Journal of Applied Operational Research (2014) **6**(2), 69–81

ISSN 1735-8523 (Print), ISSN 1927-0089 (Online)

A savings-based randomized heuristic for the heterogeneous fixed fleet vehicle routing problem with multi-trips

Jose Caceres-Cruz [1,*], Alex Grasas [2], Helena Ramalhinho [2], Angel A. Juan [1]

[1] *Open University of Catalonia, Barcelona, Spain*
[2] *Universitat Pompeu Fabra, Barcelona, Spain*

Abstract. The Vehicle Routing Problem is one of the most studied problems in the Operations Research literature, and is also one of the most applied problems in several industries from retailing to health care. In this paper we present a new version of the Vehicle Routing Problem, the Heterogeneous Fixed Fleet Vehicle Routing Problem with Multi-trips and maximum route distance in which the use of the entire fleet is forced before assigning a second trip. Our work is motivated by a real routing problem of a large-size distribution company. To solve the problem, we propose a randomized hybrid algorithm based on the well-known Clarke and Wright's Savings heuristic. Some computational experiments, using both academic benchmarks and real data from the company, contribute to validate the competitiveness of our approach.

Keywords: vehicle routing; multi-trip routing plans; savings heuristic; randomized algorithms; real-life applications

Received September 2013. Accepted December 2013

Introduction

Road route planning has become a relevant activity for most logistics and transportation companies due to its considerable cost implications. In the academic literature, these decision-making issues are categorized as Vehicle Routing Problems (VRP), a popular research stream that has undergone significant theoretical advances in the last decades. The VRP is a class of problems related with designing the least-cost delivery routes from a depot to a set of customers. These problems are central in the distribution and logistics management due to the importance of efficient routing to reduce costs and increase customer satisfaction. The classical version is the Capacitated Vehicle Routing Problem (CVRP), defined by Dantzig and Ramser (1959). In a CVRP, a direct graph $G = (V, A)$ is given, where $V = \{0,1, \dots, n\}$ is the set of $n + 1$ nodes and A is the set of arcs. Node 0 represents the depot, while the remaining nodes $V^{'} = V/\{0\}$ correspond to the n customers. Each customer $i \in V'$ requires a known supply of q_i units, i.e., its demand, from a single depot (assume $q_0 = 0$). This demand is going to be serviced by exactly one visit of a single vehicle. In this basic form, there is a homogeneous fleet of m identical vehicles with capacity Q to serve these n customers. It is assumed that the total fleet capacity is greater than or equal to the total demand. CVRP aims at finding m trips (vehicles) so that all customers are serviced and the total distance traveled by the fleet is minimized.

**Correspondence*: Jose Caceres-Cruz, Department of Computer Science, Multimedia, and Telecommunication, Open University of Catalonia, Avinguda del Tibidabo, 39, 08035 Barcelona, Spain.
E-mail: jcaceresc@uoc.edu

We focus our study on the case of heterogeneous vehicles, an assumption commonly observed in practice (e.g., retailing). Most road-transportation companies own a heterogeneous fleet because different routes might require different types of vehicles, e.g., narrow roads, available parking spaces, vehicle weight restrictions on certain roads, etc. Additionally, when vehicles are acquired over time, they are not necessarily of the same type. Also, it is common for most companies to have a limited number of vehicles available, in need of several trips per truck whenever the demand in a given time period is greater than its fleet capacity. Despite its practical relevance, this type of problems has received limited attention from academic literature (Laporte, 2009; Mingozzi *et al*., 2013). To contribute filling this gap, we present the Heterogeneous Fixed Fleet Vehicle Routing Problem with Multi-trips (HFVRPM) with maximum route distance. The HFVRPM is a generalization of the classical VRP that (i) considers a heterogeneous fleet, i.e., a fixed fleet of vehicles with different capacities, and (ii) allows a subset of these vehicles to perform multiple trips in a finite period of time (working period) if necessary. This problem is inspired on a real routing problem of a large-size distribution company operating more than 370 stores in the Northeast of Spain.

To solve the HFVRPM, we propose a hybrid algorithm that combines a randomized Clarke and Wright's Savings (CWS) heuristic (Clarke and Wright, 1964) and three local search methods: 2-Opt, a temporary memory of the best routes found, and a splitting technique. This is a simple algorithm that does not need any complex fine-tuning or special assumption on the heterogeneous VRP scenario. Therefore, its use by the company has the advantage of not requiring any special knowledge or skills to obtain good distribution routes.

The remainder of this article is structured as follows: Section 2 presents a review of the relevant literature associated to the studied problem; Section 3 describes the HFVRPM variant we analyze in more detail; Section 4 gives an explanation of our approach; Section 5 presents and discusses some numerical experiments that serve to both illustrate and validate our approach; finally, Section 6 summarizes the main contributions and results of this work.

Review of heterogeneous and multi-trip routing problems

To better understand the definition of the HFVRPM and its relationship to other VRP variants, we make the following classification in Figure 1 and provide a review of the relevant literature.

