopted. Such model parameters were defined to categorize the customers at each angular sector into fixed and mobile nodes and, in addition, identify the set of candidate tours for each mobile node. At every $\Omega^{(m)}$, the pair of subproblems I–II was iteratively solved by using ILOG OPL Studio 3.7 on a 1 Gb RAM 2 GHz Pentium IV PC. Since the lowest-cost neighbor is sought, then the best improvement (BI)

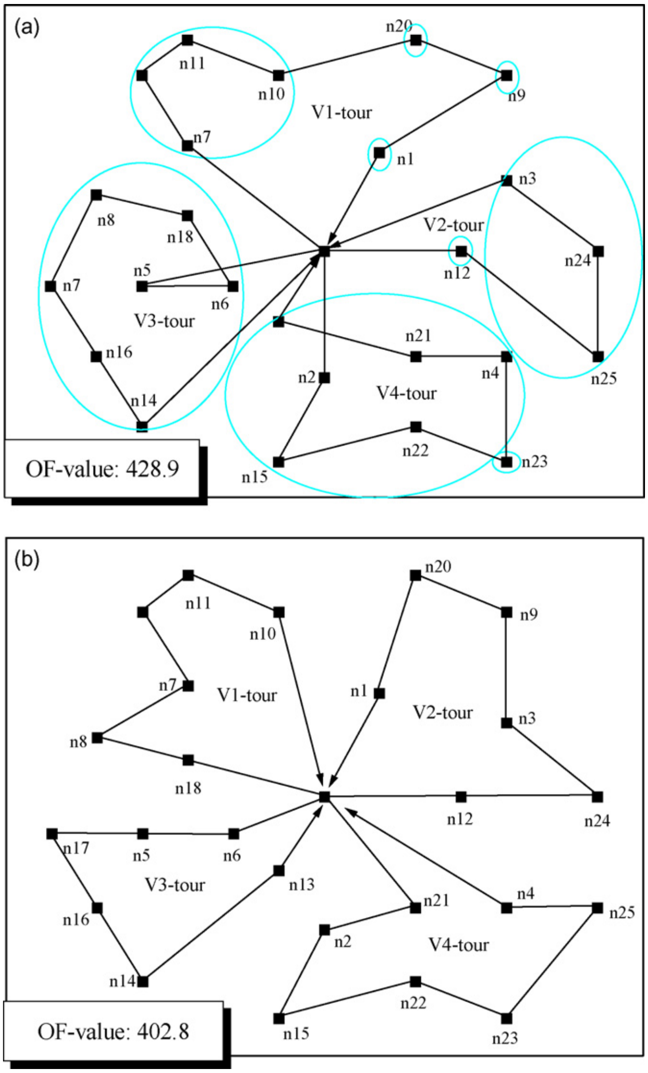

Fig. 5. (a) Initial solution for example R-112.25 and (b) final solution for example R-112.25.

strategy has been adopted. Moreover, any kind of improvement move (string crossing, string exchange or string relocation) is considered. After the RAS completes the first whole rotation, the total distance cost decreases from 428.9 to 422.0 units in 0.7 s of CPU time by just using the normal mode. Fig. 6a shows the variation of the objective function and the computer time as the improvement algorithm proceeds. During the second RAS-revolution, no improvement at all was achieved and, consequently, the perturbation mode (subproblem III) was activated. In this way, a better solution was found on the next turn and the total cost diminishes to 413.9 units in 7.9 s. Therefore, the normal mode was re-started again to further decrease the objective function up to 402.8. No improvement at all was obtained on the next two turns even if the perturbation mode is applied and, consequently, the procedure was stopped (see Fig. 6a). The new set of tours provided by the proposed VRPTW improvement algorithm is depicted in Fig. 5b. Its total transportation cost is 2.2% above the best-reported value, i.e. around one-fourth of the initial suboptimal gap. Moreover, it was found in only 13.3 s after the RAS swept the service region six times. Fig. 5b confirms that good solutions tend to comprise a set of non-overlapped petal-shaped routes though stringent time-window constraints may slightly distort some of them. The same final result has been found by tackling the whole problem R-112.25 at once, i.e. by choosing $\Omega = 2\pi$.

By making a comparison between the initial and the improved solutions for problem R-112.25, it follows that a total of seven string exchanges among tours and three local node repositioning on individual tours have been accomplished. Thus, the node string (n20, n9, n1) has been removed from V1-tour and inserted at the end of V2-tour, and simultaneously nodes (n20, n9) swaps positions to yield the subsequence (n9, n20, n1). In addition, the string (n18, n8) was relocated from V3-tour to the start of the neighboring V1-tour. Therefore, a non-balanced string exchange between neighboring trips V1 and V3 has been executed. Moreover, the node n13 initially assigned to vehicle V4 is the last visited by V3 on the final solution, while node n25 serviced by vehicle V2 at the initial solution was relocated on the V4-route. Both moves cause the relocation of nodes to other tours where they were optimally inserted. Finally, nodes (n5, n6) swap positions on the V3-tour, and node n21 moves forward along the V4-trip to be the first customer being visited. Such reordering of nodes can be regarded as local route improvements In short, none of the initial tours remains unchanged. Another interesting observation is the fact that four of the seven string exchanges/relocations and one of local node reordering moves all occur during the first rotation of the RAS by just applying the normal mode.

9.2. A variant of Solomon's benchmark problem R-104 involving the first 50 nodes

On the other hand, the starting solution for a variant of problem R-104 comprising the first 50 nodes is shown in Fig. 7a. It is a non-optimal solution provided by the initialization algorithm that consists of six tours and features a total distance cost amounting to 771.5 units, i.e. a suboptimal gap of 23%. In order to improve this solution, it was chosen as the initial point for the proposed VRPTW improvement algorithm. The procedure converges in 75.9 s after sweeping the whole service region six times. Values for the parameters φ_1 and φ_0 are similar to the ones adopted at problem R-112.25. On the first pair of rotations, the normal mode successively reduces the objective function value first from 771.5 to 684.1 units and then to 647.0 units in 44.6 s (see Fig. 6b). Since no improvement at all has been achieved on the next turn of the RAS, the perturbation mode was successfully activated on the fourth rotation to get a better solution with a total distance cost of 635.0 units, i.e. only 1.5% above the best reported value (Kallehauge et al., 2001). No further decrease of the objective function has been obtained on the next two rotations and, consequently, the procedure was stopped.

