Noname manuscript No.
(will be inserted by the editor)

Tabu Search Techniques for the Heterogeneous Vehicle Routing Problem with Time Windows and Carrier-Dependent Costs

Sara Ceschia · Luca Di Gaspero · Andrea Schaerf

the date of receipt and acceptance should be inserted later

Abstract In this work we formalize a new complex variant of the classical vehicle routing problem arising from a real-world application. Our formulation includes a heterogeneous fleet, a multi-day planning horizon, a complex carrier-dependent cost for vehicles, and the possibility of leaving orders unscheduled.

For tackling this problem we propose a metaheuristic approach based on Tabu Search and on a combination of neighborhood relations. We perform an experimental analysis to tune and compare different combinations, highlighting the most important features of the algorithm.

The outcome is that a significant improvement is obtained by a complex combination of neighborhood relations.

In addition, we compare our solver with previous work on public benchmarks of a similar version of the problem, namely the *Vehicle Routing Problem with Private fleet and Common carrier*. The conclusion is that our results are competitive with the best ones in literature.

1 Introduction

Vehicle routing is one of the most studied problems in optimization (see, e.g., Toth and Vigo, 2002). Many variants of the *Vehicle Routing Problem* (VRP) have been introduced in the literature over the years, ranging from multi-depot, to time windows, to mixed fleet, just to name a few. Nevertheless, despite the availability of this large set of classified formulations, often the practical problem that companies have to face is more complex than the standardized version discussed in scientific articles.

This is the case of the problem we came across, and thus in this work, we consider a new version of the VRP problem. We decided to deal with its exact real-world formulation, without any concession to “judicious simplification”, that would have allowed us to borrow results from existing successful solution techniques.

Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf
DIEGM, University of Udine
via delle Scienze 208, I-33100, Udine, Italy
E-mail: {sara.ceschia,l.digaspero,schaerf}@uniud.it

Our formulation, explained in detail in Section 2, includes a heterogeneous fleet, a multi-day planning horizon, a complex carrier-dependent cost function for vehicles, and the possibility of leaving orders unscheduled.

The problem formulation includes some non-linear constraints and cost components, thus the use of exact methods for its solution is quite impractical. Therefore, we resort to a metaheuristic technique, namely Tabu Search (Glover and Laguna, 1997, Hoos and Stützle, 2005), that has shown to be effective on other variants of VRP. We also make use of a combination of different neighborhood relations. The experimental analysis is carried out on a set of real-world instances, and makes use of principled statistical tests to tune the parameters and to compare different variants.

The final outcome of the experimental analysis is that the most promising techniques are obtained by a combination of different neighborhood structures.

All the instances employed in the experiments, along with the best solutions found by our methods, are available on the web at the URL `http://www.diegm.uniud.it/ceschia/index.php?page=vrptwcdc`.

In order to evaluate objectively the performance of our solver, we also test it on public benchmarks of the *Vehicle Routing Problem with Private Fleet and Common Carrier* (VRPPC) (Bolduc et al, 2007), which significantly resembles our problem and allows the comparison with other approaches. The outcomes of these comparisons show that our results are at the same level of the best ones in literature and we have been able to obtain a new best-known solution for one case.

The paper is organized as follows. In Section 2 we present the problem formulation and in Section 3 we discuss related work. The application of Tabu Search to the problem is illustrated in Section 4. Section 5 shows the experimental analysis on our instance and on benchmarks of the VRPPC. Finally, in Section 6 we draw some conclusions and discuss future work.

2 Problem Formulation

We present our problem in stages by showing one by one the features (either previously published or original) that are included in the formulation and the various costs associated with the use of the vehicles.

2.1 Features of the Problem

We describe our formulation starting from the basic version of VRP, the *Capacitated* VRP (Toth and Vigo, 2001), which is characterized by the following entities and constraints:

Customers/Orders: The basic entity of the problem is the *customer*, who requires a supply of goods, called an *order*.

More formally we are given a set of n orders $\mathcal{O} = \{1, \ldots, n\}$, each issued by the corresponding customer. Multiple orders by the same customer are grouped together, so that in this basic formulation orders and customers are indistinguishable. As a consequence, in the following we will use the terms customer and order interchangeably unless stated explicitly.

A special customer, denoted by the number 0, represents the depot of the transportation company.

Each order i has associated a *demand* $q_i \geq 0$, which is the amount of goods to be supplied.

Fleet: The transportation of goods is performed by a fleet of vehicles $\mathcal{F} = \{1, \ldots, m\}$. In the original formulation all the vehicles are identical (i.e., they have the same capacity Q) and they are located at the same central depot (called *home depot*), where they have to return upon complete delivery.

Routes: A *vehicle route* (or simply a *route*) r is a sequence $\langle 0, v_1, \ldots, v_l, 0\rangle$ starting at the depot, visiting customers $v_1, \ldots, v_l \in \mathcal{O}$ in that order, and returning back to the depot. The orders served by a route r, is the set $\{v_1, \ldots, v_l\}$, which will be denoted by $ord(r)$. It is useful to define the *predecessor* $\pi(i, r)$ of a customer i w.r.t. the route r, as the previous customer in the sequence r.

We allow the possibility of empty routes, that is $r = \langle 0, 0\rangle$. In those cases, the vehicle is not used.

Load Limits: An important constraint is that the load of each vehicle assigned to a route cannot exceed the vehicle capacity. If we define $q(r) = \sum_{i \in ord(r)} q_i$ as the total demand of route r, we impose that $q(r) \leq Q$.

Transportation Costs: Each route has associated a transportation cost, denoted by $t(r)$. It can be either the road distance or a different measure of the total expenses of going on a given way from one customer to the following one (e.g., time, tolls, $\ldots$).

The solution of a VRP calls for the determination of a set of routes $\mathcal{R} = \{r_1, \ldots, r_m\}$, (also called a *routing plan*), one for each vehicle, that minimize the total transportation cost $\sum_{j=1}^{m} t(r_j)$ and additionally fulfills the following constraints:

1. the routes satisfies all orders, i.e., $\bigcup_{j=1}^{m} ord(r_j) = \mathcal{O}$;
2. each customer is visited only once, i.e., $ord(r_i) \cap ord(r_j) = \emptyset, 1 \leq i < j \leq n$;
3. the demands of all orders are fulfilled[1].

The first extensions to the problem that we consider are represented by the so-called *Service Times* and *Time Windows*, discussed by Solomon (1987):

Service Times: Each order is associated with a service time $s_i \geq 0$ needed to unload the goods from the vehicle. The vehicle must stop at the customer location for the service time.

Travel Times: The time for traveling from one customer i to another customer j is estimated by a travel time τ_{ij}.

Time Windows: Each customer and the depot are associated with a time interval $[e_i, l_i]$ (called *time window*) in which the service should take place. The depot time window includes all the time windows of the customers.

Earliest service time: All vehicles leave at the start time of the depot window (usually set to 0), and in case of early arrivals at the location of each customer, the vehicle is required to wait until the service can start.

More formally, given a route $r = \langle 0, \ldots, j, i, \ldots, 0\rangle$, the *earliest service time* of order i on r is defined by $\alpha(i, r) = \max\{e_i, \delta(j, r) + \tau_{ji}\}$, where $\delta(j, r)$ is the *earliest departure time* from customer/depot j. This value is recursively defined as $\delta(0, r) = 0$ and $\delta(i, r) = \alpha(i, r) + s_i$.

[1] In our formulation this constraint is enforced by construction since we define the total demand of a route as the sum of the single orders, therefore we implicitly do not allow partial deliveries.

Notice that for each order i on route r, this expression enforces by construction the fulfillment of the constraint $\alpha(i,r) \geq e_i$, which prevents early arrivals. Conversely, there is still the possibility of late arrivals, i.e., situations in which $\alpha(i,r) > l_i$. In practice these situations are usually allowed but they are treated as soft constraints and are penalized as described in Section 2.3.

Secondly, we consider the case of *heterogeneous* fleet (see, e.g., Gendreau et al, 1999, Semet and Taillard, 1993) and the possibility to outsource part of the transportation to external carriers (Volgenant and Jonker, 1987).

Heterogeneous vehicles: Vehicles are not identical as in the original problem but each vehicle j has its own capacity Q_j.