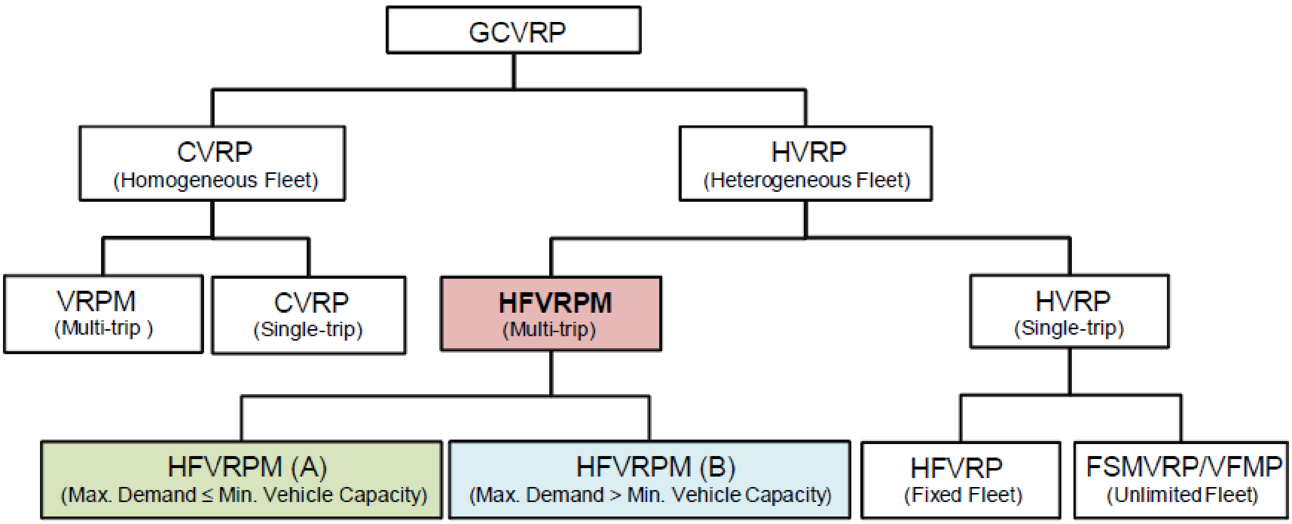

Fig. 1. Classification of the Vehicle Routing Problems related to HFVRPM

At the top of the chart there is the General Capacitated Vehicle Routing Problem (GCVRP). Depending on the fleet composition, it can be divided into two subproblems: the Homogeneous Fleet Vehicle Routing Problem (CVRP), where all vehicles have equal capacity; and the Heterogeneous Fleet Vehicle Routing Problem (HVRP), where vehicles have different capacities. We can further divide these two subproblems according to the vehicle availability. That is, if there exists a large number of vehicles or enough capacity to distribute all the demand, i.e., $\sum_{i=1}^{n} q_i \ll mQ$, each vehicle will only make one trip. This problem is the Single-trip CVRP and is the most well-known and studied problem in the class of GCVRP (Laporte, 2009; Toth and Vigo, 2002). It is generally found in the context of transportation outsourcing, where the number of vehicles is not an issue since the transportation is outsourced to a third party logistics company, with usually a large fleet of vehicles.

On the other side, when $\sum_{i=1}^{n} q_i > mQ$, it is not possible to serve all demand with the current vehicle availability, therefore vehicles are allowed to make multiple trips. This problem, the VRP with Multi-trips (VRPM), seems more relevant in practice (for more information see Baldacci *et al.*, 2008; Sen and Bulbul, 2008; and Golden *et al.*, 2008) as distribution and retailing companies do not have an unlimited number of vehicles. Despite its importance, few authors have addressed it in the past (Mingozzi *et al.*, 2013). The methods usually applied to solve this multi-trip version are based on the CWS and Tabu Search approaches. Fleischmann (1990), for instance, combines a savings heuristic with a Bin Packaging Problem (BPP) technique. The BPP is also used by Petch and Salhi (2003), where the authors combine it with the savings method in a multi-phase approach. Taillard *et al.* (1996); Brandao and Mercer (1998); and Olivera and Viera (2007) propose Tabu Search metaheuristics. Taillard *et al.* (1996) also define a set of instances that have been used by other authors as benchmarks. Multi-objective approaches based on Tabu Search and Simulated Annealing metaheuristics have been presented by Lin and Kwok (2006). The authors have compared their approaches with real and simulated data. Several of these papers include real applications to test the performance of their algorithms. In general, to solve real problems, CWS approaches lead to simple algorithms (Poot *et al.*, 2002) when compared to Tabu Search methods as no fine-tuning is needed to get good performances.

Recently, heterogeneous fleet problems are receiving increased attention from researchers inspired by real-life situations. In the general HVRP, there is a vehicle fleet M composed of m different vehicle types, i.e., $M = \{1, \dots, m\}$. The m_k vehicles of type $k \in M$ have capacity Q_K, fixed cost F_k, and variable cost per arc (i, j) traveled c_{ij}^k. A route is feasible if the total demand of customers visited by a type k vehicle does not exceed its capacity Q_K. The total cost of a solution (set of routes) is based on the total distance plus a fixed cost for using a vehicle, F_k. Baldacci *et al.* (2008); and Prins (2009) present an overview, classification and survey on the different HVRPs in the literature. Several approaches that include exact methods, heuristics and metaheuristics methods have been proposed to address the HVRP. Due to the intrinsic difficulty of this VRP family, most research focuses on the development of heuristics and metaheuristics that provide near optimal solutions for medium- and large- sized problems with a myriad of complex restrictions. Many of the successful approaches for the CVRP have been adapted to a HVRP context.

For heterogeneous fleets, almost all research focuses on single-trip models. Among these, two cases may arise according to the vehicle availability (see Figure 1): the Heterogeneous Fixed Fleet VRP (HFVRP) with a limited number of vehicles of each type, and the Fleet Size and Mix VRP (FSMVRP) also known as Vehicle Fleet Mix Problem (VFMP), with unlimited number of different vehicles. The objective in the latter is also to find the best fleet composition. Golden *et al.* (1984) were the first to address the FSMVRP with fixed costs. The authors propose a greedy heuristic over a set of 20 instances. Later, Taillard (1999) develops a Column Generation technique based on an adaptive memory procedure for generating a large number of routes (for each vehicle type) and ends using a linear programming algorithm to find the best solution. He addresses the HFVRP and the FSMVRP with fixed and variable costs, and proposes a modification of eight large-scale instances from Golden *et al.* (1984). Osman and Salhi (1996) address the FSMVRP with fixed costs and propose a Tabu Search method to solve it. For the FSMVRP, other Tabu Search techniques have been developed by Gendreau *et al.* (1999); and Brandao (2011).

For the HFVRP, Tarantilis *et al.* (2004) use a threshold accepting metaheuristic called Backtracking Adaptive Threshold Accepting and test the Taillard instances; Li *et al.* (2007) create a variant of a record-to-record travel algorithm, which is a deterministic case of the Simulated Annealing metaheuristic and outperforms Tarantilis *et al.* (2004)'s approach on the Taillard instances; and Euchi and Chabchoub (2010) apply a hybrid Tabu Search based on an adaptive memory. On the line of Genetic Algorithms, Prins (2009) presents two memetic algorithms hybridized with a local search able to solve both HFVRP and FSMVRP with fixed costs, variable costs or a combination of both. He obtains good results on the 20 instances of Golden *et al.* (1984), establishing six new best solutions.

Heterogeneous fixed fleet vehicle routing problem with multi-trips (HFVRPM)

In this paper, we study a version of the HVRP that considers a fixed fleet and allows multiple trips (HFVRPM). The HFVRPM is motivated by the work we have been conducting with a distribution company in Spain, although there exist other real applications. For example, the existence of fleets of different sizes and types for each waste classification type is common in household and commercial waste collection. For other applications, the reader is referred to Chao *et al*. (1999). Within this problem, we have considered two variants: one where the maximum demand of a given customer is smaller than the minimum vehicle capacity (HFVRPM (A)) and the other where the maximum demand of a given customer is greater than the minimum vehicle capacity (HFVRPM (B)). This is an important difference as any vehicle can serve any customer in the former, but not in the latter. The real application that motivates this study is an example of the second case, and it has never been studied before to the best of our knowledge. In addition, we also assume the following considerations regarding the available fleet and its costs:

a) The objective function is to minimize total distance.
b) The number of vehicles of each type, m_k.
c) For each vehicle type:
   i. The fixed costs are ignored (i.e., $F_k = 0, \forall k \in M$);
   ii. The routing costs are vehicle-independent ($c_{ij} = c_{ij}^k = c_{ji}^l, \forall\, k, l \in M$ and $\forall (i,j) \in A$);
   iii. There are no restrictions on the customers they can visit (due to size or maneuverability, for example).
d) Some vehicles can make multiple trips from the depot (i.e., multi-trips).
e) There is a maximum route length for all vehicles in a routing planning of one day.
f) All vehicles must be used at least once before allowing multi-trips.