The best solution found for problem R-104.50 is shown in Fig. 7b. A total of six node strings has been relocated in neighboring tours and a significant number of node reorderings on each individual tour was also executed to yield the final solution. From Figs. 7a and b, it follows that the string (n16, n5, n17, n45) initially visited by V2 was inserted on the V1-route and simultaneouly reordered to yield the subsequence (n16, n17, n45, n5). Other modifications on the starting V1-trip includes (a) the insertion of node n14 previously serviced by V2, (b) the shifting of node n6 along the route V2 from the initial to the last stop and (c) the removal of node n43 to be relocated on the V2-trip. In addition to the insertion of n43, the route V2 presents another modification consisting in relocating node n18 initially serviced by V2 as the first stop on route V3. In contrast to what happens with vehicles V1 and V2, the groups of customers serviced by V3, V5 and V6 remain unchanged or exhibit just a single modification (insertion of n18 in route V3), but the nodes on each route have been significantly reordered. Moreover, the node string on route V3 initially consisting of (n7,

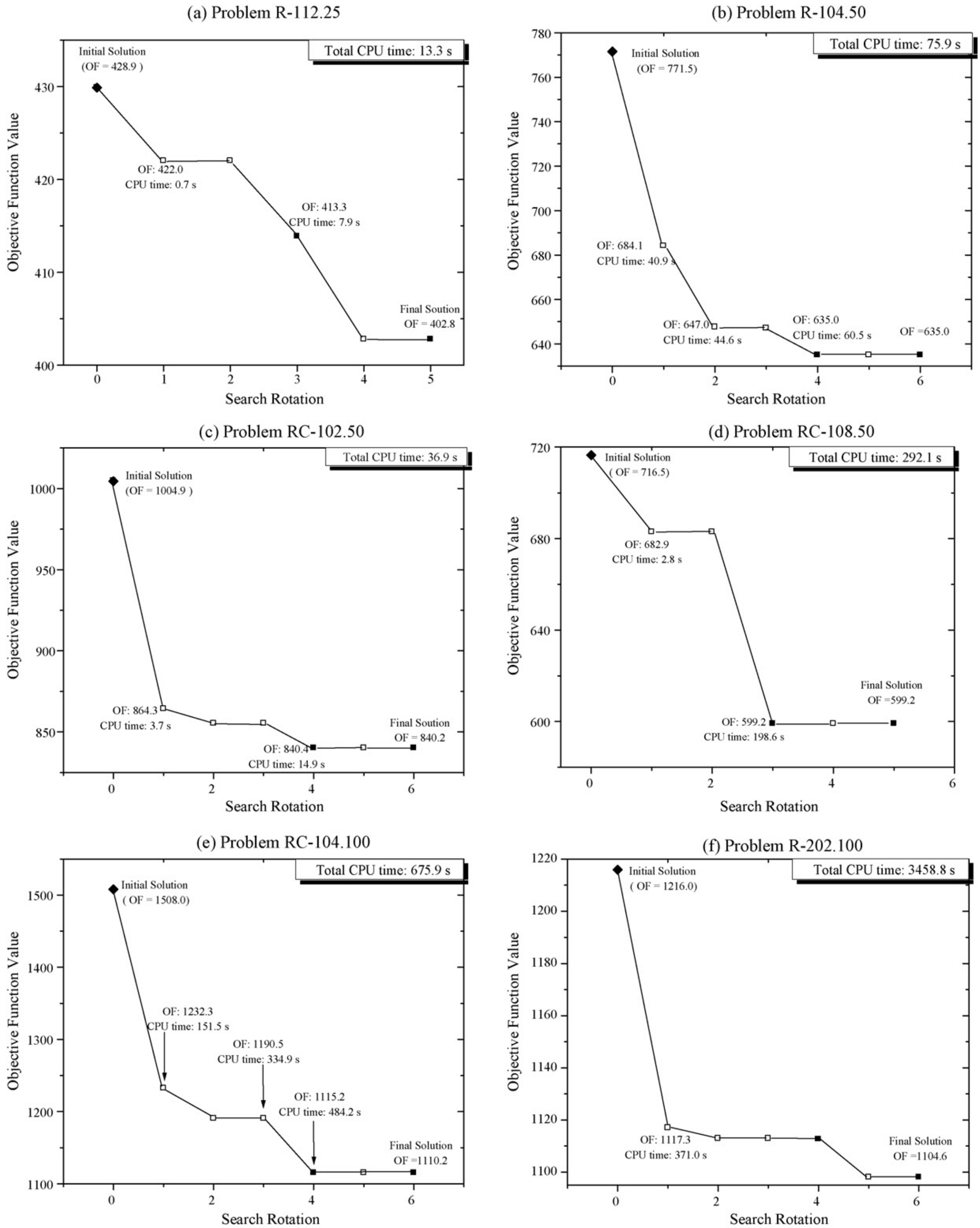

Fig. 6. Variation of the objective function with the number of RAS-rotations for some Solomon's benchmark problems.

$n19$, $n36$, $n8$, $n48$, $n47$, $n49$, $n46$) was totally rearranged to turn into ($n18$, $n48$, $n46$, $n8$, $n47$, $n36$, $n49$, $n19$, $n7$). Similarly, the chain of nodes on V6-tour initially given by ($n10$, $n30$, $n11$, $n31$, $n32$, $n20$, $n1$, $n27$) turns into the sequence ($n20$, $n31$, $n10$, $n11$, $n32$, $n20$, $n30$, $n1$). The same situation arises when the initial and the final node strings on the tour V5 are compared. In contrast, the initial sequence of nodes in V4 still appears at the final solution (see Fig. 7b).

