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

Sara Ceschia · Andrea Schaerf

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 function for the vehicles, and the possibility of leaving orders unscheduled.

We propose a metaheuristic approach based on tabu search and on a complex combination of neighborhood relations. Finally, we perform an experimental analysis to tune and compare different combinations, highlighting the most important features of the algorithm.

1 Introduction

Vehicle routing is one of the most studied problems in optimization (see, e.g., [1]). Many variants of the vehicle routing problem (*VRP*) have been introduced in the literature in the recent years, ranging from multi-depot, to time windows, to mixed fleet, just to name some. Nevertheless, despite the availability of this large set of classified formulation, sometimes the practical problem that companies have to face is still more complex than the standardized version discussed in scientific articles.

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

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

We propose for this problem a solution technique based on tabu search (*TS*) [2,3], that makes use of a combination of different neighborhood relations. The experimental analysis is carried out on a few real-world instances, and makes use of principled statistical tests to compare different variants and to tune the parameters.

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

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

The paper is organized as follows. In Section 2 we present the problem formulation. 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. 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 (in Section 2.1). In Section 2.2, we describe in detail the costs associated with the use of the vehicles. Finally, in Section 2.3 we summarize the overall objective function of the problem.

2.1 Features of the Problem

We describe our formulation starting from the basic version of VRP, the *Capacitated* VRP. We have the following entities and constraints:

1. Customer: Each customer requires a supply of a certain amount of goods (*demand*).
2. Fleet: The transportation of goods is performed by using a fleet of vehicles. The vehicles are identical (i.e., they have the same capacity) and located at a single central depot (*home depot*), where they have to return upon complete delivery.
3. Load Limits: The load of each vehicle cannot exceed the vehicle capacity.

The solution of a VRP calls for the determination of a set of *routes*, each performed by a single vehicle that starts and ends at the depot, such that all the requirements of the customers are fulfilled and the global transportation cost is minimized. The travel cost can be either the distance or a different measure of the total expenses of going on a given street (time, tolls, ...).

The first extensions we consider are represented by the so-called *Time Windows* and *Service Times*.

4. Time Windows: Each customer is associated with a time interval. The service of each customer must start within that interval (window). In case of early arrives at the location of a customer, the vehicle is allowed to wait until the service can start.
5. Service Times: Each customer is associated with a service time needed to unload the goods from the vehicle. The vehicle must stop at the customer location for the service time.

The routing cost is equal to the sum of all the times spent along the route: the traveling time, the service time, and the possible waiting time.

The depot also has a time window, that usually includes all the windows of the customers. All vehicles leave at the start time of the depot window, and they must be back before the end of it.

Secondly, we introduce the possibility that the fleet is heterogeneous and limited:

6. Vehicles and Carriers: Each vehicle has its own capacity and belongs to a *carrier*. Each carrier uses a different function to compute the routing costs, as detailed in Section 2.2, depending on the capacity of the vehicle and the length of the route.

7. Limited fleet: The number of available vehicles of each capacity and each carrier is fixed.

In our application, the planning period is not limited to a single day of exercise, but it spans over several days. To this aim, we introduce the following quantities:

8. Planning period: The planning period is composed by a number of days. Thus we have to design a set of routes for each day of a given planning period. Each vehicle performs only one route per day (it must return at the depot at the end of the day). Every day of the period we have the same fleet available.
9. Delivery dates for orders: A customer can make more than one order during the planning period, each with a different demand. For each order, the customer specifies in which days, during the planning period, the goods can be delivered.

The next extension regards the mandatory rests of drivers. In fact, by regulation, drivers must take breaks during their driving duty:

10. Driving rests: Drivers should take breaks of at least 45 minutes after 4 hours and half of consecutive driving. Therefore, in the schedule, we have to include the rest time in every route that is longer than 270 minutes.

Due to site topology and road barriers, there might be limitations on the possibility of using some vehicles to serve certain costumers.

11. Site reachability: There is a compatibility relation between customers and vehicles and every customer has to be served only by compatible vehicles.

Finally, in our case some orders might be left out of the schedule.

12. Exclusion of orders: Orders are split into *mandatory* and *optional*. Optional orders can be excluded from the schedule and consequently they have an associated fixed cost because they are not scheduled.