Prins (2002) describes a similar problem to the HFVRPM (A) and is the most related work. The author also imposes a time restriction on routes (i.e., driver's time of 300 minutes) and that the largest demand cannot exceed the smallest truck capacity. This second assumption, which we relax in HFVRPM (B), is often not satisfied in reality. Prins' assumption unveils an interesting discussion about the use of small vehicles, those that cannot handle the maximum customer‘s demand in a given instance. Optimization algorithms tend to ignore small vehicles to the detriment of bigger ones, but in reality, many companies use them for short and frequent trips. Another important difference with Prins is the vehicle assignment (part f, above). In this problem, (i) all vehicles can visit all customers (ignoring access issues); (ii) there are no costs associated to the use of vehicles; and (iii) the routing planning must promote the use of all vehicles before assigning a second trip (a new trip is counted each time the vehicle leaves the depot). This last restriction differs from other VRP with multi-trips approaches in the literature in which the use of vehicles from a homogeneous fleet has to be minimized. Figure 2 depicts a simplified example with a fleet of two vehicle types and three vehicles in total that shows the differences of vehicle assignment between our approach and the most common in literature. On the left (A), we have the assignment according to previous VRP studies with multi-trips where vehicle "Type A Number 2" is not used but the other vehicles perform two trips. On the right (B), however, according to our approach, all vehicles are used once and then vehicle "Type A Number 1" performs two trips.