9.3. *Improving the starting solution for low/medium size benchmark problems*

To show the enhancement of the initial solution found through the construction algorithm of Dondo and Cerdá (2007), the proposed VRPTW improvement methodology has been applied to nine R-class and RC-class benchmark problems involving from 25 to 50 nodes. The non-optimal value of the objective function at the

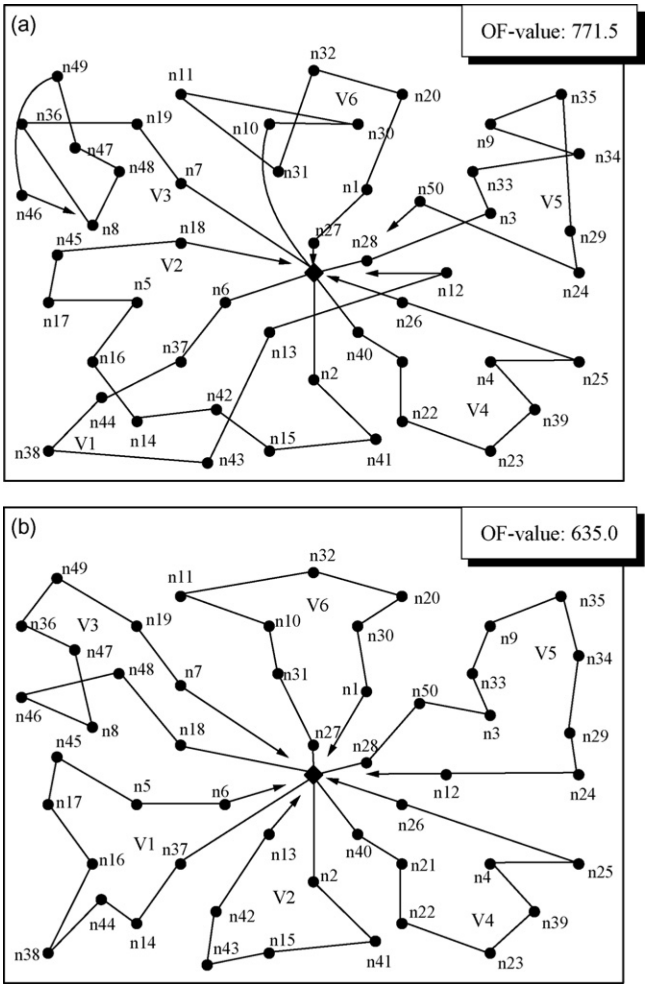

Fig. 7. (a) Initial solution for example R-104.50 and (b) best solution found for example R-104.50.

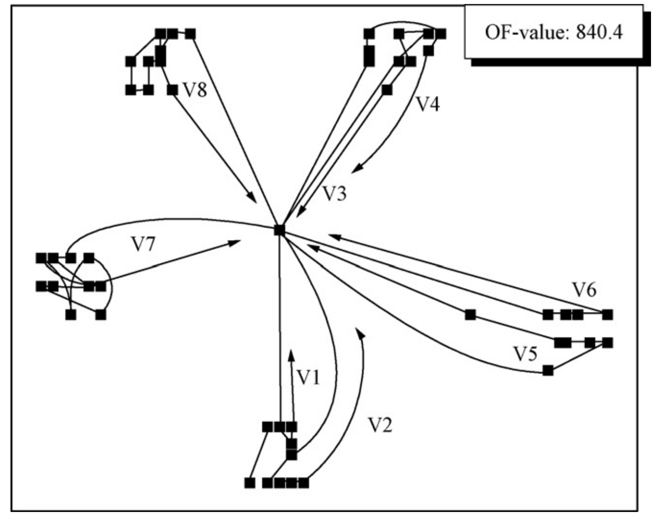

Fig. 8. Best solution found for example RC-102.50.

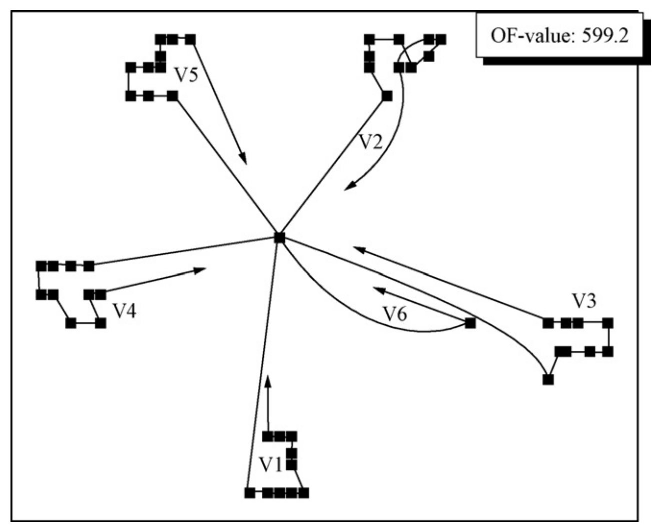

Fig. 9. Best solution discovered for example RC-108.50.

starting point for each problem is given in Table 1. Suboptimal gaps as large as 30.6% have been significantly diminished and sometimes even vanish at surprisingly low computational cost. The final suboptimal gap drops, on average, from 17.5% to less than 1.8%. By comparing our final results with the best reported solutions compiled by Kallehauge et al. (2001), it is concluded that near-optimal ones were discovered for the 50-node version of the following Solomon's RC-benchmark problems: RC-102.50, RC-103.50, RC-108.50, all involving the first 50 nodes (see Table 1 and Figs. 8 and 9). In such cases, the total distance cost is close to the best reported in the literature. However, the number of used vehicles is slightly higher but still smaller than the one initially available (see Table 1). In any case, the model parameters were adopted as follows: $\varphi_1 = 0.3$, $\varphi_0 = 0.90$ (normal mode) and $\varphi_0 = 0.60$ (perturbation mode). The optimal tours for such benchmark problems are fully described in Table 2. Variations of the objective function and the CPU time after each turn of the RAS for problems RC-102.50 and RC-108.50 are

Table 1
Improving non-optimal solutions reported in Dondo and Cerdá (2007).

Problem	Nodes	Vehic	Best known solutions	Initial solution	Final solution	Suboptimality gap (%)		CPU time[a]
						Initial	Final	
R-102	25	6	547.9	623.2	547.9	14.9	0.0	3.55
R-103	25	5	454.6	478.5	463.3	5.2	1.9	5.03
R-104	50	6	625.4	771.5	635.0	23.4	1.5	75.90
R-107	25	5	425.3	483.2	444.1	13.6	4.2	4.76
R-111	25	5	428.9	517.3	436.1	19.3	1.4	17.06
R-112	25	4	394.0	428.9	402.8	8.9	2.2	13.30
RC-102	50	8	822.5	1004.9	840.4	22.2	2.2	36.98
RC-103	50	6	710.9	928.4	730.0	30.6	2.6	62.54
RC-108	50	6	598.1	716.5	599.2	19.8	0.1	292.08

[a] Seconds in a 2.0 GHz 1 GB-Ram Pentium IV PC using ILOG OPL Studio.

Table 2
Near-optimal solutions for some 50-node VRPTW benchmark problems.