Carriers: Each vehicle j belongs to a *carrier*, denoted by $carr_j$, which is an external subcontractor of the transportation company.
Each carrier, including the company itself, uses a different function $t_j(r)$ to bill the routing costs to the transportation company, depending on the capacity of the vehicle employed and the length of the route (see Section 2.2 for details).

Moreover, in our problem, the planning period is not limited to a single day, but it spans over multiple days and each customer can place more than one order to be delivered in different days. In order to consider these features, we introduce the following entities:

Planning period: The planning period is composed by a number of consecutive days $\mathcal{D} = \{1, \ldots, d\}$. Therefore, we have to design a set of routing plans $\mathcal{R}^* = \{\mathcal{R}_1, \ldots, \mathcal{R}_d\}$, one for each day of the planning period.
Each vehicle performs only one route per day (it must return to the depot at the end of each day) and the same fleet is available on all the days of the planning period. We denote by r_{jk} the route travelled by vehicle j on day k.

Multiple orders: Each customer can issue different orders in different days. Therefore, at this stage orders and customers become different but related entities. For each order i we now define the (unique) customer associated to it, which will be denoted by $cust(i)$.

Delivery dates: As a consequence of introducing a multi-day perspective, there is also the possibility of specifying an interval of days $[\eta_i, \theta_i]$ in which the order i should be delivered.
Similarly to time windows case, delivery dates are treated as a soft constraint and their violations are penalized as described in Section 2.3.

Additional features of the problem we consider, concern the limitations of using some types of vehicles in particular situations and the possibility to leave orders out of the schedule.

As for the vehicle limitations, sometimes, due to site topology and road barriers, there might be impossible to use some vehicles to serve certain customers. Other limitations could regard the area of operation of some carriers, in the sense that they do not accept to deliver in specific regions (e.g., too far from their headquarters). Both alternatives are modeled by the following constraint:

Site reachability: It is given a compatibility matrix ρ, such that order i can be served by vehicle j only if $\rho_{ij} = 1$.

Since some of the real-world instances could be over-constrained in terms of the number of orders to be delivered, we give the possibility of define a priority on orders. Therefore we distinguish between *mandatory* orders, which must be served in a solution, and *optional* orders, which can be excluded. These concepts are captured in the following:

Mandatory/Optional orders: The set of orders $\mathcal{O}$ is partitioned into two sets $\mathcal{M}$ and $\mathcal{P}$ (where $\mathcal{O} = \mathcal{M} \cup \mathcal{P}$, $\mathcal{M} \cap \mathcal{P} = \emptyset$). Orders in $\mathcal{M}$ are mandatory, and must be delivered; orders in $\mathcal{P}$ are optional, therefore they can be discarded at a given cost γ_i.

The original constraints on VRP solutions must be adapted to deal with the new elements added in these stages. The constraints are modified as follows:

1. the routes satisfy all mandatory orders: $\mathcal{M} \subseteq \bigcup_{j=1}^{m} \bigcup_{k=1}^{d} ord(r_{jk}) \subseteq \mathcal{O}$;
2. each order is delivered at most once: $ord(r_{jk}) \cap ord(r_{j'k'}) = \emptyset, 1 \leq j < j' \leq n, 1 \leq k < k' \leq d$;

Finally, since, by regulation, drivers must take breaks during their activity, a set of mandatory *rests* of drivers must be set:

Driving rests: After a long consecutive working period, drivers should take a rest of a given minimum duration. In our case, a rest of 45 minutes after 4 hours and 30 minutes of consecutive driving is imposed by law.
Consequently, the earliest arrival time for the delivery of order i on route $r = \langle 0, \ldots, j, i, \ldots, 0 \rangle$ must be modified in order to take account of the mandatory rests: $\alpha'(i, r) = \max\{e_i, \delta(j, r) + \tau_{ji} + \zeta(i, r)\}$, where $\zeta(i, r)$ can be either 0 or 45 minutes according to the working/rest times patterns of the predecessors of order i in route r.
It is important to observe that in our formulation also service times and waiting times are accounted as working times for computing rests.

2.2 Vehicle Cost Functions

Since we consider the possibility to rely on external carriers for deliveries, we have to deal with different ways to compute transportation costs, even within a single problem instance. As an example, some carrier companies could bill the transportation company for the service on the basis of the route, other carriers could consider the size of the delivered goods, etc. Therefore, in order to be general enough, we designed our solver so that an external code for computing these costs can be invoked.

In the cases we have examined we have identified some common criteria for computing the transportation costs. In practice, the following four cost functions are used (where $dist_{ij}$ is the road distance between customers i and j):

1. A fixed cost for the vehicle c plus a cost ξ_1 per travel unit (measured in €/Km). If we denote with $\|r\|$ the total distance traveled in route r, i.e., $\|r\| = \sum_{i \in ord(r)} dist_{\pi(i,r)i}$, we have:
$$t(r) = c + \xi_1 \cdot \|r\|$$

2. A fixed cost for the vehicle c plus a cost ξ_2 per load unit (measured in €/Kg), which is dependent on the farthest location. If we denote the maximum distance between the depot and a costumer in the route with $\|r\|^*$, i.e., $\|r\|^* = \max_{i \in ord(r)}\{dist_{0i}\}$, we have:
$$t(r) = c + \xi_2\left(\|r\|^*\right) \cdot q(r)$$
3. A fixed cost for the vehicle c plus a cost ξ_1 per travel unit up to a predefined level of load L (dependent on the vehicle capacity), a cost per load unit ξ_2 dependent on the farthest location for larger loads. That is:
$$t(r) = \begin{cases} c + \xi_1 \cdot \|r\| & q(r) \leq L \\ c + \xi_2(\|r\|^*) \cdot q(r) & q(r) > L \end{cases}$$
4. A fixed cost for the vehicle c plus a cost ξ_3 per load unit, which is dependent both on the total load $q(r)$ and on the farthest location. That is:
$$t(r) = c + \xi_3\left(\|r\|^*, q(r)\right) \cdot q(r)$$

Since the value of the load cost coefficients ξ_2 and ξ_3 depends on the distance of the farthest customer $\|r\|^*$, the carrier should define such a value for each customer (and, in the case of ξ_3, also for each load level). Normally, the carriers partition their area of operation in regions and specify the load cost coefficient for every region (each customer location belongs to a region). The load cost coefficient selected to compute the cost is the largest of the route, i.e. the one associated with the region of the farthest customer.

2.3 Constraints and Objective Function

Similarly to other optimization problems, constraints are split into two categories: *hard* and *soft* constraints. A legal solution to the problem must satisfy all the hard constraints, whereas soft constraints can be violated and they are included in the objective function to be minimized.

Summarizing, in our formulation we deal with the following hard constraints:

- H1 The load of each vehicle must not exceed its capacity, i.e., $q(r_{jk}) \leq Q_j$, for $1 \leq j \leq m, 1 \leq k \leq d$.
- H2 Vehicles must return to the depot before a *shutdown time* $\bar{l}_0$ (in our case, $\bar{l}_0$ is fixed to 1 hour after the end of the depot time window l_0). Notice that late returns within l_0 and $\bar{l}_0$ are possible but they will be penalized as explained later, whereas solutions with a return time to the depot after $\bar{l}_0$ are infeasible.
- H3 The compatibility relation must be satisfied, i.e., an order i should be served by vehicle j for which the ρ_{ij} relation holds.
- H4 All mandatory orders must be delivered, i.e., $\mathcal{M} \subseteq \bigcup_{j=1}^{m} \bigcup_{k=1}^{d} ord(r_{jk})$.
- H5 The route timetable must obey the regulations on driving rests.