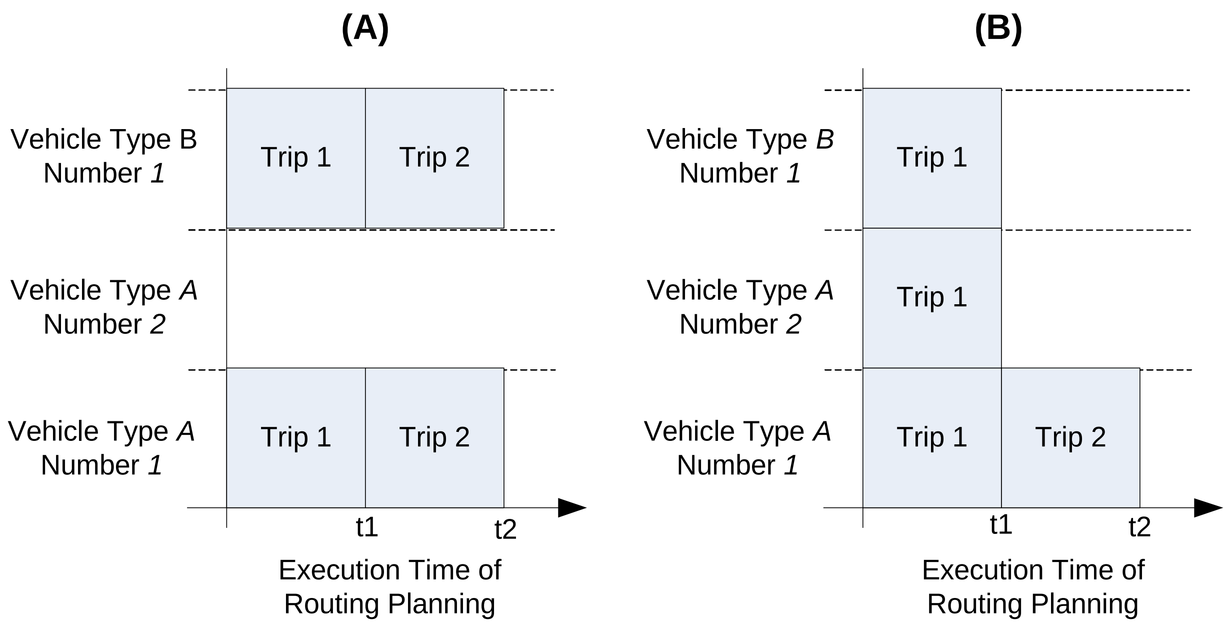

Fig. 2. Example of the use of vehicles in the addressed HFVRPM

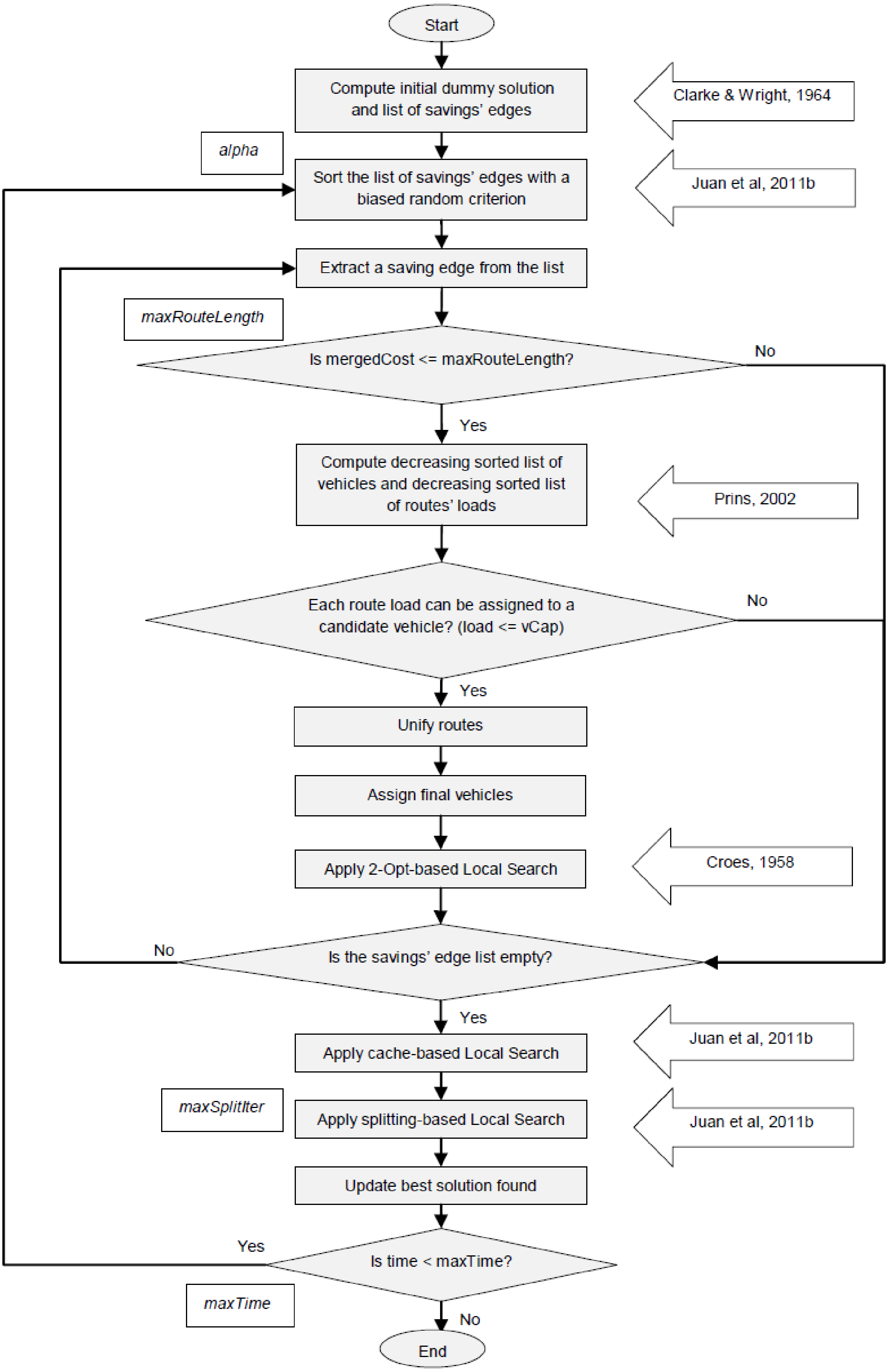

Fig. 3. Flowchart of our approach

Solution method

This section explains our approach, called Rand-MER, for solving the HFVRPM and discusses some of its main design properties, such as: (a) the biased randomization of the Prins (2002)'s MER deterministic heuristic, which allows transforming it into a multi-start probabilistic algorithm; and (b) the use of two additional local search methods developed in Juan *et al*. (2011b), which are based on cache and splitting techniques.

We use a heuristic based on the SR-GCWS (Simulation in Routing via the Generalized Clarke and Wright's Savings heuristic) algorithm originally proposed by Juan *et al*. (2011b) to solve the CVRP. One motivation to apply the CWS method to the HFVRPM was to achieve the desirable metaheuristic features described by Cordeau *et al*. (2002): accuracy, speed, simplicity and flexibility. The SR-GCWS is a hybrid algorithm that combines the parallel version of the classical CWS heuristic with Monte Carlo simulation and random number generators to produce a set of alternative solutions for a given CVRP instance.

As to savings approaches for the HVRP, Christofides and Eilon (1969) have made a preliminary comparison between the CWS algorithm with a branch-and-bound approach and a 3-Opt tour approach for addressing the sub-problems of HVRP. The authors test the approaches with a set of 10 proposed instances. The last approach outperformed the first one. Golden *et al*. (1984) propose an adaptation of the CWS heuristic for the FSMVRP, combining the concept of heterogeneous fleet within the saving. Then the saving formula considers the variation of the fixed costs associated with the route merging. Prins (2002) uses a special adaptation of the CWS and proposes two algorithms for single- and multi-trips contexts: the first, called New Merge Heuristic (MER) is responsible for the construction of routes, while the second, called MER2, uses a BPP method to adapt the single-trip algorithm to a multi-trip problem. We focus on the first one that includes a 2-Opt process after each positive two-route merging.

We also include a new component for managing fleets of vehicles combining the random version of CWS with two local search methods (Juan *et al*., 2011b). The first search is based on a memory (cache) to store the best routes among a group of nodes, and the second search is based on a splitting technique to find better solutions in smaller instances produced by the clustering of nodes. Figure 3 depicts a high-level flowchart of the proposed algorithm, whose overall description is included next.

Given a HFVRPM instance, the algorithm first constructs an initial solution as proposed in the classical savings heuristic. In this initial solution, a virtual truck is assigned to each customer. Also, as proposed in the aforementioned heuristic, the algorithm computes the savings associated with each edge connecting two different customers. Put in simple terms, the savings associated with a given edge are computed based on its cost reduction (distance and/or time-based). The edges are then stored into a list, which is sorted from highest to lowest savings. At this point, a multi-start procedure begins. Typically, this procedure is executed over and over until a time-based ending condition is reached. At each iteration of this multi-start procedure, the following steps are performed:

1. A new savings list of edges is obtained by randomizing the original savings list through the Geometric probability distribution, as suggested in Juan *et al*. (2011b). By randomizing the savings list used on CWS, the deterministic heuristic described in the next step is transformed into a probabilistic algorithm – as it is also GRASP (Greedy Randomized Adaptive Search Procedure) (Festa and Resende, 2009). Therefore, each combination of edges has a chance of being selected. Since the construction phase tends to select a different set of edges, this type of algorithm helps to explore the feasible solution space in a diversified way (Juan *et al*., 2014). This allows obtaining different outputs at each iteration of the multi-start procedure. Furthermore, by using a biased probability distribution –the Geometric in this case– most of the logic behind the classical savings heuristic is kept, i.e., edges with higher savings will be more likely to be selected from the list than those with lower savings.
2. Then, until the savings list gets empty, an iterative process begins in which the edge at the top of the randomized list is extracted. This edge will connect two different routes. In order to merge these two routes, the extreme points of the edge must be 'ends' to their respective routes, i.e., they need to be directly connected to the depot. Moreover, both capacity and maximum-route-length constraints must be validated. A similar method to the one proposed by Prins (2002) is used to validate the capacity constraint in a heterogeneous fleet, i.e., the list of vehicles is sorted from highest to lowest capacity, while the list of routes is sorted from highest to lowest accumulated demands. After that, a temporary assignment between the two lists is searched; if a successful match including all previously merged routes plus the new one is found, then the capacity constraint is validated and the temporary assignment becomes final; otherwise,

the temporary assignment is discarded and the merge becomes infeasible. After each merge, a fast 2-Opt local search (Croes, 1958) is run over the new route.

3. Once all edges in the savings list have been considered, the resulting solution is then improved through the two local search methods proposed in Juan *et al.* (2011b): First, a cache (hash table) of routes is employed to quickly update any route in the current solution by the best-known route –among those routes found in previous iterations– covering the same set of nodes. Second, using proximity criteria, the current solution is divided into several sets of routes together with their associated vehicles; then, each of these subsets is considered as a smaller and feasible HVRP problem over which steps 1 and 2 above can be applied to find better 'local' routing plans. These local searches do not change the assigned vehicles; instead they try to find a better configuration of nodes. As a last step, the enhanced solution provided by these local search methods is compared against the best solution obtained so far by the multi-start procedure and, whenever appropriate, this best solution is updated. The combination of our algorithm with local search methods can help finding better quality solutions. Some local searches can be focused on improving the node configuration inside a route or proposing better vehicle assignments to build routes. However the application of these techniques can also consume computational time. So a balance between the local search method and the diversified exploration of the heuristic method is necessary for an efficient approach.