Vehicle	Route	Used capacity	Travel time	Traveled distance
Example R-104 (50 nodes)				
V1	D n37 n14 n44 n38 n16 n17 n45 n5 n6 D	128	218.1	117.2
V2	D n2 n41 n15 n43 n42 n13 D	55	180.2	84.3
V3	D n18 n48 n46 n8 n47 n36 n49 n19 n7 D	142	228.7	124.5
V4	D n40 n21 n22 n23 n39 n4 n25 n26 D	140	225.6	101.1
V5	D n28 n50 n3 n33 n9 n35 n34 n29 n24 n12 D	122	226.7	114.4
V6	D n27 n31 n10 n11 n32 n20 n30 n1 D	134	176.6	96.5
Example RC-102(50 nodes)				
V1	D n22 n49 n20 D	60	167.1	76.7
V2	D n19 n23 n21 n18 n48 n25 n48 D	140	219.5	106.9
V3	D n39 n36 n40 n38 n41 D	130	157.6	90.8
V4	D n42 n44 n43 n35 n37 D	70	214.2	94.7
V5	D n33 n26 n28 n30 n32 n50 D	100	202.1	121.8
V6	D n34 n31 n29 n27 D	80	154.4	114.4
V7	D n14 n47 n11 n15 n16 n9 n10 n13 n17 n12 D	200	229.1	129.1
V8	D n3 n1 n45 n5 n8 n7 n6 n46 n4 n2 D	190	206.0	106.0
Example RC-108 (50 nodes)				
V1	D n25 n23 n21 n48 n18 n19 n49 n20 n22 n24 D	200	203.8	103.8
V2	D n41 n42 n44 n43 n40 n38 n37 n35 n36 n39 D	200	232.8	104.4
V3	D n33 n32 n30 n28 n26 n27 n29 n31 n34 D	150	217.3	127.3
V4	D n12 n14 n47 n17 n16 n15 n13 n9 n11 n10 D	200	197.2	97.2
V5	D n2 n6 n7 n8 n46 n45 n4 n5 n1 n3 D	190	211.1	95.9
V6	D n50 D	30	128.3	70.6

depicted in Figs. 6(c)–(d), respectively. The algorithm was stopped after five RAS turns for problem RC-108.50 while six rotations were required for problems RC-102.50. In every case, most of the tour improvements were obtained through 2–3 successful applications of the normal mode. Instead, the perturbation mode only yielded a moderate cost saving for problem RC-102.50. The best solutions for problems RC-102.50 and RC-108.50 were found in 36.92 s and 292.08 s, respectively.

9.4. Solutions to 100-node Solomon's benchmark problems

Tables 3 and 4 present the results found for a set of nineteen 100-node R/RC/C benchmark problems involving from 7 to 19 vehicles (Solomon, 1987). For each problem, the adopted values for the model parameters at normal and perturbation modes, the final solution, and the required CPU time are all included in Table 3. To sweep the whole geographical area to be serviced by the vehicle

Table 3
Best solutions discovered and CPU time requirements for some large-scale VRPTW examples.

Example	Initial solution	Final solution	Total CPU time[a] (s)	Procedure parameters						
				Normal mode				Perturbation mode		
				φ_0	φ_1	Ω	T	φ_0	Ω	T
100-node Solomon's examples										
C-103	843.0	827.8	314.2	1.2	0.4	π/2	60	0.4	π/3	60
C-104	843.0	824.9	391.6	1.2	0.4	π/2	60	0.4	π/3	60
RC-102	1480.7	1420.2	1276.3	0.9	0.3	π/2	60	0.4	π/4	60
RC-103	1554.9	1344.2	664.9	0.9	0.3	π/2	60	0.4	π/4	60
RC-104	1508.0	1110.2	675.9	0.9	0.3	π/2	60	0.4	π/4	60
RC-105	1906.5	1576.5	1237.8	1.2	0.3	π/2	60	0.3	π/3	60
RC-106	1548.2	1330.8	731.6	0.8	0.2	π/2	60	0.2	π/4	60
R-102	1846.9	1484.4	263.2	0.6	0.2	π/2	120	0.3	π/4	120
R-103	1488.0	1242.4	1206.3	0.4	0.1	π/2	120	0.4	π/4	120
R-104	1175.3	1059.9	838.2	0.5	0.1	π/2	60	0.3	π/4	60
R-105	1492.4	1374.9	1059.3	0.5	0.1	π/2	60	0.3	π/4	60
R-106	1401.5	1269.0	451.5	0.5	0.1	π/2	60	0.3	π/4	60
R-107	1220.2	1156.4	1332.1	0.9	0.3	π/2	120	0.4	π/4	120
R-108	1162.4	1002.2	683.6	0.9	0.3	π/2	60	0.4	π/4	60
R-109	1357.8	1245.2	261.6	0.9	0.3	π/2	60	0.4	π/4	60
R-110	1219.9	1129.5	215.3	0.9	0.3	π/2	60	0.4	π/4	60
R-201	1815.0	1216.7	1565.5	0.3	0.1	π/2	180	0.2	π/5	180
R-202	1216.0	1104.6	3458.8	0.3	0.1	π/2	180	0.2	π/5	180
R-203	1155.0	957.0	2329.6	0.3	0.1	π/2	180	0.2	π/5	180
200-node Homberger's examples										
R1_2_2	5090.5	4217.7	6399.6	0.3	0.1	π/4	180	0.2	π/8	180
R1_2_3	4801.4	3707.3	3232.0	0.3	0.1	π/4	180	0.2	π/8	180
R1_2_4	3994.8	3545.0	5805.0	0.3	0.1	π/4	180	0.2	π/8	180
R1_2_5	4794.6	4329.2	2285.0	0.3	0.1	π/4	180	0.2	π/8	180

[a] Seconds in a 2.0 GHz 1 GB-Ram Pentium IV PC using ILOG OPL Studio.

Table 4
Comparison with the best known solutions reported in the literature.