The other problem features described in Section 2.1 are considered as soft constraints and they become part of the objective function $F(\mathcal{R}^*) = w_{\mathsf{S1}} \cdot F_{\mathsf{S1}}(\mathcal{R}^*) + w_{\mathsf{S2}} \cdot F_{\mathsf{S2}}(\mathcal{R}^*) + w_{\mathsf{S3}} \cdot F_{\mathsf{S3}}(\mathcal{R}^*) + w_{\mathsf{S4}} \cdot F_{\mathsf{S4}}(\mathcal{R}^*)$, which is the linear combination of the following components:

S1 The delivery of an order on a day not included in its delivery days is penalized proportionally to its demand:

$$F_{\mathsf{S1}}(\mathcal{R}^*) = \sum_{k=1}^{d}\sum_{j=1}^{m}\sum_{i\in ord(r_{jk})} \left(1 - \chi_{[\eta_i,\theta_i]}(k)\right)\cdot q_i$$

where $\chi_I(x)$ is the characteristic function of interval I, i.e.,

$$\chi_I(x) = \begin{cases} 1 & x \in I \\ 0 & \text{otherwise} \end{cases}$$

S2 The delivery of an order after the end of its time window is penalized proportionally to the delay:

$$F_{\mathsf{S2}}(\mathcal{R}^*) = \sum_{k=1}^{d}\sum_{j=1}^{m}\sum_{i\in ord(r_{jk})} \max\{0, \alpha'(i, r_{jk}) - l_i\}$$

S3 Optional orders not delivered are penalized according to their cost γ_i:

$$F_{\mathsf{S3}}(\mathcal{R}^*) = \sum_{i\in\mathcal{O}\setminus\bigcup_{k=1}^{d}\bigcup_{j=1}^{m} ord(r_{jk})} \gamma_i$$

S4 The transportation costs for each vehicle is computed according to the carrier agreements t_j, in one of the forms described in Section 2.2:

$$F_{\mathsf{S4}}(\mathcal{R}^*) = \sum_{k=1}^{d}\sum_{j=1}^{m} t_j(r_{jk})$$

The weights of the various components are not fixed at some global level, but they are set by the operator for each specific case. To this regard, setting such weights is rather a complex task because the relative importance of the numerous components is difficult to establish. With the purpose of simplifying this process and having an immediate grasp of the costs, we decide to represent the costs directly in a real currency (€ in our case). Moreover, in order to deal with an objective function that can be represented in integer arithmetic and it is fine-grained enough, we set the cost unit to 1/1'000th of €.

3 Related Work

The *Vehicle Routing Problem* was first introduced by Dantzig and Ramser (1959) in what they called *The Truck Dispatching Problem*. It was formulated as a branch of the *Traveling Salesman Problem* (TSP) with multiple vehicles and routes. Subsequently, many other extensions that include time windows, different depots, pick-up and delivery options, heterogeneous fleet and periodic routing have been proposed.

We review here the articles in the literature that deal with those variants of the VRP that are the closest to our problem. For a complete review on the VRP and on VRPTW see (Laporte, 2009, Marinakis and Migdalas, 2007, Toth and Vigo, 2001) and (Bräysy and Gendreau, 2005a,b), respectively.

Baldacci et al (2007) have given an overview of approaches to solve the case of a fleet of vehicles characterized by different capacities, the HVRP (H for *Heterogeneous*). Some recent construction heuristics for the HVRP include the ones based on column generation methods (Choi and Tcha, 2007, Taillard, 1999) and on a sweep-based algorithm (Renaud and Boctor, 2002). The first metaheuristic approach for HVRP was proposed by Semet and Taillard (1993). Other Tabu Search approaches for this problem have been developed by Gendreau et al (1999), Osman and Salhi (1996), and Wassan and Osman (2002). Ochi et al (1998) have proposed a parallel genetic algorithm in conjunction of Scatter Search. Tarantilis et al (2003, 2004) have presented a list-based threshold accepting metaheuristic and finally Li et al (2007) have developed a deterministic variant of Simulated Annealing.

The HVRP with *Time Windows* (HVRPTW) has been much less studied than the HVRP. The first work is reported by Liu and Shen (1999), where they develop a saving-based construction heuristic. They have also created three sets of instances to test this new problem variant. More recently, Dell'Amico et al (2006) have proposed a solution approach based on a parallel insertion procedure and Bräysy et al (2008) have used a deterministic annealing metaheuristic.

The VRP with *Private fleet and Common carrier* (VRPPC) describes a situation where, by hypothesis, the total demand exceeds the capacity of the internal fleet, so an external transporter is necessary (*common carrier*). In this situation the problem is twofold: select customers that should be served by the external carrier and define routes of the internal fleet to serve remaining customers. The cost of serving an order by the common carrier is fixed for each order, without referring to any routing. Therefore, using the common carrier for an order is analogous to order exclusion in our formulation. The single vehicle case was formulated by Volgenant and Jonker (1987) and consequently solved exactly (for $n < 200$) by Diaby and Ramesh (1995). The VRPPC was formally introduced by Chu (2005) who solved it heuristically: he firstly applied a modified version of the classical saving procedure (Clarke and Wright, 1964) followed by some local exchanges between routes. Bolduc et al (2007) have proposed a heuristic called *SRI* that is composed of three steps: the selection of customer served by the external carrier, the construction of the solution (routing) and the improvement, through the application of sophisticated exchanges. Subsequently, Bolduc et al (2008) have presented a perturbation metaheuristic, called *RIP* (Randomized construction, Improvement, Perturbation), which essentially combines a descent method with diversification strategies. Recently, Côté and Potvin (2009) have obtained the best known results on a set of benchmarks using a Tabu Search approach.

Our problem is related also to the so-called *Team Orienteering Problem* (TOP). In the TOP a set of potential customers is available and a profit is collected from the visit to each customer. A fleet of vehicles is available to visit the customers, within a given time limit. The objective is to identify the customers which maximize the total collected profit while satisfying the given time limits for each vehicle. The TOP appeared first under the name *Multiple Tour Maximum Collection Problem* proposed by Butt and Cavalier (1994), but the definition of TOP was introduced by Chao et al (1996) that proposed a heuristic algorithm. Tabu Search approaches for this variant of the problem has been recently presented by Tang and Miller-Hooks (2005) and Archetti et al (2007).

4 Application of Tabu Search

First of all, it is important to observe that the presence of non-linear constraints (H5) and cost function components (the family of F_{S4} vehicle costs) makes it quite impractical to apply exact methods on this problem formulation. Therefore we resort to metaheuristic techniques for tackling the problem.

The solver we developed is based on the Tabu Search (Glover and Laguna, 1997) metaheuristic. At each step of the search process a subset of the neighborhood is explored and the neighbor that gives the minimum cost value becomes the new solution independently of the fact that its cost value is better or worse than the current one. The subset is induced by the *tabu list*, i.e., a list of the moves recently performed, whose *inverses* are currently forbidden and thus excluded from the exploration. In many cases (including our), the inverse is not a single move, but rather a set of moves determined by the values of a collection of attributes that are considered tabu.

The search stops after ii_{max} iterations without an improvement. Thus, the parameters of the technique to set are the length of the tabu list (tl), the *inverse* rule and the maximum number of idle iterations (ii_{max}).

In order to apply Tabu Search to our VRP problem we have to define several features. We first illustrate the search space and the procedure for computing the initial state. Then, we define the neighborhood structure and the prohibition rules, and finally we describe the set of search components employed and the high-level strategies for combining them.

4.1 Search Space, Cost Function, and Initial Solution

The local search paradigm is based on the exploration of a search space composed of all the possible complete assignments of values to the decision variables, possibly including also the infeasible ones.

In our case, a state is composed by a set of routes, one for each vehicle on each of the planning days.

An order can appear in only one route (i.e., it is *scheduled*) or it can be left outside, in the set of *unscheduled orders*. Thus, for each scheduled order, the solution specifies the day when the order is delivered, the vehicle, and the position in the corresponding route (the arrival time at the client is deterministically computed, given its position in the route, according to the rules presented in Section 2.1).

An example of a state is shown in Fig. 1, in which different gray levels are used to highlight routes performed on each of the two days composing the planning horizon (r_{jk} identifies the route j travelled on day k). Notice that some customers are left out of all routes whereas others are visited more than once because they place orders on different days.

The search space is restricted to states that satisfy constraints H3 (site reachability) and H4 (mandatory orders), whereas constraints H1 (vehicle capacity) and H2 (late return) can be violated and are included in the cost function with a high weight (H5 is satisfied by construction).

The cost function is thus the (monetary) sum of all soft constraints S1–S4, plus the *distance to feasibility* for H1 and H2 multiplied by a suitable high weight. For H1 the distance to feasibility is the sum of the quantities (measured in Kg) that exceed the vehicle capacity. For H2, there are several ways to define the distance to feasibility.