Eventually, once the time-based criterion is reached, the best solution found by the described multi-start procedure is the one returned. An interesting property of this approach is that it can be naturally and easily parallelizable. In effect, due to its probabilistic nature, the searching path followed by the aforementioned procedure greatly depends upon the seed employed to initialize the pseudo-random number generator –which is used during the randomization of the savings list. Therefore, using an object-oriented programming terminology, it is possible to simultaneously run different 'instances' of the algorithm 'class' by simply changing the initial seed. These independent instances can then be run in different threads, cores, or even computers, as discussed in Juan *et al*. (2011a).

Algorithm pseudo-code

Our Rand-MER algorithm (Figure 4) needs as input data: (i) the nodes information (geographical location and individual demands), (ii) the costs of moving from one node to another, and (iii) the fleet composition. The procedure requires four parameters: (1) *alpha*, associated with the Geometric distribution employed during the randomization process of the savings list; (2) *maxTime*, the stopping time of the multi-start process; (3) *maxRouteLength*, associated with the maximum route length allowed; and (4) *maxSplitter*, associated with the maximum number of iterations for the splitting local search.

First (line 1), the procedure generates a list of edges connecting any two nodes. This list is sorted according to the savings obtained when using each edge. Then, an initial dummy solution is generated (line 2). In this solution, one round-trip route starting at the depot is considered for each client. After that, a multi-start process begins (lines 3-27). This multi-start process is especially useful for several reasons: (a) it allows the randomized algorithm to escape from local minima; and (b) it facilitates parallelization of the approach –this can be achieved by running different instances of the algorithm, each one using a different seed for the pseudo-random number generator. At each iteration of the multi-start process, a biased randomization of the savings list is produced (line 5). At the end of each iteration, a new solution is iteratively constructed by merging routes, if feasible, according to the randomized list (lines 6-19). Each merged route is improved through a classical 2-Opt local search process (line 17). Once a solution is generated, a memory-based local search process is applied (line 20). In this process, each route in the solution is checked against a cache, which contains the best found-so-far route covering the same set of nodes. Moreover, if the solution can be considered a "promising" one, a divide-and-conquer process is also applied (lines 21-23). Eventually, the algorithm returns the best solution produced in the entire multi-round process (line 28).

One of the key steps in our approach is the vehicle assignment process that takes place during the merging of any two routes. Figure 5 shows the pseudo-code of the procedure employed to validate a potential merging as feasible. Notice that in order to merge two routes, three conditions must be satisfied (lines 1-3). First, both nodes in the connecting edge must be external nodes, i.e., both have to be directly connected with the depot. Second, the length of the merged route cannot be greater than the maximum allowed. Finally, it must be possible to cover each merged route with a truck. To check this condition, all merged routes (including the new one) are sorted from the highest to the lowest aggregated demand (line 5), while all vehicles are similarly sorted by capacity (line 6).

Then, starting from the top of both lists, the next vehicle is assigned to the next merged route as far as the truck capacity can cover the route demand (lines 4-13). If this assignment is not feasible after a certain point, then multi-trips are considered, i.e., vehicles already covering one route are assigned to a second one as long as traveling times allow to cover both routes in the specified time period. Of course, if some merging routes cannot be covered by any vehicle, then the potential merging process is discarded and a new potential merging is considered as long as the edges list is not empty. Notice that our approach allows for the realistic multi-trip scenario and, at the same time, it tries to use all vehicles in the fleet before assigning additional trips to some of them. This is a relevant difference with regards to the vehicle assignment proposed in Prins (2002), where multi-trips of larger vehicles are promoted and preferred over the use of the entire fleet. In our case, however, the company was interested in using the entire fleet in order to reduce total delivery times as much as possible.

```
procedure Rand-MER(nodes, costs, vehicles)
//Create Clarke & Wright's savings sorted list
01  savingsList ← computeSavingsList(nodes, costs)
//Build initial dummy solution
02  initialSol ← constructInitialSol(nodes, costs)
//Multi-start process with a stopping criteria
03  while stopping criteria not satisfied do // Defined by the MaxTime parameter
      //Reset the base solution to the dummy solution
04    currentSol ← initialSol
      //Biased randomized selection of savings
05    randomSavingsList ← biasedRandomization(savingsList)
      //Execute the route merging process
06    while random savings list is not empty do
07      savingEdge ← selectNextEdge(randomSavingsList)
08      routeA ← getStartingRoute(savingEdge)
09      routeB ← getClosingRoute(savingEdge)
        // Check merging conditions for CWS-HVRP approach
10      mergeIsValid ← validateMergeConstraints(savingEdge, routeA, routeB, vehicles, currentSol)
11      if mergeIsValid then
        //Merge routes A and B into A
12        unifiedRoute ← unifyRoutes(routeA, routeB, savingEdge)
13        currentSol ← deleteRoute(currentSol, routeB)
        //Assign final vehicles to routes
14        for each route in currentSol do
15          route ← assignFinalVehicle(getCandidateVehicle(route))
16        end for
        //Optimize route applying a Local Search
17        unifiedRoute ← improveRouteUsingTwoOpt(unifiedRoute)
18      end if
19    end while
      //Optimize solution applying Local Searches
20    currentSol ← improveSolutionUsingCache(currentSol, costs)
21    if currentSol is a "promising" then
22      currentSol ← improveSolutionUsingSplitting(currentSol, costs, vehicles)
23    end if
      //Keep the best found solution
24    if currentSol outperforms bestHvrpSol then
25      bestHvrpSol ← currentSol
26    end if
27  end while
//Return the best found solution so far
28  return bestHvrpSol;
end procedure
```

Fig. 4. General pseudo-code of the Rand-MER algorithm

```
procedure validateMergeConstraints(edge, routeA, routeB, vehicles, solution)
  if (nodes of edge are internal in routeA or routeB) OR (cost(routeA) + cost(routeB) - saving(edge) > MaxRouteLength) then
    return false
  end if
//Vehicle Assignment: Create a new candidate route joining routes A and B and deleting these two from solution
  candidateSolution ← getRoutesWithNewCandidate(solution, routeA, routeB)
//Sort list of routes in decreasing order of loads
  routeList ← getSortedRouteList(candidateSolution)
  vehicleList ← getSortedVehicleList(vehicles)
//Assign each route to the first bigger-free truck
  for each route in routeList do
    vehicle ← getFirstAvailableVehicle(vehicleList, route)
    if capacity(vehicle) < load(route)
      return false
    end if
    route ← setCandidateVehicle(vehicle)
  end for
  return true
end procedure
```