Example	Best known solution			Our best solution		
	Vehicles	Distance	Refs.	Vehicles	Distance	Subopt. gap %
100-node Solomon's examples						
C-103	10	826.8	Kallehauge et al. (2001)	10	827.8	0.1
C-104	10	824.9	Kallehauge et al. (2001)	10	824.9	–
RC-102	14	1457.5	Kallehauge et al. (2001)	14	1420.2	–
RC-103	11	1258.2	Kallehauge et al. (2001)	13	1344.2	6.8
RC-104	10	1135.5	Kallehauge et al. (2001)	12	1110.2	–
RC-105	15	1513.7	Kallehauge et al. (2001)	16	1576.5	4.1
RC-106	11	1424.7	Berger et al. (2004)	15	1330.8	–
R-102	18	1466.6	Kallehauge et al. (2001)	19	1484.4	1.2
R-103	14	1208.7	Kallehauge et al. (2001)	14	1242.4	2.8
R-104	9	1007.2	Mester et al. (2005)	12	1059.9	5.2
R-105	15	1355.3	Kallehauge et al. (2001)	16	1374.9	1.4
R-106	13	1234.6	Kallehauge et al. (2001)	14	1269.0	2.8
R-107	13	1064.6	Kallehauge et al. (2001)	13	1156.4	8.6
R-108	12	960.9	Rousseau, Gendreau, and Pesant (2002)	14	1002.2	4.3
R-109	13	1146.9	Kallehauge et al. (2001)	14	1245.2	8.6
R-110	12	1068.0	Kallehauge et al. (2001)	12	1129.5	5.7
R-201	8	1143.2	Kallehauge et al. (2001)	7	1216.7	6.4
R-202	3	1191.7	Rousseau et al. (2002)	7	1104.6	–
R-203	3	939.5	Mester et al. (2005)	7	957.0	1.8
200-node Homberger's examples						
R1_2_2	18	4054.4	Mester and Bräysy (2005)	20	4217.7	4.0
R1_2_3	18	3382.6	Mester and Bräysy (2005)	19	3707.3	9.6
R1_2_4	18	3067.9	Mester and Bräysy (2005)	19	3545.0	9.0
R1_2_5	18	4112.9	Mester and Bräysy (2005)	20	4329.2	5.0

fleet, four locations of the rotating angular sector ($\Omega = \pi/2$) were considered when applying the normal mode. The number of angular sectors into which the service region is partitioned rises to 8 ($\Omega = \pi/4$) when the perturbation mode is activated. The starting points also reported in Table 3 were always generated by the route construction procedure of Dondo and Cerdá (2007), sometimes supported by a post-processing stage to remove, if any, time constraint violations. On average, the CPU time required to discover the final solutions reported in Table 3 amounts to 751.5 s, while the suboptimality gap is less than 3.1%. As expected, problems of type-2 (R-201, R-202 and R-203) featuring wider time windows were the most time-consuming. In such cases, the geographical region has been divided into 10 sectors rather than 8 when applying the perturbation mode. Though the total run time amounts to 3458.8 s, most of the transportation cost savings (almost 84%) in problem R-202 was achieved in 371 s by making a single RAS-rotation in normal mode (see Fig. 6f). On average, the relative contributions of subproblems I–III to the total computer time for 100-node benchmark problems were, on average, 30%, 20% and 50%, respectively.

The parameter T in Table 3 stands for the maximum CPU time available to solve the VRPTW mathematical formulation under normal or perturbation mode at each location $\Omega^{(m)}$. The adopted value of T ranges from 60 s to 120 s in normal/perturbation mode. In other words, the MILP branch-and-bound procedure applied to solve subproblems I–III was stopped after reaching the time limit T. Usually, the time required to solve the MILP model is much lower than T except for R-problems of type-2 while solving subproblem III. In such cases, the value of T has been increased to 180. Comparison of the results with the best solutions reported in the literature based on the number of used vehicles and the total travel distance is shown in Table 4. For three of the problems (RC-102, RC-104, R-202), new better solutions were identified (see Tables 4 and 5 and Figs. 10 and 11). Note that the new better solutions use a few more vehicles than the best known solutions. Table 4 also includes the sources reporting the best solutions for such Solomon's problems.

9.5. Solutions to examples with 200 nodes and multiple depots

Similarly to the set of 100-node examples proposed by Solomon (1987), a new family of benchmark problems involving a much higher number of customers, a homogeneous vehicle fleet and a single depot was introduced by Homberger and Gehring (1999). Problems with 200, 400, 600, 800 and 1000 customers are available. The new problem instances have been designed following the same guidelines used to define Solomon's 100-node benchmark problems. Therefore, each one is identified by: the fleet size, the common vehicle capacity, travel distances between nodes, the customer spatial distribution and the time-window width distribution. Likewise Solomon's problems, the new instances are divided into three categories: C, R and RC. Two problem sets have also been proposed for each category: problems of type-1 assume narrow time

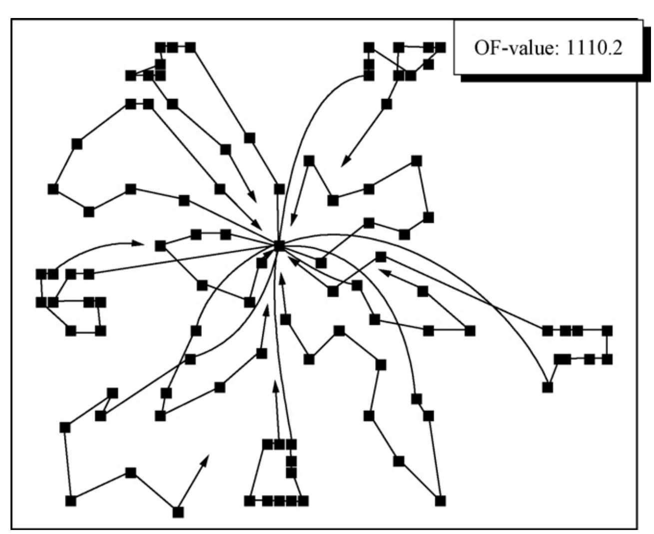

Fig. 10. Best solution discovered for example RC-104.100.

Table 5
New best solutions found for some 100-node VRPTW problems.