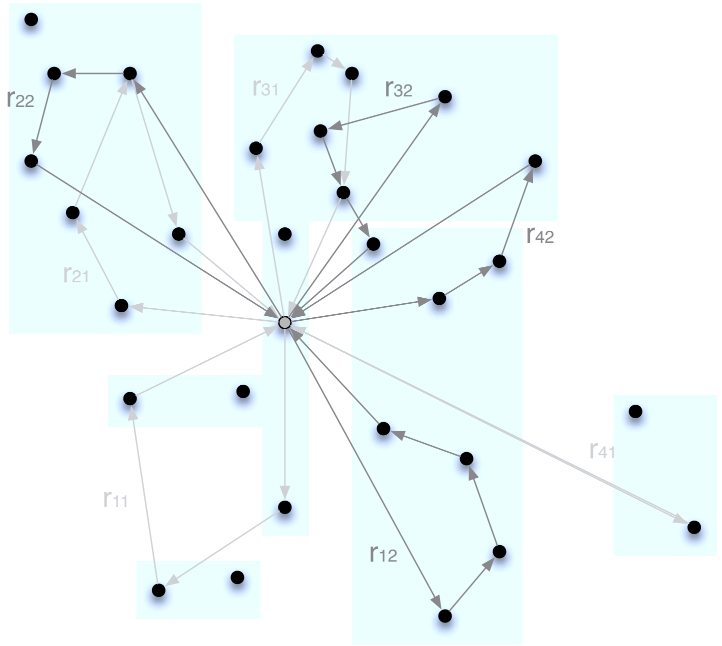

Fig. 1 An example of a state composed of 4 routes for 2 days (r_{jk} identifies the route j on day k).

Our choice is to count the number of orders (including the return to the depot) that are in a route that finishes after the shutdown time. This solution is more effective than summing up the delays w.r.t. the shutdown time, because it creates smoother trajectories from solutions with many violation toward the total elimination of them. For example, in the case of two orders of the same client that are late, if we only count the delays, in order to obtain an improvement we would need to move both orders at the same time (which is not done by our neighborhoods). Conversely, in our solution, every single order removed from the route improves the cost function independently of the fact that the total delay is reduced.

The initial solution is constructed at random, but satisfying some of the constraints. That is, we create a state of the search space that satisfies the constraints about the site reachability (H3), the driving rests (H5) and the delivery day (S1). This is made by assigning each order $i \in \mathcal{O}$ to a randomly selected feasible day $k \in [\eta_i, \theta_i]$ of the planning horizon $\mathcal{D}$ and to a random vehicle j, chosen among the compatible ones ($\rho_{ij} = 1$).

Once the day and the vehicle are selected, the route r is univocally identified, so we can assign the order to a random position in the selected route. The fulfillment of constraint H5 is enforced by construction. In addition, in the initial solution, all orders are scheduled, so that constraints H4 and S3 are also satisfied completely.

4.2 Neighborhood Relations

The neighborhood of a solution is usually implicitly defined by referring to a set of possible moves, which define transitions between solutions. A move is composed by *attributes* that identify the resources involved in the move. In our problem, we are dealing with the assignment of an order to three kinds of resources: the day, the vehicle and the position in the route.

We consider the following three neighborhood relations:

Insertion (Ins): This neighborhood is defined by the removal of an order from a route and its insertion in another one in a specific position. An order can also be inserted in the list of the unscheduled ones (the position is not meaningful in this case) or put back from this list to a route. The list of unscheduled orders is in practice treated as an additional special route, with the main difference that the position of orders in this sequence is irrelevant.

A move m of type Ins is identified by five attributes $m = \langle o, or, op, nr, np \rangle$ where o represents an order, or and op the old route and the old position in the old route, and np and nr, the new route and new position in the new route, respectively.

Inter-route swap (InterSw): This neighborhood is defined by exchanging an order with another one belonging to a different route. A move m of this type is identified by six attributes $m = \langle o1, o2, r1, r2, p1, p2 \rangle$, where $o1$ and $o2$ are orders, $r1$ and $r2$ are the routes of $o1$ and $o2$, and $p1$ and $p2$ are the positions of the orders in the routes.

Intra-route swap (IntraSw): This neighborhood is defined by exchanging an order with another one in the same route. A move m of this type is identified by five attributes $m = \langle o1, o2, r, p1, p2 \rangle$, where $o1$ and $o2$ are orders, r is the route, $p1$ and $p2$ are the positions of the orders in the route.

Notice that, given the state, some of the attributes are dependent from each other. For example, given an Ins move $m = \langle o, or, op, nr, np \rangle$, the order o identifies the pair (or, op) and vice versa. It is however useful to have all of them in the representation of the move for the definition of the prohibition rules.

4.3 Prohibition Rules

In the seminal version of Tabu Search, for the purpose to prevent cycling in the search trajectory, when a move m is in the tabu list, the move m' that would lead back to the same state (i.e., the *inverse* of m) is prohibited.

Nevertheless, in many cases there is a further risk that the search remains trapped in the proximity of some local minimum and iterates chaotically around it. In these cases, it is necessary to have some diversification mechanisms for "pushing" the search away from the minimum. This is obtained by generalizing the prohibition behavior with the definition of a general relation (called *prohibition rule*) between pairs of moves (m_t, m_e) that states that move m_t is excluded from the neighborhood by the fact that move m_e is in the tabu list. This enables the possibility that the presence of a move m_t in the tabu list results in the prohibition of a large set of moves, rather than the single inverse one. The prohibition rules are based on the values of the attributes of the two moves, the one in the tabu list m_t and the one under evaluation m_e.

It is quite difficult to tell *a priori* which is the most suitable prohibition rule for a given neighborhood, therefore for each of them we have defined and tested several ones, of different restrictive levels. They are compared experimentally in Section 5.3.

Rule	Condition	Description of tabu moves	Strength
PR1	$o_e = o_t \wedge or_e = nr_t$	moves removing the order o_t from the route nr_t	0.088%
PR2	$nr_e = or \wedge or_e = nr_t$	moves putting back any order from nr_t to or_t	0.485%
PR3	$o_e = o_t \wedge nr_e = or_t$	moves reinserting the order o_t in the route or_t	1.470%
PR4	$o_e = o_t$	moves involving the same order o_t	1.471%
PR5	$or_e = nr_t$	moves removing any order from the route nr_t	5.583%
PR6	$nr_e = or_t$	moves reinserting any order into the route or_t	7.978%

Table 1 Prohibition rules for TS(Ins).

For the Ins neighborhood, assuming that the move $m_t = \langle to, tor, op_t, nr_t, np_t \rangle$ is in the tabu list and the move $m_e = \langle o_e, or_e, op_e, nr_e, np_e \rangle$ is the move to be evaluated, we consider the six alternatives shown in Table 1. The last column shows the so-called *tabu strength*, which is defined as the (average) percentage of the entire neighborhood that a single move in the list prohibits.

Similarly, we have tested several prohibition rules also for the two neighborhoods IntraSw and InterSw. However, prohibition rules for these two neighborhoods have a more limited influence, and thus we report only the ones which proved to be the most effective for our instances. Specifically, for IntraSw, if $m_e = \langle o1_e, o2_e, r_e, p1_e, p2_e \rangle$ is the move to be tested and $m_t = \langle o1_t, o2_t, r_t, p1_t, p2_t \rangle$ is in the tabu list, the condition

$$o1_e = o1_t \vee o2_e = o2_t \vee o1_e = o2_t \vee o2_e = o1_t$$

is imposed. It forbids to make a move where any of the two orders of m_e is equal to any of those of m_t. Its tabu strength is 7.613%. The same condition is used also for InterSw. If $m_e = \langle o1_e, o2_e, r1_e, r2_e, p1_e, p2_e \rangle$ is the move to be tested and $m_t = \langle o1_t, o2_t, r_t, r_t, p1_t, p2_t \rangle$ is the move in the tabu list, then

$$o1_e = o1_t \vee o2_e = o2_t \vee o1_e = o2_t \vee o2_e = o1_t$$

is imposed. For InterSw the tabu strength of this prohibition rule is 5.933%.

4.4 Tabu Dynamics

We make use of the Robust Tabu Search scheme, as it is called by Taillard (1991), in which the length of the tabu list is *dynamic*. This is obtained by assigning at random to each performed move the number of iterations in which it will remain in the tabu list. In detail, we set two values tl and δ_{tl} and we assign to each accepted move a tabu tenure randomly selected between $tl - \delta_{tl}$ and $tl + \delta_{tl}$. In all our experiments δ_{tl} is set to 2 (based on preliminary experiments), whereas tl is subject to tuning.