2.2 Vehicle Cost Functions

Carrier companies have different ways to bill the customer, based on the route and the size of the delivered goods. To this regard, our solver for computing the cost invokes an external code, that in principle could contain any computable function.

However, in the cases we have examined this is restricted to the following four functions:

1. A fixed cost for the vehicle plus a cost per travel unit (Km).
2. A fixed cost for the vehicle plus a cost per load unit (Kg) dependent on the farthest location.
3. A fixed cost for the vehicle plus a cost per travel unit up to a predefined level of load (dependent on the vehicle capacity), a cost per load unit for larger loads.
4. A fixed cost for the vehicle plus a cost per load unit, with different values dependent on the total load and on the farthest location.

In fact, each customer is located in a region and the carrier specifies the load cost coefficient for every region. The coefficient used to compute the cost per load unit is the largest of the route, i.e. the one associated with the region of the farthest customer (with respect to the depot).

2.3 Objective Function

As usual, we split the constraints in hard and soft ones. The first ones must be necessarily satisfied by a solution, whereas the latter ones can be violated and they contribute to the objective function to be minimized.

In our problem, there are three types of hard constraints:

H1 The capacity of the vehicle cannot be exceeded.
H2 The return time to the depot cannot be later than a fixed time, that in our case is set to 1 hour after the end of the time window of the depot. In other words, a late return up to 1 hour is penalized but still feasible, a longer one is infeasible.
H3 The site reachability has to be guaranteed for each pair customer-vehicle.

The objective function is a weighted linear combination of many components that are related to the features described in Section 2.1. The cost components are the following:

S1 Delivery of an order in a day not included in its delivery days.
S2 Delivery of an order in a time outside its window.
S3 Optional orders not delivered.
S4 Cost of using a vehicle (as described in Section 2.2).

Notice that S1, S2, and S3 are soft constraints, while S4 refers to the cost to be minimized. In addition, we have a component that takes into account the *empty space* in each vehicle:

S5 The empty space in the vehicles.

This is performed by a non-linear function that penalizes a balanced load on all vehicles in favour of large free spaces in some of them and small in others. This latter case is preferable because large unused spaces can be more easily used for possible unforeseen changes and additional loads.

The weights of the various components are not fixed, but they are set by the operator for each specific instance.

3 Related Work

The Vehicle Routing Problem was first introduced by Dantzig and Ramsey [4] 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 developed (see the review by Toth and Vigo [5]).

We review here the articles in the literature that deal with those variants of the VRP that are the closest to our problem.

Baldacci *et al* [6] give 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 [7,8] and on sweep-based algorithm [9].

The first metaheuristic for HVRP was proposed in [10]. Then tabu search approaches for this problem family are developed in [11,12,13]. Ochi *et al* [14] propose

a parallel genetic algorithm in conjunction of scatter search. Tarantilis *et al* [15,16] present a list-based threshold accepting metaheuristic where a randomly selected solution is accepted only if the difference between the cost of the new solution and the cost of the current one is less than a control parameter that is reduced during the search process in a non-monotonic schedule. Liu *et al* [17] develop a record to record travel algorithm and they propose a new set of five test instances that have 200–360 customers.

The HVRP with Time Windows (*HVRPTW*) has been much less studied than the HVRP. The first work is reported by Liu and Shen [18], where they develop a saving-based construction heuristic. They also create three sets of instances to test this new problem variant. More recently, Dell'Amico *et al* [19] proposed an algorithm based on a parallel insertion procedure. Bräysy *et al* [20] presented a solution approach divided in three phases: firstly, a saving-based heuristic creates an initial solution, then a local search procedure focuses on the reduction of the vehicle and finally, the solutions are improved by a simulated annealing metaheuristic.

The VRP with Private fleet and Common carrier (*VRPPC*) describes a situation where, for hypothesis, the total demand exceeds the capacity of the internal fleet, so external transporter is necessary (*common carrier*). The objective is to supply customers, using the private fleet or the common carrier, minimizing the routing costs. VRPPC was first proposed in [21], but, in spite of its practical importance, up to our knowledge it has attracted less research effort. The problem is solved in [22,23,24] always using heuristic approaches.