Vehicle	Route	Capacity	Travel time	Traveled distance
Example RC-102 (100 nodes)				
V1	D n85 n63 n76 n51 n84 n56 D	95	154.0	80.6
V2	D n90 n66 D	22	166.2	29.2
V3	D n2 n45 n1 n3 n5 n8 n46 n4 n100 n70 D	176	224.1	102.1
V4	D n14 n47 n73 n79 n6 n7 n55 D	111	188.1	114.6
V5	D n11 n15 n16 n9 n10 n13 n17 n12 n82 D	189	233.6	114.9
V6	D n42 n44 n61 n81 n54 n68 D	76	184.0	94.6
V7	D n48 n21 n23 n19 n18 n22 n49 n20 n24 n25 D	200	224.0	118.1
V8	D n62 n29 n30 n32 n89 D	58	227.2	133.1
V9	D n91 n92 n94 n67 n71 n93 n96 n80 D	115	232.5	74.2
V10	D n52 n99 n87 n59 n97 n75 n58 D	121	228.9	115.0
V11	D n39 n36 n40 n38 n41 n43 n35 n37 n72 D	188	239.5	130.6
V12	D n65 n69 n88 n53 n78 n60 n98 D	120	202.7	91.1
V13	D n33 n28 n27 n26 n31 n34 n50 n95 D	156	207.4	123.8
V14	D n64 n57 n86 n74 n77 n83 D	97	229.1	98.3
Example RC-104 (100 nodes)				
V1	D n85 n63 n89 n76 n51 n84 n56 n64 n66 D	153	193.4	103.4
V2	D n52 n59 n87 n97 n75 n58 n77 D	113	225.5	126.1
V3	D n69 n98 n53 n82 n65 n90 D	72	160.9	50.4
V4	D n88 n60 n78 n73 n79 n7 n6 n55 D	154	179.0	99.0
V5	D n12 n14 n15 n11 n10 n9 n13 n16 n17 n47 D	200	234.4	111.6
V6	D n68 n70 n1 n3 n5 n45 n8 n46 n4 n2 n100 D	197	209.2	94.2
V7	D n20 n49 n19 n18 n48 n23 n21 n25 n24 n22 D	200	221.4	85.9
V8	D n33 n32 n30 n28 n26 n27 n29 n31 n34 n94 n91 D	186	235.9	125.9
V9	D n80 n96 n93 n71 n72 n54 n81 n61 D	99	159.3	79.3
V10	D n99 n86 n74 n57 n83 D	88	200.9	61.0
V11	D n42 n44 n43 n38 n37 n35 n36 n40 n39 n41 D	200	239.8	102.5
V12	D n92 n95 n62 n50 n67 D	62	123.7	70.9
Example R-202 (100 nodes)				
V1	D n83 n45 n8 n18 n89 D	63	774.0	62.8
V2	D n27 n69 n31 n88 n62 n30 n90 n10 n32 n70 n1 D	155	831.3	112.8
V3	D n39 n67 n23 n75 n72 n73 n21 n40 n53 n12 n56 n74 n54 n4 n55 n25 n26 D	266	873.2	194.1
V4	D n42 n15 n57 n87 n2 n22 n41 n43 n97 n13 n58 D	136	868.1	131.8
V5	D n46 n36 n47 n63 n11 n19 n64 n49 n48 n7 n52 n82 D	195	587.0	177.2
V6	D n28 n33 n65 n71 n29 n76 n3 n79 n78 n81 n9 n51 n20 n66 n35 n34 n68 n24 n80 n77 n50 D	303	899.9	250.5
V7	D n94 n95 n59 n92 n37 n14 n38 n44 n16 n61 n86 n85 n99 n96 n6 n5 n84 n17 n91 n100 n98 n93 n60 D	358	872.4	175.4

windows and small vehicle capacity, while type-2 problems involve wider time windows and a larger vehicle capacity. For instance, R1_2_2 is a R-class problem of type 1/variant-2 (more stringent time windows) with 200 nodes. Four 200-node benchmark problems were solved. In every case, the initial solution was found by using the three-phase hybrid construction algorithm of Dondo and Cerdá (2007). In some cases, however, especially if an infeasible solution has been generated, a post-processing stage was subsequently applied to eliminate time-window violations and/or tour overdurations before starting the improvement process. The selected values for the parameters (φ_1, φ_0, T) are given in Table 3. The geographical region to be serviced was partitioned into eighth sectors (normal mode) or sixteen sectors (perturbation mode) to reduce the size of the MILP formulation to be solved. Numerical results for R-class examples of type-1 all involving 200 nodes, a homogeneous fleet of variable size and a random distribution of customer locations are shown at the bottom of Table 3. Comparison of the solutions found with the best ones reported in the literature is made in Table 4. The suboptimality gap is, in all cases, less than 10%. The toughest

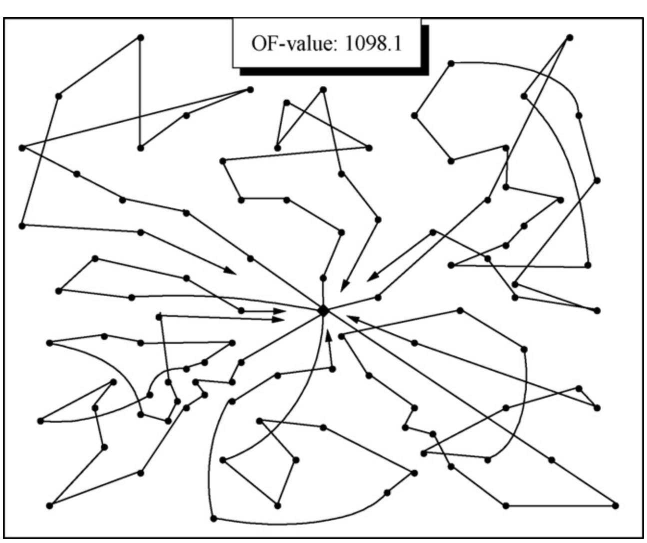

Fig. 11. Best solution discovered for example RC-202.100.

Table 6
Worst-case and average performances of the proposed methods for 100-node and 200-node VRPTW problems of different types.

Problem type	Worst case performance (gap %)	Average performance (gap %)
100-node Solonon's problems		
C-1	0.1	0.05
RC-1	6.8	0.08
R-1	8.6	4.51
R-2	6.4	0.30
200-node Homberger's problems		
R1_2	9.6	6.90

Table 7
Comparing the performance of the proposed approach with several available VRPTW (metha)heuristic techniques.

Problem type	GTA	RT	TB	PHGA	B-VH	CW	DC[a]
C1							
Vehicles	10.00	10.00	10.00	10.00	10	10.00	10.00
Distance	828.38	828.45	828.45	828.50	828.95	834.05	826.35
CPU time	1800	3200	14630	1800	1800	649	353
RC1							
Vehicles	11.92	12.33	11.90	11.88	11.88	12.12	14.00
Distance	1388.13	1269.48	1381.31	1414.86	1456.49	1388.15	1356.38
CPU time	1800	2600	11264	1800	1800	2900	892
R1							
Vehicles	12.38	12.58	12.33	12.17	12.08	12.50	14.22
Distance	1210.83	1197.42	1220.35	1251.40	1288.35	1241.89	1218.21
CPU time	1800	2700	13774	1800	1800	1382	701

[a] This work.

benchmark problem R1_2_4 reported in the paper featuring wider time windows requires a total running time of 2806.8 s, i.e. near 46 min.