The tabu status of a move can be overruled by the so-called *aspiration criterion* which makes a move acceptable even if it is tabu. In this work, we use a basic aspiration criterion which states that a move is accepted if it improves on the current best solution. This criterion is called NB by Hvattum (2009), who claims that it is a “safe choice” that works well in general cases.

When all moves are tabu and no aspiration applies, the Tabu Search algorithm executes the best of all the moves, thus ignoring the tabu status.

4.5 Search Techniques

On the basis of the three neighborhood relations defined, we come up with three basic Tabu Search techniques, that we call TS(Ins), TS(IntraSw), TS(InterSw).

It is important to observe that the search space is not connected under the IntraSw and InterSw neighborhood structures. Indeed, both neighborhoods do not change the number of orders in a route, thus there is no trajectory that goes from a state with a given number of orders in a route to a state with a different one. Consequently, TS(IntraSw) and TS(InterSw) must be used in combination with TS(Ins), since they are not effective when used by themselves.

We use a sequential solving strategy for combining Tabu Search algorithms based on different neighborhood functions, as proposed (among others) by Di Gaspero and Schaerf (2006) under the name of *token-ring* search. Token ring works as follows: Given an initial state and a set of algorithms, it makes circularly a run of each algorithm, always starting from the best solution found by the previous one. The overall process stops either when a full round of the algorithms does not find an improvement or the time granted is elapsed. Each single technique stops when it does not improve the current best solution for a given number of iterations (*stagnation*).

We identify the following five strategies, applying the token ring to the basic neighborhood structures. In the following, token ring is denoted by the symbol $\triangleright$.

1. TS(Ins)
2. TS(Ins) $\triangleright$TS(IntraSw)
3. TS(Ins) $\triangleright$TS(InterSw)
4. TS(Ins) $\triangleright$TS(IntraSw) $\triangleright$TS(InterSw)
5. TS(Ins) $\triangleright$TS(InterSw) $\triangleright$TS(IntraSw).

We also consider solvers that use the union of many neighborhoods: The algorithm based on this compound neighborhood, denoted by the $\oplus$ symbol in (Di Gaspero and Schaerf, 2006), selects at each iteration a move belonging to any of the neighborhoods as part of the union. We therefore have three more strategies:

6. TS(Ins $\oplus$ IntraSw)
7. TS(Ins $\oplus$ InterSw)
8. TS(Ins $\oplus$ IntraSw $\oplus$ InterSw)

Finally, we experiment two further techniques that use both union neighborhoods and token ring search:

9. TS(Ins) $\triangleright$TS(IntraSw $\oplus$ InterSw)
10. TS(Ins) $\triangleright$TS(Ins $\oplus$ IntraSw $\oplus$ InterSw)

All these ten strategies will be analyzed and compared experimentally in the next section.

5 Experimental Analysis

In this section, we first introduce the benchmark instances and the general settings of our analysis, and then we move to the experimental results. We analyze our techniques in stages, starting from the simplest algorithm which uses one single neighborhood, then moving to the more complex ones. Finally we show the results of the best configuration of our solver on benchmarks of the VRPPC.

Inst.	#Orders	#Customers	#Vehicles	#Carries	#Regions
case1	139	44	7	4	19
case2	166	56	7	4	22

Table 2 Features of the two scenarios.

5.1 Benchmark Instances

Two cases coming from different real-world scenarios provided by our industrial partner, BeanTech s.r.l., are at our disposal for the experimental part. The main features of these two cases are shown in Table 2. In order to highlight the importance of the various components of the problem, starting from each case we have created 9 different instances, named with the letters from A to I. The A instances are the original ones, whereas instances B–I are obtained by perturbing one specific feature, so as to produce instances that are realistic but specifically biased toward stressing the use of a particular feature.

Table 3 shows the resulting 18 instances, along with the values of a set of indicators that describe the features (perturbed values are in bold). In detail, the columns are defined as follows:

– **Days (D):** The number of days in the planning horizon.
– **Filling Ratio (FR):** The ratio between the total demand and the total capacity of the vehicles multiplied by the number of days.
– **Time Windows (TW):** The average ratio between the time window of the orders and the time window of the depot.
– **Day Window (DW):** The average number of available days of the orders divided by the total number of days.
– **Compatibility (C):** The density of the compatibility matrix between vehicles and orders.
– **Mandatory Orders (MO):** The percentage of mandatory orders.
– **Space Occupancy (SO):** The average percentage of space taken by an order in a vehicle.
– **Outsourcing (OUT):** If this indicator is set to F (False), it means that there are no external carriers and the routing costs depend only on the total distance travelled; if the indicator is set to T (True), there are external carriers and consequently different ways to compute costs.

Instances H are modified in such a way that each orders should be dispatched on the first day of its delivery dates (*as soon as possible*); alternatively for instances I each order should be dispatched on the last day of its delivery dates (*as last as possible*). For all instances the cost γ_i of not delivering an optional order i ($i \in \mathcal{P}$) is set equal to its demand q_i.

Summarising, we tested our algorithms on a benchmark composed of 18 instances, which have been made available through the web at the URL `http://www.diegm.uniud.it/ceschia/index.php?page=vrptwcdc`.

5.2 General Settings and Implementation

All the algorithms have been implemented in the C++ language, exploiting the EasyLocal++ framework (Di Gaspero and Schaerf, 2003). The experiments have been

Inst.	D	FR	TW	DW	C	MO	SO	OUT
case1-A	3	27.47	90.91	89.47	87.97	14.04	10.12	T
case1-B	3	27.47	90.91	89.10	87.97	**100.00**	11.09	T
case1-C	**2**	41.21	90.91	76.27	87.89	13.56	9.77	T
case1-D	**4**	20.60	90.91	64.74	88.83	10.26	7.39	T
case1-E	3	27.47	**73.12**	89.47	87.97	14.04	10.12	T
case1-F	3	27.47	**40.51**	89.47	87.97	14.04	10.12	T
case1-G	3	27.47	90.91	89.47	100	14.04	10.12	**F**
case1-H	3	27.47	90.91	**33.33**	87.97	14.04	10.12	T
case1-I	3	27.47	90.91	**33.33**	88.45	14.89	12.27	T
case2-A	3	47.41	90.91	66.30	85.87	16.67	11.06	T
case2-B	3	47.41	90.91	65.84	85.71	**100.00**	12.29	T
case2-C	**2**	71.12	90.91	80.25	85.36	17.28	12.29	T
case2-D	**4**	35.56	90.91	53.47	86.38	14.81	9.21	T
case2-E	3	47.41	**81.41**	66.30	85.87	16.67	11.06	T
case2-F	3	47.41	**41.62**	66.30	85.87	16.67	11.06	T
case2-G	3	47.41	90.91	66.30	100	16.85	11.19	**F**
case2-H	3	47.41	90.91	**33.33**	84.87	19.12	14.64	T
case2-I	3	47.41	90.91	**33.33**	85.88	17.05	11.31	T

Table 3 Features of instances.

performed on an Intel QuadCore PC (64 bit) running Ubuntu Linux 10.04, the software has been compiled using the GNU C++ compiler (v. 4.4.3).

The stopping criterion of the basic algorithms is the detection of stagnation, which can occur at different times. Therefore, in order to compare different combinations in a fair way, we decide to set a maximum number of iterations equal to 1000 for each basic TS(·) component of the token ring strategy. Moreover, we also impose a maximum of 3 full rounds or 1 idle round as the stopping criterion of the whole token ring procedure. We run 100 trials for each configuration of the parameters.

For each individual TS(·) component of the search strategy, the maximum number of iterations from the last improvement is set to 500 for the Ins neighborhood and all the unions including it, and 300 for the other neighborhoods.

For the experiments we set $w_{\mathsf{S1}} = 30$, $w_{\mathsf{S2}} = 10$, $w_{\mathsf{S3}} = 250$ and $w_{\mathsf{S4}} = 1$.

5.3 Results on Prohibition Rules for TS(Ins)

The first set of experiments focuses on the Insertion neighborhood and compares different prohibition rules for TS(Ins).

Obviously, for each prohibition rule the best tabu list length can be different. Therefore we have to tune the length for each rule independently and then compare the prohibition rules among themselves, each with its best setting.