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* in [25], but the definition of TOP was introduced by Chao *et al* in [26] that proposed a heuristic algorithm. Tabu search approaches have been recently presented by [27,28].

4 Application of Tabu Search

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 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 this case, first of all a solution defines if the order is scheduled or it is excluded. Then, for each scheduled order, the solution has to specify the day when the order is

delivered, the vehicle, and the position in the corresponding route (the arriving time at the client is deterministically computed given the position).

The initial solution is selected at random. That is, we create a state of the search space that satisfies only the constraints about the delivery day and the site reachability. This is made by assigning each order to a randomly selected feasible day of the planning horizon, then to a random vehicle between those that can reach the customer location and finally to a random position in the selected route. At this first stage, all the orders are scheduled.

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. In our problem, we are dealing with the assignment of an order to three kinds of resources: the days, the vehicles and the position in the route.

The first neighborhood, called Insertion (Ins), is defined by simply removing an order from a route and inserting it in another one in a specific position. An order can also be inserted in the list of the unscheduled ones (the position is indifferent) 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 is irrelevant.

A move of type Ins is identified by 5 attributes $\langle o, or, op, nr, np \rangle$ 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.

The Inter-route swap (InterSw) neighborhood, instead, is defined by exchanging an order with another one that is in a different route. A move of this type is identified by 6 attributes $\langle o1, o2, r1, r2, p1, p2 \rangle$, where $o1$ and $o2$ are orders, $r1$ and $r2$ are the routes of $o1$ and $o2$, $p1$ and $p2$ are the positions of the orders in the routes.

Finally, the Intra-route swap (IntraSw) neighborhood, is defined by exchanging an order with another one in the same route. A move of this type is identified by 5 attributes $\langle o1, o2, r, p1, p2 \rangle$, where $o1$ and $o2$ are orders, r is the route, $p1$ and $p2$ are the positions of the orders in the route.

4.3 Prohibition Rules

A prohibition rule determines which moves are prohibited by the fact that a given move m is tabu. The prohibition rules are based on the values of the attributes of the two moves, the one in the list and the one under evaluation.

For each neighborhood structure we have tested several prohibition rules of different restrictive level. For the Ins neighborhood, assuming that the move $m_t = \langle o_t, or_t, op_t, nr_t, np_t \rangle$ 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 6 alternatives shown in Table 1. The last column shows the percentage (on all moves) of the entire neighborhood that a single move prohibits if it happens to appear in the tabu list. In detail, for a given state, the value is calculated as the ratio of number of couple of moves where the first one is the inverse of the other to the total number of couple of moves.

Rule	Condition	Description of tabu moves	Tabu level
PR1	$o_e = o_t \wedge or_e = nr_t$	moves removing the order o_t from the route nr_t	0.098%
PR2	$nr_e = or_t \wedge or_e = nr_t$	moves putting back any order from nr_t to or_t	0.417%
PR3	$o_e = o_t \wedge nr_e = or_t$	moves reinserting the order o_t in the route or_t	1.794%
PR4	$o_e = o_t$	moves involving the same order o_t	1.799%
PR5	$or_e = nr_t$	moves removing any order from the route nr_t	5.054%
PR6	$nr_e = or_t$	moves reinserting any order into the route or_t	7.923%

Table 1 Prohibition rules for TS(Ins)

Similarly, we have tested several prohibition rules also for the two neighborhoods IntraSw and InterSw. However, for the sake of brevity, we report here 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

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

is imposed. It forbids to make a move where any order of m_e is equal to the first order of any move m_t belonging to the tabu list.

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

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

is the condition imposed; so that, it is forbidden to make a move that involves the same two routes of the tabu one.

4.4 Tabu Dynamics

We make use of the Robust Tabu Search scheme, as called in [3]. In this scheme the tabu tenure is *variable*, consequently each performed move remains in the list for a number of iterations randomly selected between a minimum and a maximum value. In detail, we set two values t and δt and we assign to each accepted move a tabu tenure randomly selected between $t - \delta t$ and $t + \delta t$. In all our experiments δt is set to 2 (based on preliminary runs), whereas t is subject to tuning.