Fig. 5. Validation for merging of two routes

Numerical experiments

While there are some standard benchmarks easy to replicate for the homogeneous (capacitated) VRP and VRPM (Taillard *et al*, 1996), those for the heterogeneous VRP are more scarce and are adapted to the specific heterogeneous version of the problem they are dealing with, e.g., with or without fixed and/or variable costs associated with the use of each type of vehicle. To complicate things further, some of these instances use probability distributions to generate the spatial coordinates of customers, as well as their associated demands, which make reproducing the experiments difficult. For problems including multiple trips, we were not able to find instances to replicate and compare to. Given these limitations, we use the most applicable test benchmarks we could find in order to measure the performance of our approach:

a) *Prins' instances:* proposed in Prins (2002), these are twenty random instances, denoted as Prins_*i* ($i = 1, 2,..., 20$). Each instance contains 100 customers uniformly distributed in a *200 x 200* km^2 grid. Each customer's demand is uniformly distributed in *[1, 100]*. The depot is placed at the center of the grid, and the maximum time allowed per route is *300* minutes (or *350* km at a speed of *70* km/h). The fleet is composed of $k = 9$ types of vehicles with $m_k = 2$ for all $k = 1, 2,..., 9$. Each type of vehicle has a capacity given by $Q_k = 600 - 50(k - 1)$.
b) *Golden and Taillard's instances*: the first work (Golden *et al*, 1984) proposes 20 instances for the FSMVRP of different sizes, and the second (Taillard, 1999) defines the number of available vehicles of each type. The first 12 instances are quite small –they have less than 50 nodes–, so we have not considered them. Also, instances *13, 16,* and *18* cannot be solved with the MER heuristic since they do not satisfy some of the Prins' assumptions. For our algorithm, we selected eight test instances, denoted as GT_*i*, with $i = 13,\ldots, 20$. The number of customers in these instances, originally proposed by Christofides and Eilon (1969), is between 50 and 100. Information about the fleet composition in these instances is displayed in Table 1.
c) *Li's instances*: five large-scale HVRP instances (Li *et al*, 2007), inspired in Golden *et al*. (1984) and denoted as H*i*, with $i = 1,\ldots, 5$. The number of customers in these instances is between 200 and 360. Each instance has a geometric symmetry, with nodes located in concentric circles around the depot. Information about the composition of the fleets for these instances is displayed in Table 2:

Table 1. Specifications for 8 instances with 6 vehicle types.

Instance	Q_A	m_A	Q_B	m_B	Q_C	m_C	Q_D	m_D	Q_E	m_E	Q_F	m_F	%
GT_13	20	4	30	2	40	4	70	4	120	2	200	1	95.39
GT_14	120	4	160	2	300	1							88.48
GT_15	50	4	100	3	160	2							94.76
GT_16	40	2	80	4	140	3							94.76
GT_17	50	4	120	4	200	2	350	1					95.38
GT_18	20	4	50	4	100	2	150	2	250	1	400	1	95.38
GT_19	100	4	200	3	300	3							76.74
GT_20	60	6	140	4	200	3							95.92

Q_k: capacity of vehicle type k (k = A, B, C, D, E, F); m_k = number of vehicles of type k available; %: 100 x (total demand/total capacity).

Table 2. Specifications for 5 instances with 6 vehicle types.

Instance	Q_A	m_A	Q_B	m_B	Q_C	m_C	Q_D	m_D	Q_E	m_E	Q_F	m_F	%
H1	50	8	100	6	200	4	500	3	1000	1			93.02
H2	50	10	100	5	200	5	500	4	1000	1			96.00
H3	50	10	100	5	200	5	500	4	1000	2			94.76
H4	50	10	100	8	200	5	500	2	1000	2	1500	1	94.12
H5	50	10	100	8	200	5	500	1	1000	2	1500	1	92.31

Q_k: capacity of vehicle type k (k = A, B, C, D, E, F); m_k = number of vehicles of type k available; %: 100 x (total demand/total capacity).

To test these sets of instances, both the MER and Rand-MER algorithms have been implemented as a Java application. These implementations have been executed on a Java Virtual Machine (JVM) version 1.6 using a computer with the following characteristics: a Windows 7 Professional SP1 64 bits operating system, an Intel Xeon E5603 1.60Ghz processor, and 8 GB of RAM.

We run 10 independent executions per instance, using a different seed for the pseudo-random number generator. Each execution was run for 1 minute. Table III shows a comparison, for the three data sets, between the outcomes obtained with the MER heuristic and our algorithm. The first three columns describe the name of the instance, the number of customers visited, and the total demand delivered, respectively. Next, for each algorithm the number of routes in the corresponding solution and the computation time are given. The last column shows the percentage gap between both solutions. Notice that our approach clearly outperforms the MER heuristic (average gap about 4.76%). However, as discussed before, the exact values obtained in Prins (2002) could not be replicated due to the use of random inputs. In small- and medium-size instances, the CPU times of MER are less than 1 second –as expected for a deterministic-greedy algorithm or heuristic. However the time for exploring the solution space with our multi-start approach (Rand-MER), can vary on each instance. For this, we keep the best found solution in a given exploration time period (1 minute). Results in Table III are useful to directly compare our algorithm to the original MER heuristic – which is the only one that solves the HFVRPM. However, unlike Prins and Li instances, instances GT_13, GT_16, and GT_18 do not satisfy the MER assumption in which any vehicle can cover any customer demand. This constraint can be somewhat unrealistic. In fact, in our real application the company serves large-size customers which cannot be covered with the smaller vehicles in its fleet.

We also tested our algorithm in 10 real instances provided by the distribution company. This company has a fleet with 8 different types of vehicles where only one mid-size type can make more than one trip in a day (from 169 total available vehicles, more than 80% is allowed to make a second trip). More details of the real enterprise case considered and its data instances have been explained in Grasas *et al.* (2013). The solution needs multiple trips as the total daily demand is greater than the sum of all vehicle capacities. In addition, some customer demands were higher than the capacity of the smallest vehicle (HFVRPM (B)); therefore we had to adapt the MER heuristic

('Modified MER') to solve the instances and compare the results with those obtained with our approach. As in the previous experiment, the algorithm was executed with 10 different seeds per instance, employing a total computing time of 10 minutes per instance. Table IV presents the corresponding results: (a) the modified version of the MER heuristic where a change has been done for the multi-trips constraint proposed in this paper and (b) the complete version of our Rand-MER algorithm, including the multi-start biased randomization process. For this last version, both the average cost of the 10 runs per instance, as well as the best-found solution for each instance are provided.

Table 3. Results on 3 datasets after 10 minutes of execution per instance (single-trip case).