Fig. 2 shows, in form of box-and-whiskers plots, the results obtained on the two original instances for different values of the tabu length and for three prohibition rules, namely PR1, PR4, and PR5. We select to show PR5 because it provides the best results, and PR1 and PR4 because they correspond to diverse tabu strength levels.

The figures highlight that the best prohibition rule is PR5, and this is confirmed by performing a statistical comparison of the different results, yielding to p-values of the pairwise two-sided *Student* t-test (Venables and Ripley, 2002) that are inferior to 0.0001.

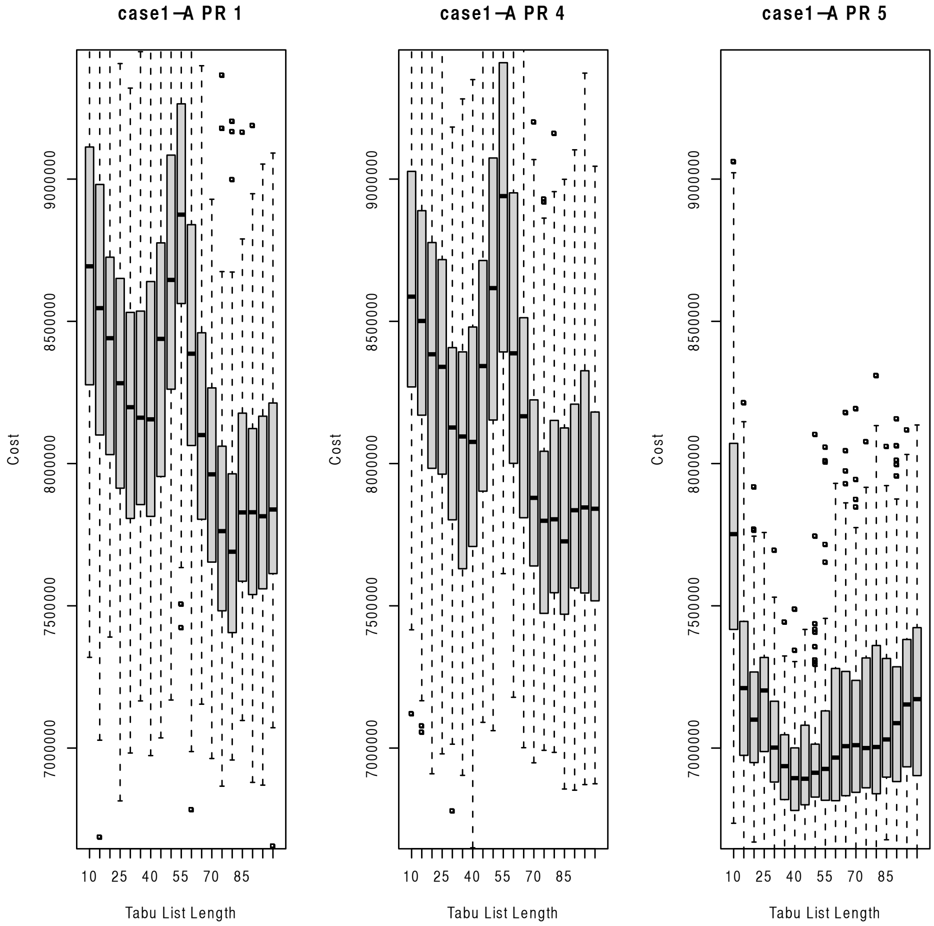

Fig. 2 Results for different tabu list lengths of TS(Ins) for case1-A with different prohibition rules.

They also show an interesting phenomenon regarding the tabu list length for prohibition rules PR1 and PR4: The curves have two different minima. Our interpretation is that the first one is related to the depth of local minima in the search space and it represents the "normal" behavior of Tabu Search, the second "spurious" minimum is due to the situation in which most of the moves are tabu, and the search alternates between performing non tabu moves and tabu moves. Anyway, this situation provides an effective diversification and, in our case, this second minimum is indeed the lowest one. This phenomenon is confirmed by the fact that for PR1 and PR4 in such conditions about 30% of the moves performed are tabu. It is worth remarking that this anomalous behavior occurs only on the less performing prohibition rules.

The results obtained for the other instances and prohibition rules are similar to those shown in the figures and are omitted.

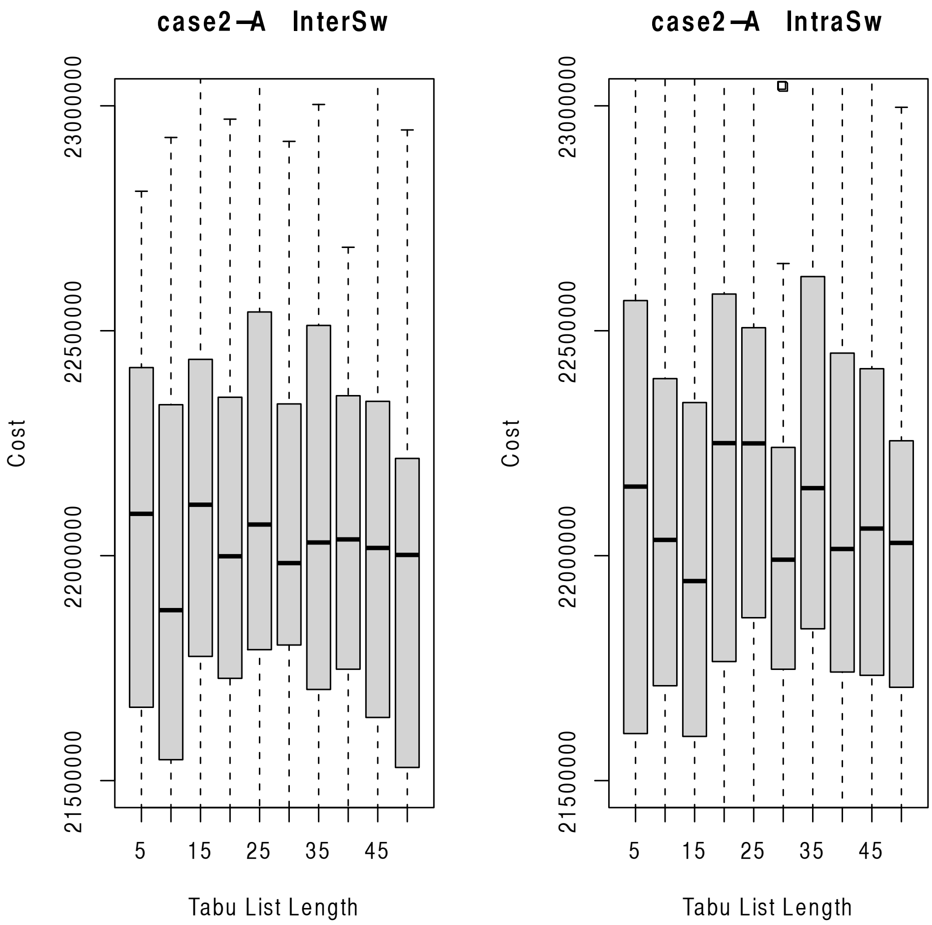

Fig. 3 Results for different tabu list lengths of TS(IntraSw) and TS(InterSw) for case2-A.

5.4 Results on Inter-route swap and Intra-route swap Neighborhoods

As already noticed, TS(InterSw) and TS(IntraSw) cannot be used alone because the search space is not connected under their neighborhood relations. Consequently, it would be meaningless to tune those algorithms starting from random solutions and instead we apply the tuning procedure to the strategies TS(Ins)▷TS(InterSw) and TS(Ins)▷TS(IntraSw).

In Fig. 3 we show the outcome of the experiments. In this case, for TS(InterSw) and TS(IntraSw) we can see that the results are not really affected by the length of the tabu list. We select the value 10 for TS(InterSw) and 15 for TS(IntraSw) in the following experiments, although there is no significant statistical difference with the other values. In fact their role is to diversify during the search process, moving on plateaux of the search space.