4.5 Search Techniques

Based on the three neighborhood relations, we have three basic Tabu Search techniques, that we call TS(Ins), TS(IntraSw), TS(InterSw).

Notice however that under the IntraSw and InterSw neighborhood structures the search space is not connected. In fact, for both it is not possible to 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. Therefore, for our problem TS(IntraSw) and TS(InterSw) are not effective as search techniques by themselves, but they must be used in combination with TS(Ins).

We use a sequential solving strategy for combining tabu search algorithms based on different neighborhood functions, as proposed (among others) in [29]: given an initial

Name	#O	#C	#V	#VT	#R	#D
PAC1-A	139	44	7	4	19	4
PAC1-B	166	56	7	4	22	4

Table 2 Features of the instances

state and a set of algorithms, its search 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 all of them 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 thus identify five strategies (denoted by the $\triangleright$ symbol in [29]) resulting of the combination of the basic neighborhood structures: TS(Ins), TS(Ins)$\triangleright$TS(IntraSw), TS(Ins)$\triangleright$TS(InterSw), TS(Ins)$\triangleright$TS(IntraSw)$\triangleright$TS(InterSw), and TS(Ins)$\triangleright$TS(InterSw)$\triangleright$TS(IntraSw).

We also consider solvers that use the union of many neighborhood: the algorithm based on this compound neighborhood (denoted by the $\oplus$ symbol in [29]) selects, at each iteration, a move belonging to any of the neighboorhoods as part of the union. We compare for example the TS(Ins)$\triangleright$TS(InterSw) solver with the TS(Ins$\oplus$InterSw) one, that makes use of the union of Ins and InterSw. We do not report the results in detail, because preliminary experiments show that the union regularly gives worse results.

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 simpler algorithm which uses one single neighborhood, then moving to the more complex ones.

5.1 Benchmark Instances

We have four instances coming from real cases: two of them completely independent, the others with the same data but a different partition of orders into mandatory and optional. For each instance, we tested three different realistic configurations of the weights of the cost components. The configurations of the weights make the solutions quite different from each other, thus resulting in 12 independent instances.

Table 2 summarizes the ranges of the size of the instances in terms of number of orders (#O), number of clients (#C), number of vehicles (#V), number of vehicle types (#VT), number of regions (#R) and number of days of the planning period (#D).

5.2 General Settings and Implementation

All the algorithms have been implemented in the C++ language, exploiting the EasyLocal++ framework [30]. The experiments were performed on an Intel QuadCore PC (64 bit) running Debian Linux 4.0, the software is compiled using the GNU C++ compiler (v. 4.1.2).

In order to compare different combinations in a fair way, we grant to all of them the same running time for each run. In particular, in all experiments we grant 100

Prohibition Rule	Tabu Length	Avg. Cost	p-value
PR5	15	230345.72	—
PR6	35	233419.30	$3.29 \cdot 10^{-13}$
PR4	35	232501.85	$7.81 \cdot 10^{-29}$
PR1	35	232661.77	$1.49 \cdot 10^{-29}$
PR3	45	237109.48	$6.40 \cdot 10^{-52}$
PR2	40	242538.63	$9.95 \cdot 10^{-64}$

Table 3 Comparison of prohibition rules for TS(Ins)

seconds for each trial, and we run 30 trials for each configuration of the parameters. The random seed is used as *blocking factor*, so that all configurations have the same 30 random seeds, and thus they start from the same initial solution.

For each individual TS(·) component of the search strategy the maximum number of iterations from the last improvement is set to 500. The full strategy stops only based on the timeout, independently of the improvements obtained in the last rounds.

To compare the results we use the *Student's t-test*, provided that the underlying distributions can be assumed to be normal and independent [31].

If the calculated p-value is below the threshold chosen for statistical significance (typically 0.05), then the null hypothesis which usually states that the two groups do not differ, is rejected in favor of an alternative hypothesis, which states that an algorithm is superior to another on the proposed instances.

The t-test has two different versions depending on whether the two samples are:

- unpaired, independent of each other, or
- paired, so that each member of one sample has a unique relationship with a particular member of the other sample.