			MER			*Rand-MER*					
Instance	*Customers*	*Total Demand Delivered*	*Cost (1)*	*Routes*	*Time (sec)*	*Best Cost (2)*	*Routes*	*Time (sec)*	*Gap (2-1)*	*Average 10 Seeds (3)*	*Gap (3-1)*
Prins_1	100	4806	2537.62	10	0.09	2457.29	10	8.00	-3.17%	2482.20	-2.18%
Prins_2	100	4895	2731.76	11	0.14	2591.28	10	7.79	-5.14%	2604.80	-4.65%
Prins_3	100	5218	2712.66	11	0.07	2672.80	11	6.23	-1.47%	2691.01	-0.80%
Prins_4	100	4763	2398.98	10	0.08	2316.77	10	25.55	-3.43%	2337.92	-2.55%
Prins_5	100	5069	2750.92	11	0.06	2586.02	11	5.59	-5.99%	2611.97	-5.05%
Prins_6	100	4896	2711.17	10	0.09	2613.88	10	7.13	-3.59%	2619.71	-3.37%
Prins_7	100	5268	2844.43	12	0.07	2705.84	11	12.44	-4.87%	2726.04	-4.16%
Prins_8	100	5310	2812.29	12	0.05	2680.22	12	30.69	-4.70%	2709.45	-3.66%
Prins_9	100	5403	2757.86	12	0.06	2600.38	12	5.83	-5.71%	2623.62	-4.87%
Prins_10	100	4462	2426.23	10	0.09	2365.94	9	7.28	-2.48%	2379.53	-1.92%
Prins_11	100	5269	2832.11	12	0.06	2653.04	12	6.24	-6.32%	2695.42	-4.83%
Prins_12	100	4860	2645.14	10	0.07	2510.77	10	5.69	-5.08%	2548.09	-3.67%
Prins_13	100	4772	2626.43	10	0.09	2454.11	10	8.10	-6.56%	2464.20	-6.18%
Prins_14	100	5065	2658.18	11	0.08	2573.93	11	47.16	-3.17%	2584.44	-2.77%
Prins_15	100	4716	2768.28	10	0.09	2597.41	10	8.40	-6.17%	2617.78	-5.44%
Prins_16	100	5504	3013.82	13	0.05	2813.44	12	5.69	-6.65%	2839.76	-5.78%
Prins_17	100	4632	2563.45	10	0.08	2381.50	10	38.03	-7.10%	2401.90	-6.30%
Prins_18	100	4727	2517.39	11	0.08	2411.38	10	49.10	-4.21%	2427.30	-3.58%
Prins_19	100	5398	3003.30	12	0.06	2857.48	12	39.16	-4.86%	2874.62	-4.28%
Prins_20	100	4730	2539.31	10	0.10	2415.98	10	14.48	-4.86%	2452.30	-3.43%
		Average	*2692.57*	*10.9*	*0.08*	*2562.97*	*10.65*	*16.93*	*-4.78%*	*2584.60*	*-3.97%*
GT_13	50	973	NA	NA	NA	821.34	17	37.64	NA	844.93	NA
GT_14	50	973	569.76	7	0.01	539.01	6	39.12	-5.40%	540.90	-5.07%
GT_15	50	777	677.57	9	0.01	633.79	9	9.24	-6.46%	634.17	-6.40%
GT_16	50	777	NA	NA	NA	637.94	9	3.48	NA	638.47	NA
GT_17	75	1364	796.14	11	0.04	770.54	10	5.11	-3.22%	775.66	-2.57%
GT_18	75	1364	NA	NA	NA	787.57	12	19.87	NA	788.53	NA
GT_19	100	1458	853.15	6	0.17	760.24	6	73.28	-10.89%	773.94	-9.29%
GT_20	100	1458	1032.78	13	0.08	984.80	13	8.26	-4.65%	998.00	-3.37%
		Average	*767.73*	*9.17*	*0.05*	*741.90*	*10.25*	*24.50*	*-6.12%*	*749.33*	*-5.34%*
H1	200	4000	9188.10	18	3.16	8459.35	18	32.25	-7.93%	8637.68	-5.99%
H2	240	4800	6535.64	25	8.27	6457.83	24	29.89	-1.19%	6480.85	-0.84%
H3	280	5600	11121.00	20	19.03	10990.18	19	76.63	-1.18%	11114.85	-0.06%
H4	320	6400	9480.08	23	53.96	9331.28	21	59.67	-1.57%	9413.13	-0.71%
H5	360	7200	12502.36	18	208.65	11918.75	18	143.71	-4.67%	12360.08	-1.14%
		Average	*9765.44*	*20.80*	*58.61*	*9431.48*	*20.00*	*68.43*	*-3.31%*	*9601.32*	*-1.75%*

Solutions are represented by the number of routes and the total distance-based cost. For each instance, the number of stores visited and the total demand delivered are also displayed. Notice that more routes does not necessarily imply higher costs. Multi-trips appear in almost all solutions (except for D) since the number of routes exceeds the total number of vehicles. Our algorithm also outperforms the 'Modified MER' in these real instances. Both algorithms improve the solutions provided by the company. Notice that the 'Modified MER' does not include a biased randomization of the construction phase. This remains greedy as the original algorithm, and its average execution time is about 2.6 seconds.

Table 4. Results for real multi-trip instances after 10 minutes of execution per instance.

			Modified MER (*)			*Rand-MER*					
Instance	*Stores Visited*	*Total Demand Delivered*	*Cost (1)*	*Routes*	*Time (sec)*	*Best Cost (2)*	*Routes*	*Time (sec)*	*Gap (2-1)*	*Average 10 Seeds (3)*	*Gap (3-1)*
A	372	77913	40001.86	182	3.02	39534.11	180	63.22	-1.17%	39841.99	-0.40%
B	366	79130	41821.23	187	2.41	41072.46	183	55.20	-1.79%	41399.65	-1.01%
C	371	91901	50337.90	221	2.90	49669.31	219	66.48	-1.33%	50082.32	-0.51%
D	364	63078	31697.62	136	1.73	31378.63	135	58.94	-1.01%	31543.09	-0.49%
E	372	83571	46372.77	202	2.70	45485.83	200	29.73	-1.91%	45836.63	-1.16%
F	373	85773	46327.54	211	2.76	45275.62	206	7.67	-2.27%	45681.39	-1.39%
G	372	84023	45939.70	199	2.60	45165.12	197	28.53	-1.69%	45493.28	-0.97%
H	374	85539	45070.87	202	2.76	44386.64	200	65.94	-1.52%	44909.39	-0.36%
I	370	89596	49613.35	214	2.88	49053.97	212	59.57	-1.13%	49354.83	-0.52%
J	372	76846	39712.54	177	2.25	38973.19	175	29.33	-1.86%	39252.86	-1.16%
Average			*43689.54*	*193.10*	*2.60*	*42999.49*	*190.70*	*46.46*	*-1.57%*	*43339.54*	*-0.80%*

(*) Including the proposed vehicle assignment with multi-trip and using whole fleet described in section 3.