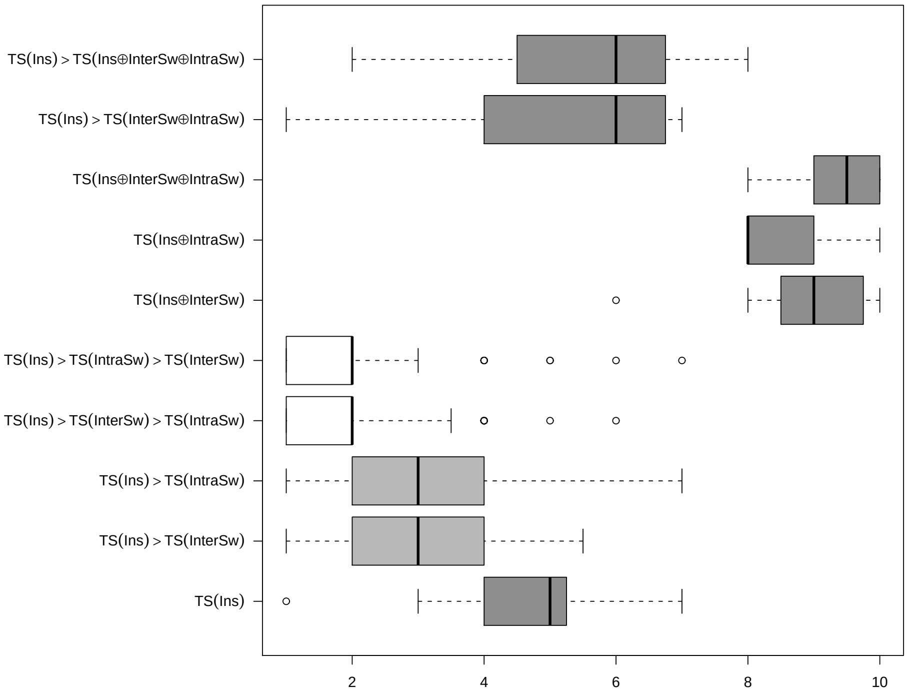

Fig. 4 Results of the F-Race selection procedure on the composite solvers.

5.5 Results on Composite Solvers

Our final experiment on the problem concerns the comparison of composite solvers. For each component of the solvers, the parameters are set to the best values found in the previous experiments.

We perform a F-Race selection (Birattari et al, 2002) based on Friedman two-way analysis of variance by ranks as implemented in the EasyAnalyzer framework (Di Gaspero et al, 2007). At each step of the selection procedure each solver is tested on one instance (among the available ones) and it is assigned a rank according to the value of the cost function. This way we are able to compare the results on different instances independently on the inherent difference in the cost functions. As soon as statistical evidence that a given solver is inferior with respect to the others is collected, it is discarded and the comparison proceeds only among the remaining ones.

We decide to set $p < 0.05$ as for the confidence level employed in the F-Race procedure and we allow for at most 100 replicates.

In this case, in order to compare different solvers in a fair way, we decide to add a timeout mechanism that stops the overall solver when the total time granted is elapsed. Each solver is allowed to run for 500 seconds.

The results are presented in Fig. 4, in from of box-and-whiskers plots, showing the distribution of the ranks obtained by each configuration. Moreover, boxes are filled with a gray level which is proportional to the stage of the F-Race procedure in which the corresponding algorithm has been discarded (the darker, the sooner). This way, the

Inst.	S1 Date Window	S2 Time Window	S3 Order Outside	S4 Vehicle Fixed Cost	S4 Vehicle Travel Cost	H2 Late Return	Total Cost
case1-A	46.95	101.20	375.25	2700	3205.250	0	6428.650
case1-B	28.38	172.63	0.00	3200	3633.178	1	7034.188
case1-C	291.72	89.33	187.50	3200	3431.114	0	7199.664
case1-D	161.64	110.02	147.50	2600	3703.168	0	6722.328
case1-E	8.43	74.18	40.00	2600	3264.802	0	5987.412
case1-F	24.45	145.21	1039.50	2600	3099.078	0	6908.238
case1-G	14.76	99.93	1149.75	1500	2810.880	0	5575.320
case1-H	714.00	155.81	477.25	2800	3370.175	0	7517.235
case1-I	799.92	74.07	673.25	2500	3874.780	0	7922.020
case2-A	637.68	150.74	11568.75	3200	5197.776	0	20754.946
case2-B	791.82	371.61	0.00	4500	7801.087	8	13464.517
case2-C	316.47	120.85	12631.00	3200	5022.794	0	21291.114
case2-D	583.17	127.72	11501.25	3100	5368.957	0	20681.097
case2-E	462.84	83.88	11847.00	2700	5185.437	0	20279.157
case2-F	437.94	271.39	12263.25	3200	5042.965	0	21215.545
case2-G	237.87	151.19	12860.00	2100	3580.944	0	18930.004
case2-H	1792.44	150.15	11538.50	3000	5390.057	0	21871.147
case2-I	1668.75	213.28	11671.50	3000	5272.349	0	21825.879

Table 4 Values (in €) of the different cost components for the best solutions.

algorithms that were found as equally good at the end of the procedure are denoted by white boxes.

The final outcomes of the selection procedure report that only the two solvers TS(Ins) ▷TS(IntraSw) ▷TS(InterSw) and TS(Ins) ▷TS(InterSw) ▷TS(IntraSw) survive the F-Race, revealing that all three moves are necessary for obtaining high quality solutions.

Interestingly enough, the solver that uses the union of the three moves does not achieve good results and it has been discarded early by the F-Race procedure. In our opinion, the explanation of this fact is twofold: On the one hand, the use of IntraSw and InterSw moves in the initial stage of the search leads to bad quality local minima, because it tends to optimize single routes before spreading the orders correctly in the various routes. On the other hand, in the later stage of the search, the presence of a large set of IntraSw and InterSw moves having small improvements prevents the search from making the "disruptive" Ins moves necessary to find deeper local minima. Conversely, the token-ring solvers, by focusing on each single move type, allow the search to perform a more effective diversification at different stages of the search.

Table 4 shows the costs of the different components of the objective function for the best solution for each instance. The table reports also the number of (hard) violations: the corresponding value represents the number of orders that are in some routes that return after the shutdown time (H2).

Unsurprisingly, the main cost comes up from the traveling of the vehicles and their fixed cost of use. Among the other components, the most relevant is the one related to unscheduled orders (S3). However, the other two that are related to delivery in the wrong day (S1) and delivery after the end of the time window (S2) are not negligible.

In addition, looking at the different results for instances A–I of the same case, it is evident that the tightness of a specific "resource" (time window, day, . . .) makes the cost of the corresponding component higher in the best solution.

Inst.	CPLEX	Chu (2005)		Bolduc et al (2007)		Bolduc et al (2008)		Us	
		z	sec	z	sec	z	sec	z	sec
Chu-H-01	387.5*	387.5	0.02	387.5	0.00	387.5	0.35	387.5	0.11
Chu-H-02	586.0*	631.0	0.03	586.0	0.02	586.0	1.90	586.0	0.70
Chu-H-03	823.5*	900.0	0.08	826.5	0.03	826.5	3.50	823.5	1.96
Chu-H-04	1389.0*	1681.5	0.06	1389.0	0.08	1389.0	5.85	1389.0	7.79
Chu-H-05	1441.5	1917.0	0.28	1444.5	0.09	1441.5	10.40	1441.5	16.93
B-H-01	423.5*	503.0	0.02	423.5	0.02	423.5	1.85	423.5	0.11
B-H-02	476.5*	476.5	0.05	476.5	0.02	476.5	3.65	476.5	0.76
B-H-03	777.0*	884.0	0.11	804.0	0.03	778.5	4.75	777.0	2.13
B-H-04	1521.0*	1737.0	0.06	1564.5	0.09	1521.0	15.85	1521.0	7.81
B-H-05	1609.5	1864.5	0.16	1609.5	0.13	1609.5	12.90	**1578.0**	16.30

Table 5 Results on benchmarks Chu-H and B-H of the *Vehicle Routing Problem with Private fleet and Common carrier.*

Indeed, we can see that if all orders are mandatory (B), the solver is not able to find a feasible solution, and the cost component related to vehicles (S4) increases, because the solver leads towards solutions that use all available resources. The reduction of the planning horizon (C) causes a larger number of deliveries on a wrong day (S1) and higher value of vehicle costs; on the other hand, its extension brings to solutions with less orders unscheduled.

Results highlight that the constraint related to Time Windows is really tight and its relaxation (E) or restriction (F) has a strong effect on all other cost components. In addition, as we could expect, the lowest value of the total cost comes out for the G, confirming the idea that using the internal fleet is cheaper (if available). Finally, results of the last two cases show that if we schedule all orders only on the first day (H) or on the last day (I) of the planning horizon, that impacts on every cost component, and we obtain solutions with the highest total cost.