In our experiments, we use the paired version by associating each run of one algorithm with the one of the other algorithm on the same instance and having the same random seed (thus the same initial solution).

5.3 Results on Prohibition Rules for the Insertion Neighborhood

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

Obviously, for each prohibition rule the best tabu 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.

Table 3 shows the comparison between the best settings of each rule. It presents, for each rule, the selected length value and the average cost, in ascending order. The last column shows the p-value of the comparison between the best configuration (PR5, first line) and the current one. We remember that a small p-value (lower than 0.05, separated by an horizontal line in the tables) means that the two distribution are significantly different and that one configuration is superior to the other on these instances.

The table shows that the p-values are extremely small, the the prohibition rule PR5 is clearly superior to all the others. It also shows that for the selected rule PR5 the best length is considerably smaller than for all the others. In the subsequent experiments we use only this rule.

IntraSw		
Tabu Length	Avg. Cost	p-value
6	231659.8	—
4	231700.04	0.50
8	231719.22	0.32
10	231742.35	0.16

InterSw		
Tabu Length	Avg. Cost	p-value
30	230210.94	—
20	230256.03	0.66
25	230333.95	0.05
35	230482.98	0.04

Table 4 Comparison of tabu length of TS(IntraSw) and TS(InterSw)

Solver	Avg. Cost	p-value
TS(Ins)▷ TS(InterSw)	230430.12	—
TS(Ins)▷ TS(IntraSw)▷ TS(InterSw)	230456.04	0.67
TS(Ins)▷ TS(InterSw)▷ TS(IntraSw)	230504.91	0.16
TS(Ins)▷ TS(IntraSw)	231109.04	$7.55 \cdot 10^{-13}$
TS(Ins)	231146.45	$5.525 \cdot 10^{-14}$

Table 5 Comparison of composite solvers

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

As already noted, TS(InterSw) and TS(IntraSw) cannot be used alone because the search space is not connected under their neighborhood relation, thus only a subset of all possible solution would be explored. Consequently, in order to tune the parameters of TS(InterSw) and TS(IntraSw) we make use of the strategies TS(Ins)▷TS(InterSw) and TS(Ins)▷TS(IntraSw), respectively.

We report in Table 4 the tuning of the tabu length of TS(IntraSw) and TS(InterSw), keeping the parameters of TS(Ins) as set in the previous experiment.

Table 4 shows that TS(IntraSw) is not much affected to tabu length variations (p-values are high). Anyway the best configuration has a short tabu list. The outcome is not distinct also for TS(InterSw), in fact the most fitting tabu length is 30, but 20 wins over 25.

5.5 Results on Composite Solvers

Our last experiment regards the comparison of the composite solvers. For each component of the solvers, the parameters are set to the value obtained in the previous experiments.

Table 5 reports the results, ordered as usual by the p-values w.r.t. to the best configuration.

The outcome is that the best solver is TS(Ins)▷TS(InterSw), but there are other two combinations, namely TS(Ins)▷TS(IntraSw)▷TS(InterSw) and TS(Ins)▷TS(InterSw)▷ TS(IntraSw), that are not statistically distinguishable.

6 Conclusions and Future Work

We have proposed a hybrid metaheuristic approach based on tabu search for a highly complex version of the classical vehicle routing problem arising from a real world situation.

The experimental analysis shows that the best results are obtained by a combination of Ins and InterSw neighborhood, whereas the use of the IntraSw one is not useful in our cases.

For the future we plan to test the use of other neighborhood relations and other techniques and to compare them in a principled way. In addition, 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 thank Luca Di Gaspero for helpful comments on the paper. We are also 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