Concluding remarks

In this paper, we discuss the importance of considering realistic constraints in Vehicle Routing Problems. In particular, we focus on those that affect the fleet of vehicles available for the routing and the possibility of vehicles making multiple trips. Our work contributes to the research on VRP on three levels: (i) we describe a new version of the classical VRP, the Heterogeneous Fleet VRP with Multi-trips; (ii) we provide a classification of vehicle routing problems based on the constraints we relax (heterogeneous fleet and multi-trips); and (iii) we propose a new algorithm to solve the HFVRPM. We also present a computational experiment comparing the proposed algorithm to other approaches designed for similar problems, and describe the results obtained in a real application.

Our algorithm is based on a randomized version of the MER heuristic (Rand-MER), which also benefits from some local search processes initially developed for the homogeneous VRP. One of its advantages is that it can be easily implemented in practice, since it is relatively simple and it does not require any complex fine-tuning process. The use of known biased-probabilistic distributions adds none or few parameters to the algorithm. In our case, only one parameter is included and its value has been successfully used before in Juan *et al*. (2011b). Additionally the use of the family of biased distributions and their parameters are quite known in the mathematical and simulation literature. Therefore, it can be employed in real-life scenarios with little effort. We have tested our approach with some classical benchmarks as well as with real data from a distribution company in Spain. The experimental results seem to validate the competitiveness of the proposed algorithm, since it is able to improve both other academic approaches as well as real routing plans developed by experts in the company. Finally notice that the biased randomization can be easily applied to a wide range of classical heuristics which are oriented to other optimization problems.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Science and Innovation (ECO2009-11307, TRA2010-21644-C03) and the CYTED-HAROSA Network (CYTED2010-511RT0419), in the context of the IN3-ICSO program (http://dpcs.uoc.edu) and the Business Analytics Research Group (http://www.upf.edu/barg/). We acknowledge the personnel of the distribution company for their cooperation in this project.

References

Baldacci R, Battarra M, and Vigo D. (2008). Routing a heterogeneous fleet of vehicles. In: Golden B, Raghavan S and Wasil EE (eds). The Vehicle Routing Problem: Latest Advances and New Challenges. Springer: New York, pp 3-27.

Brandao J (2011). A Tabu Search Algorithm for the heterogeneous fixed fleet vehicle routing problem. Computers and Operations Research 38(1): 140-151.

Brandao J and Mercer A (1998). The multi-trip vehicle routing problem. Journal of the Operational Research Society, 49(8): 799-805.

Clarke G and Wright J (1964). Scheduling of vehicles from a central depot to a number of delivering points. Operations Research, 12(4): 568-581.

Cordeau J F, Gendreau M, Laporte G, Potvin J Y, and Semet F (2002). A guide to vehicle routing heuristics. Journal of the Operational Research Society, 53: 512-522.

Croes G (1958). A method for solving traveling salesman problems. Operations Research, 6: 791-812.

Chao I M, Golden B, and Wasil E (1999). A computational study of a new heuristic for the site-dependent vehicle routing problem. Information Systems and Operational Research (INFOR), 37(3): 319-336.

Christofides N and Eilon S (1969). An algorithm for the vehicle dispatching problem. Journal of the Operational Research Society, 20(3): 309-318.

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

Euchi J and Chabchoub H (2010). A Hybrid Tabu search to solve the heterogeneous fixed fleet vehicle routing problem. Logistics Research 2(1): 3-11.

Festa P, Resende MGC (2009). An annotated bibliography of GRASP—Part I: Algorithms. Int Trans Oper Res 16:1-24.

Fleischmann M (1990). The vehicle routing problem with multiple use of vehicles: Working paper, Universität Hamburg.

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

Golden B, Assad A, Levy L, and Gheysens F (1984). The fleet Size and Mix Vehicle-Routing Problem. Computers and Operations Research, 11(1): 49-66.

Golden B, Raghavan S, and Wasil E E (eds) (2008). The Vehicle Routing Problem: Latest Advances and New Challenges. Springer: New York.

Grasas A, Cáceres-Cruz J, Lourenço H, Juan A, Roca M (2013). Vehicle routing in a Spanish distribution company: Saving using a savings-based heuristic. OR Insight, 26(3) 191-202.

Juan A A, Caceres-Cruz J, Gonzalez S, Riera D, Barrios B (2014). Biased Randomization of Classical Heuristics. In: Wang J (ed), Encyclopedia of Business Analytics and Optimization. IGI Global.

Juan A A, Faulin J, Jorba J, Caceres J, and Marques J (2011a). Using Parallel and Distributed Computing for Solving Real-time Vehicle Routing Problems with Stochastic Demands. Annals of Operations Research

Juan A A, Faulin J, Jorba J, Riera D, Masip D, and Barrios B (2011b). On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics. Journal of the Operational Research Society, 62(6): 1085-1097.

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

Li F, Golden B, and 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.

Lin C K Y and Kwok R C W (2006). Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. European Journal of Operational Research, 175(3): 1833-1849.

Mingozzi A, Roberti R, and Toth P (2013). An Exact Algorithm for the Multitrip Vehicle Routing Problem. INFORMS Journal on Computing, 25(2): 193–207.

Olivera A and Viera O (2007). Adaptive memory programming for the vehicle routing problem with multiple trips. Computers & Operations Research, 34: 28-47.

Osman I H and Salhi S. (1996). Local search strategies for the mix fleet routing problem. In: Rayward-Smith V J, Osman I H, Reeves C R and Smith G D (eds). Modern Heuristic Search Methods. Wiley: Chichester, pp 131-153.

Petch R J and Salhi S (2003). A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. Discrete Applied Mathematics, 133(1-3): 69-92.

Poot A, Kant G, and Wagelmans A P M (2002). A savings based method for real-life vehicle routing problems. Journal of the Operational Research Society, 53: 57-68.

Prins C (2002). Efficient heuristics for the Heterogeneous Fleet Multitrip VRP with Applications to a Large-Scale Real Case. Journal of Mathematical Modelling and Algorithms, 1(2): 135-150.

Prins C (2009). Two memetic algorithms for heterogeneous fleet vehicle routing problems. Engineering Applications of Artificial Intelligence, 22(6): 916-928.

Sen A and Bulbul K (2008). A Survey on Multi Trip Vehicle Routing Problem, VI International Logistics and Supply Chain Congress. Istambul, Turkey.

Taillard E D (1999). A heuristic column generation method for the heterogeneous fleet VRP. Rairo-Recherche Operationnelle-Operations Research, 33(1): 1-14.

Taillard E D, Laporte G, and Gendreau M (1996). Vehicle routing with multiple use of vehicles. Journal of the Operational Research Society, 47(8): 1065-1070.

Tarantilis C D, Kiranoudis C T, and Vassiliadis V S (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research, 152: 148-158.

Toth P and Vigo D (eds) (2002). The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM Publishing: Philadelphia, PA.