To test the algorithm performance when tackling a multi-depot heterogeneous fleet VRPTW problem, the Solomon's R-104 problem was modified and solved. The new instance of problem R-104(m) includes three depots and three different types of vehicles: small trucks with 100 units of capacity, medium size vehicles (200 units) and large trucks (300 units). The objective function initially equals to 1548.8 drops to 1069.9 in 1256 s of CPU time. The solution is depicted in Fig. 12. Continuous, dashed and dotted lines were used to depict tours involving large, medium and small vehicles, respectively.

9.6. Performance analysis

Table 6 summarizes the worst case and the average performances for the proposed VRPTW improvement algorithm, both measured in terms of the suboptimal gap with respect to the best solutions reported in the literature. Such performance measures were obtained by considering the final results found for 100-node C1/RC1/R1 Solomon's problems and 200-node R1_2 Homberger's problems. Accounting for the average performance values shown in Table 4, it can be concluded that the behavior of the proposed algorithm is quite satisfactory for the set of 100-node Solomon's problems. Even in the most unfavorable 200-node problem series (i.e. R1_2), the worst case gap lies below the 10% threshold.

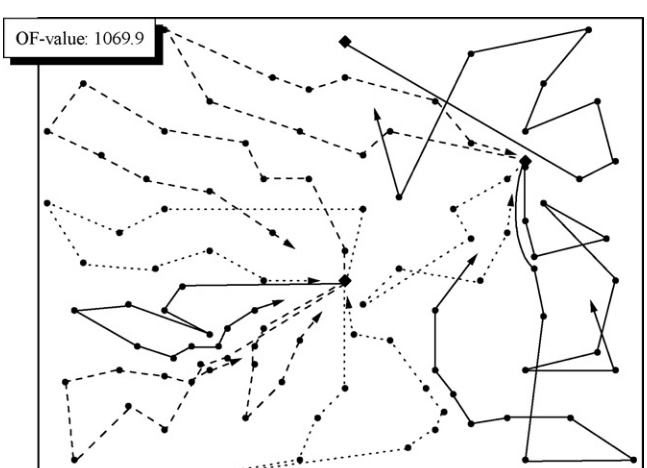

Fig. 12. Best solution found for the 3-depot R-104(m).100 problem.

In turn, Table 7 compares our method with a number of the best available heuristic techniques in terms of three key parameters: average number of used vehicles, average traveled distance and average CPU time. To make the comparison, just the computational results reported for 100-node Solomon's problems were considered. The reference techniques included in Table 7 are: Ant Colony System-GTA (Gambardella et al., 1999), Tabu Search-RT (Rochat & Taillard, 1995), Threshold Method-TB (Taillard et al., 1997), Parallel Hybrid Genetic Algorithm-PHGA (Berger, Barkaoui, & Bräysy, 2004), Large Neighborhood Search-BVH of Bent and Van Hentenryck (2004) and the *k*-exchange reduction-CW method (Cordone & Wolfer-Calvo, 2001). It can be observed that our algorithm performs extremely well in terms of traveled distance and CPU time but uses a slightly higher number of vehicles. Overall, it looks very competitive with regards to other heuristic algorithms in both solution quality and computer time.

10. Conclusions

Transportation costs for shipping products from depots to customers heavily depend on the proper selection of routes and schedules for the vehicle fleet providing such delivery services. A novel MILP improvement framework for large-scale time window-constrained vehicle routing problems involving heterogeneous fleets and multiple depots has been developed. It is a local search approach that fully explores a rather large neighborhood around the current solution so as to provide a better set of vehicle tours in an efficient manner. To achieve this goal, the approach relies on two key building blocks: a spatial decomposition scheme and a new MILP mathematical representation for the VRPTW improvement problem. A significant reduction in the problem size is obtained by properly adopting a single parameter Ω 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. Therefore, every sector comprises a subset of entire tours and some of them may simultaneously belong to a pair of adjacent sectors. As a result, customer exchanges between neighboring zones are also considered. In addition, a pair of parameters (φ_0, φ_1) permits to define the set of feasible improvement moves on every sector, i.e. the neighborhood to be explored. The spatial decomposition scheme is heuristically supported by the fact that improving actions mostly take place among nearby routes.

On the other hand, the proposed MILP problem formulation allows to efficiently exploring a rather large solution space around the starting point on every sector by accounting for all kinds of improvement moves. To get MILP-models of moderate size, the improvement moves have been classified into two groups: (1)

balanced/non-balanced node string exchanges between neighboring tours and (2) node string relocations on every individual tour. In this way, the VRPTW improvement problem for each zone can be decomposed into a pair of lower-size MILP subproblems that are sequentially solved, using the best solution for the other subproblem (the normal mode) as the initial point. Soft time-window and tour overdurations can be handled by including penalty terms in the objective function to also account for promising non-feasible routes. However, this sequential search scheme can be trapped on a local optimum. If so, the complete MILP problem formulation simultaneously considering both groups of improvement moves should be tackled (the perturbation or mixed-mode). By properly adjusting the parameters (Ω, φ_0, φ_1,) defining (a) the number of sectors to be independently explored, and (b) the subset of complete tours and the set of feasible exchange moves to be considered on each zone, the problem size and the required CPU time for the mixed-mode MILP-formulation both remain under control. Nonetheless, it is adopted an upper limit T on the model solution time for any individual sector beyond that the branch-and-bound procedure is stopped. If the CPU time constraint is binding, the incumbent solution at time T is stored and used to define the neighborhood to be explored on the next major iteration.

The starting point for the proposed VRPTW improvement technique is provided by the incomplete optimization algorithm developed by Dondo and Cerdá (2007). This cluster-based route construction algorithm not only optimizes the set of vehicle routes at the level of clusters but also provides the best tour-depot assignments. It proved to be very robust to tackle C-class problems with up to 100 clustered nodes. However, only low-size RC and R-class problems involving 25–50 nodes can be solved at reasonable computer times but the final solutions feature rather large suboptimal gaps. This is why the performance of our VRPTW improvement algorithm in terms of solution quality and CPU time has been mostly evaluated by solving RC-class and R-class Solomon's benchmark problems comprising up to 100-nodes with either a single central depot or multiple depots. In the latter case, tour-depot assignments at the starting point are kept without changes throughout the improvement process. On average, the proposed improvement algorithm converges to a good feasible solution with a suboptimality gap less than 3.3%. To test its performance with multi-depot heterogeneous fleet VRPTW examples, a modified version of Solomon's R-104 problem including two additional depots and three different vehicle capacities was solved. Moreover, the algorithm has also been applied to four VRPTW problem instances involving 200 nodes introduced by Homberger and Gehring (1999).