1. Toth, P., Vigo, D. (eds.): The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. S.I.A.M., Philadelpia, PA (USA) (2002)
2. Glover, F., Laguna, M.: Tabu search. Kluwer Academic Publishers (1997)
3. Hoos, H.H., Stützle, T.: Stochastic Local Search – Foundations and Applications. Morgan Kaufmann Publishers, San Francisco, CA (USA) (2005)
4. Datzig, G., Ramser, J.: The truck dispatching problem. Management Science **6**(1), 80–91 (1959)
5. Toth, P., Vigo, D.: An overview of vehicle routing problems. In: The vehicle routing problem, pp. 1–26. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2001)
6. Baldacci, R., Battara, M., Vigo, D.: Routing a heterogeneous fleet of vehicles. Technical report deis or. inge 2007/1, DEIS, University Bologna, via Venezia 52, 47023 Cesena, Italy (2007)
7. Taillard, E.: A heuristic column generation method for the heterogeneous fleet vrp. RAIRO Recherche Oprationnelle **33**(1), 1–14 (1999)
8. Choi, E., Tcha, D.W.: A column generation approach to the heterogeneous fleet vehicle routing problem. Comput. Oper. Res. **34**(7), 2080–2095 (2007)
9. Renaud, J., Boctor, F.F.: A sweep-based algorithm for the fleet size and mix vehicle routing problem. European Journal of Operational Research **140**, 618–628 (2002)
10. Semet, F., Taillard, E.: Solving real-life vehicle routing problems efficiently using tabu search. Annals of Operations Research **41**, 469–488 (1993)
11. Osman, I.H., Salhi, S.: Local search strategies for the vehicle fleet mix problem. Modern Heuristic Search Methods (1996)
12. Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E.D.: A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research **26**(12), 1153–1173 (1999)
13. Wassan, N.A., Osman, I.H.: Tabu search variants for the mix fleet vehicle routing problem. Journal of the Operational Research Society **53**(7), 768–782 (2002)
14. Ochi, L.S., Vianna, D.S., Drummond, L.M.A., Victor, A.O.: A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. Parallel And Distributed Processing **1388**, 216–224 (1998)
15. Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S.: A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. Journal of the Operational Research Society **54**(1), 65–71 (2003)
16. Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S.: A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research **152**(1), 148–158 (2004)
17. Li, F., Golden, B., Wasil, E.: A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Comput. Oper. Res. **34**(9), 2734–2742 (2007)
18. Liu, F.H., Shen, S.Y.: The fleet size and mix vehicle routing problem with time windows. Journal of the Operational Research Society **50**(7), 721–732 (1999)

19. Dell'Amico, M., Monaci, M., Pagani, C., Vigo, D.: Heuristic approaches for the fllet size and mix vehicle routing problem with time windows. Tech. rep., DISMI, University of Modena and Reggio Emilia, Italy (2006)
20. Bräysy, O., Dullaert, W., Hasle, G., Mester, D., Gendreau, M.: An effective multi-restart deterministic annealing metaheuristic for the fleet size and mix vehicle routing problem with time windows. Transportation Science (2006)
21. Volgenant, T., Jonker, R.: On some generalizations of the travelling-salesman problem. Journal of the Operational Research Society **38**(11), 1073–1079 (1987)
22. Diaby, M., Ramesh, R.: The distribution problem with carrier service: a dual based approach. ORSA Journal on Computing **7**(1), 24–35 (1995)
23. Bolduc, M., Renaud, J., Boctor, F.: A heuristic for the routing and carrier selection problem (2005). Working paper
24. Chu, C.W.: A heuristic algorithm for the truckload and less-than-truckload problem. European Journal of Operational Research **127**(3), 657–667 (2005)
25. Butt, S.E., Cavalier, T.M.: A heuristic for the multiple tour maximum collection problem. Comput. Oper. Res. **21**(1), 101–111 (1994)
26. Chao, I.M., Golden, B.L., Wasil, E.A.: The team orienteering problem. European Journal of Operational Research **88**(3), 464 – 474 (1996)
27. Tang, H., Miller-Hooks, E.: A tabu search heuristic for the team orienteering problem. Comput. Oper. Res. **32**(6), 1379–1407 (2005)
28. Archetti, C., Hertz, A., Speranza, M.: Metaheuristics for the team orienteering problem. JH **3**, 49–76 (2007)
29. Di Gaspero, L., Schaerf, A.: Neighborhood portfolio approach for local search applied to timetabling problems. Journal of Mathematical Modeling and Algorithms **5**(1), 65–89 (2006)
30. Di Gaspero, L., Schaerf, A.: EasyLocal++: An object-oriented framework for flexible design of local search algorithms. Software—Practice and Experience **33**(8), 733–765 (2003)
31. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer (2003)