We must remark that the effect of perturbations is more evident for case1 than for case2; this is probably due to a higher filling ratio (see Section 5.1) for the latter, that leaves less freedom during the search process.

5.6 Comparison with benchmarks of the VRPPC

In order to evaluate the performance of our solver also on public benchmarks, we adapt it to solve the *Vehicle Routing Problem with Private fleet and Common carrier.*

We test our solver on two instance families, namely Chu-H (Chu, 2005) and B-H (Bolduc et al, 2007). For these instances the number of customers ranges from 5 to 29, the internal fleet is heterogeneous and the external carrier cost was set equal to 6 times the distance between the depot and the corresponding customer. We run one of the two best solvers, namely TS(Ins) ▷TS(IntraSw) ▷TS(InterSw), for 400 trials, and for each instance we set the tabu list length equal to a fifth of the number of orders.

Table 5 compares the best solution values obtained by Chu (2005), Bolduc et al (2007), and Bolduc et al (2008) with ours. The first column reports the solution values computed by Bolduc et al after a maximum of 150 hours of computation time of CPLEX (v. 9.0). The values marked with * are proven optimal solutions.

The outcome is that our solver is able to obtain the optimum or the best solution value for all instances, and besides, for instance B-H-05 it has found a new best known result.

6 Conclusions and Future Work

We have modeled a highly complex version of the classical vehicle routing problem, arising from a real world situation. We have also proposed an approach based on Tabu Search for its solution. To this aim, we have investigated the use of three different neighborhoods and the experimental analysis shows that the best results are obtained by a combination of all of them. We also demonstrate that on public benchmarks of *Vehicle Routing Problem with Private fleet and Common carrier* our results are very competitive.

For the future, we plan to test the use of other neighborhood relations and other search techniques and to compare them in a principled way on a larger set of instances. We also plan to experiment our solution techniques on similar problems within the VRP family. Finally, we plan to investigate the relative importance of the various cost components and the relation of their weights with the tuning of the parameters of the solver.

Acknowledgements We are grateful to BeanTech s.r.l. for bringing this problem to our attention, providing to us the instances, and supporting Sara Ceschia with a grant for this work.

References

Archetti C, Hertz A, Speranza MG (2007) Metaheuristics for the team orienteering problem. Journal of Heuristics 3:49–76

Baldacci R, Battara M, Vigo D (2007) Routing a heterogeneous fleet of vehicles. Technical Report 2007/1, DEIS, University of Bologna

Birattari M, Stützle T, Paquete L, Varrentrapp K (2002) A racing algorithm for configuring metaheuristics. In: *et al* WBL (ed) GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, New York, pp 11–18

Bolduc MC, Renaud J, Boctor F (2007) A heuristic for the routing and carrier selection problem. European Journal of Operational Research 183(2):926–932, DOI 10.1016/j.ejor.2006.10.013

Bolduc MC, Renaud J, Boctor F, Laporte G (2008) A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. Journal of the Operational Research Society 59:776–787

Bräysy O, Gendreau M (2005a) Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Transportation Science 39(1):104–118, DOI 10.1287/trsc.1030.0056

Bräysy O, Gendreau M (2005b) Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. Transportation Science 39(1):119–139, DOI 10.1287/trsc.1030.0057

Bräysy O, Dullaert W, Hasle G, Mester D, Gendreau M (2008) An effective multi-restart deterministic annealing metaheuristic for the fleet size and mix vehicle routing problem with time windows. Transportation Science 42(3):371–386

Butt SE, Cavalier TM (1994) A heuristic for the multiple tour maximum collection problem. Computers and Operations Research 21(1):101–111

Chao IM, Golden BL, Wasil EA (1996) The team orienteering problem. European Journal of Operational Research 88(3):464–474

Choi E, Tcha DW (2007) A column generation approach to the heterogeneous fleet vehicle routing problem. Computers and Operations Research 34(7):2080–2095

Chu CW (2005) A heuristic algorithm for the truckload and less-than-truckload problem. European Journal of Operational Research 127(3):657–667

Clarke G, Wright JW (1964) Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research 12(4):568 – 581

Côté JF, Potvin JY (2009) A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. European Journal of Operational Research 198(2):464–469

Dantzig G, Ramser J (1959) The truck dispatching problem. Management Science 6(1):80–91

Dell'Amico M, Monaci M, Pagani C, Vigo D (2006) Heuristic approaches for the fllet size and mix vehicle routing problem with time windows. Tech. rep., DISMI, University of Modena and Reggio Emilia, Italy

Di Gaspero L, Schaerf A (2003) EasyLocal++: An object-oriented framework for flexible design of local search algorithms. Software—Practice and Experience 33(8):733–765

Di Gaspero L, Schaerf A (2006) Neighborhood portfolio approach for local search applied to timetabling problems. Journal of Mathematical Modeling and Algorithms 5(1):65–89

Di Gaspero L, Roli A, Schaerf A (2007) EasyAnalyzer: an object-oriented framework for the experimental analysis of stochastic local search algorithms. In: Stützle T, Birattari M, Hoos H (eds) Engineering Stochastic Local Search Algorithms (SLS-2007), Springer-Verlag, no. 4683 in Lecture Notes in Computer Science, pp 76–90

Diaby M, Ramesh R (1995) The distribution problem with carrier service: a dual based approach. ORSA Journal on Computing 7(1):24–35

Gendreau M, Laporte G, Musaraganyi C, Taillard ED (1999) A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers and Operations Research 26(12):1153–1173

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

Hoos HH, Stützle T (2005) Stochastic Local Search – Foundations and Applications. Morgan Kaufmann Publishers, San Francisco, CA (USA)

Hvattum LM (2009) On the value of aspiration criteria in tabu search. In: The VIII Metaheuristics International Conference (MIC 2009), Hamburg, Germany

Laporte G (2009) Fifty Years of Vehicle Routing. Transportation Science 43(4):408–416

Li F, Golden B, Wasil E (2007) A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Computers and Operations Research 34(9):2734–2742

Liu FH, Shen SY (1999) The fleet size and mix vehicle routing problem with time windows. Journal of the Operational Research Society 50(7):721–732

Marinakis Y, Migdalas A (2007) Annotated bibliography in vehicle routing. Operational Research 7(1):27–46, DOI 10.1007/BF02941184

Ochi LS, Vianna DS, Drummond LMA, Victor AO (1998) A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. In: Parallel and Distributed Processing, vol 1388, pp 216–224

Osman IH, Salhi S (1996) Local search strategies for the vehicle fleet mix problem. In: Rayward-Smith VJ, Osman IH, Reeves CR, Smith GD (eds) Modern Heuristic Search Methods, John Wiley & Sons, chap 8, pp 131–153

Renaud J, Boctor FF (2002) A sweep-based algorithm for the fleet size and mix vehicle routing problem. European Journal of Operational Research 140:618–628

Semet F, Taillard E (1993) Solving real-life vehicle routing problems efficiently using tabu search. Annals of Operations Research 41:469–488

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35(2):254–265

Taillard E (1999) A heuristic column generation method for the heterogeneous fleet vrp. RAIRO Recherche Opérationnelle 33(1):1–14

Taillard ED (1991) Robust taboo search for the quadratic assignment problem. Parallel Computing 17(4-5):443–455

Tang H, Miller-Hooks E (2005) A tabu search heuristic for the team orienteering problem. Computers and Operations Research 32(6):1379–1407

Tarantilis CD, Kiranoudis CT, Vassiliadis VS (2003) A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. Journal of the Operational Research Society 54(1):65–71

Tarantilis CD, Kiranoudis CT, Vassiliadis VS (2004) A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research 152(1):148–158

Toth P, Vigo D (2001) An overview of vehicle routing problems. In: The vehicle routing problem, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp 1–26

Toth P, Vigo D (eds) (2002) The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications, S.I.A.M., Philadelpia, PA (USA)

Venables WN, Ripley BD (2002) Modern applied statistics with S, 4th edn. Statistics and Computing, Springer

Volgenant T, Jonker R (1987) On some generalizations of the travelling-salesman problem. Journal of the Operational Research Society 38(11):1073–1079

Wassan NA, Osman IH (2002) Tabu search variants for the mix fleet vehicle routing problem. Journal of the Operational Research Society 53(7):768–782