Since the distribution problem is indeed dynamic in nature, the VRPTW problem data changes with time. Unexpected events such as traffic jam, new customer orders or vehicle malfunctioning make the best routing obtained with static data infeasible or at least a non-optimal one. Solution to the VRPTW problem within a dynamic environment will be studied on a next paper.

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. Nomenclature

Sets

I	set of transportation requests
V	set of available vehicles

Binary variables

X_{ij}	sequencing binary variable denoting that node i is visited before ($X_{ij} = 1$) or after node j ($X_{ij} = 0$)
Y_{iv}	assignment variable that controls the allocation of node i to vehicle v

Continuous variables

AT_i	vehicle arrival time at node i
C_i	routing cost from the depot to node i
E_i	violation of ith-time window (earliness) due to arrivals before a_i
L_i	violation of ith-time window (tardiness) due to arrivals later than b_i
OC_v	overall variable cost for vehicle v
OD_v	overduration of trip v
OT_v	overall routing time for vehicle v

Parameters

α_v	unit-time penalty for vth-tour overduration.
β_i	unit-time penalty cost for early arrival at node i
γ_i	unit-time penalty cost for late arrival at node i
ε	small positive scalar for the stopping criterion
φ_1	angle between the delimiting rays of the cone defining fixed nodes
φ_0	angle between the delimiting rays of the cone defining mobile nodes
θ_{iv}	angle between the v-trip axis and the ray connecting node i to the depot
Ω	angle between the delimiting rays of the rotating angular sector (RAS)
a_i	earliest service start time at node i
b_i	latest service start time at node i
c_{ij}^v	traveling cost between nodes i and j along the trip v.
d_{mx}	maximum distance between a node and the center of gravity of a candidate tour
q_v	maximum capacity of vehicle v
st_i	service time at node i
t_{ij}^v	travel time between nodes i and j along the trip v.
tv_v^{max}	maximum allowed service time for vehicle v
T	CPU time limit
l_i	amount of load to pick up at node i

References

Ahuja, R. K., Ergun, O., Orlin, J. B., & Punnen, A. P. (2002). A survey of very large scale neighborhood search techniques. *Discrete Applied Mathematics*, *123*, 75–102.

Ball, M., Magnanti, T., Monma, C., & Nemhauser, G. (1995). *Network routing* Amsterdam: Elsevier Science.

Bent, R., & Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, *38*(4), 515–530.

Berger, J., Barkaoui, M., & Bräysy, O. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, *31*(12), 2037–2053.

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.

Bräysy, O., Dullaert, W., & Gendreau, M. (2004). Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. *Journal of Heuristics*, *10*(6), 587–611.

Cordone, R., & Wolfer-Calvo, R. (2001). A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics*, *7*(2), 107–129.

Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, *176*(3), 1478–1507.

Fagerholt, K. (2004). A computer-based decision support system for vessel fleet scheduling: experience and future research. *Decision Support Systems*, *37*(1), 35–47.

Gambardella, L. M., Taillard, E., & Agazzi, G. (1999). MACS-VRPTW: a multiple ant colony systems for vehicle routing problems with time windows. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 63–76). London: McGraw-Hill.

Gendreau, M., Hertz, A., & Laporte, G. (1992). New insertion and post-optimization procedures for the traveling salesman. *Operations Research*, *40*, 1086–1094.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, *40*, 1276–1290.

Gendreau, M., Laporte, G., & Potvin, J. Y. (1994). *Local search algorithms for the vehicle routing problem*. Technical report, Centre de Recherche sur les Transport, Universite de Montreal.

Gillet, B., & Miller, L. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, *22*, 340–349.

Glover, F. W., & Laguna, M. (1997). *Tabu search*. Kluwer Academic Publishers.

Homberger, J., & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR*, *37*, 297–318.

Jetlund, A. S., & Karimi, I. A. (2004). Improving the logistics of multi-compartment chemical tankers. *Computers and Chemical Engineering*, *28*, 1267–1283.

Kallehauge, B., Larsen, J., & Madsen, O. (2001). *Lagrangean duality applied on vehicle routing problems with time windows*. Experimental results. Technical report IMM-TR-2001-9.

Laporte, G., Gendreau, M., Potvin, J. Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, *7*(4–5), 285–300.

Lin, S., & Kernighan, B. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, *21*, 2245–2269.

Mester, D., & Bräysy, O. (2005). Active guided evolution strategies for the large scale vehicle routing problems with time windows. *Computers & Operations Research*, *32*(6), 1593–1614.

Mester, D., Bräysy, O., & Dullaert, W. (2005). *A multi-parametric evolution strategies algorithm for vehicle routing problems*. Working paper, Institute of Evolution, University of Haifa, Israel.

Or, I., 1976. Ph.D. Thesis, Dept of Industrial Engineering and Management Sciences, Northwestern University.

Osman, I. H., & Laporte, G. (1996). Meta-heuristics: a bibliography. *Annals of Operations Research*, *63*, 513–623.

Papadimitrou, C., & Steiglitz, K. (1982). *Combinatorial optimization*. Englewood Cliffs, NJ: Prentice-Hall.

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

Rousseau, L., Gendreau, M., & Pesant, G. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, *8*(1), 43–58.

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

Taillard, É. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, *23*, 661–673.

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

Tarantilis, C. D., & Kiranoudis, C. T. (2007). A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector. *European Journal of Operational Research*, *179*(3), 806–822.

Thompson, P., & Psaraftis, H. (1993). Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Operations Research*, *41*, 935–946.

Van Breedam, A. (1994). PhD thesis, Faculty of Appl. Eco., University of Antwerpen.

Van der Bruggen, L., Lenstra, J., & Schuur, P. (1993). Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science*, *27*, 298–311.

Van der Bruggen, L., Gruson, R., & Salomon, M. (1995). Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research*, *81*, 460–473.

Xu J., & Kelly J. (1995). *A robust network flow-based tabu search approach for the vehicle routing problem*. Technical report, Graduate School of Business, University of Colorado.