To cite this article:

Molina, J. C., Salmeron, J. L., & Eguia, I. (2020). An ACS-based memetic algorithm for the heterogeneous vehicle routing problem with time windows. *Expert Systems with Applications*, *157*, 113379.

An ACS-based Memetic algorithm for the Heterogeneous Vehicle Routing Problem with time windows

Jose C. Molina*[a], Jose L. Salmeron[b,c], Ignacio Eguia[a]

Abstract

This paper presents a solution methodology to solve the heterogeneous vehicle routing problem with time windows (HVRPTW). This problem appears when a limited fleet of vehicles, characterized by different capacities, fixed costs and variable costs, is available for serving a set of customers which have to be visited within a predefined time window. The objective is to perform the route design minimizing the total fixed vehicle costs and distribution costs and satisfying all problem constraints. The problem is solved using an ant colony system (ACS) algorithm which has been successfully applied to combinatorial optimization problems. Moreover, to improve the performance of the ACS on the HVRPTW, a hybridized ACS with local search, called memetic ACS algorithm is proposed where the local search is performed by a variable neighborhood tabu search algorithm. Experiments are conducted on sets of benchmark instances from the scientific literature to evaluate the performance of the proposed algorithm. The results show that the algorithm has a good performance on the HVRPTW. In particular, out of the 80 instances, it obtained 65 new best solutions and matched 6 within reasonable computational times.

Keywords: Heterogeneous VRPTW; Ant Colony System; Memetic algorithms;VNTS

1. Introduction

Transportation plays an important role in product and services distribution organizations. This logistic activity is responsible for moving the products from a place of origin to a point of destination, both at the beginning of the activities of a supply chain until the final delivery to the customer. Therefore, route management has become a fundamental aspect in companies whose activity is mainly based on transport. An effective management has a significant impact on service quality, and it can also increase efficiency and productivity, minimizing different types of costs such as costs of fuel, driver costs, fixed vehicle costs, etc.

The vehicle routing problem (VRP) was first proposed by Dantzig & Ramser (1959) and is a well-known combinatorial optimization problem which tries to minimize the total distance traveled by a set of homogeneous vehicles while satisfying the demand of a given set of customers. It considers the assumptions that each vehicle performs a single route without exceeding its capacity constraint, routes start and end at a depot and every customer must be served by exactly one vehicle. The diversity of the VRP is very broad since there are many variants, with each of them focused on solving a specific problem. As a consequence, numerous studies involving the VRP and its variants have been published in the scientific literature over the last few years. For further information, see e.g., the surveys of Cordeau et al. (2007), Irnich et al. (2014) and the book by Toth & Vigo (2014).

The fleet of vehicles in a company is usually heterogeneous in real-life as the company incorporates vehicles of different characteristics over time (Hoff et al., 2010). Moreover, the use of heterogeneous fleet of vehicles has multiple advantages. Having vehicles of different capacities results in a better adaptation to the customers´ varying demand and allows to perform the delivery in a more cost-effective way (Tarantilis et al, 2004). In addition, restrictions on the vehicle size and weight may be required to access urban areas (Semet, 1995). The Heterogeneous Fleet VRP (HF-VRP) is a variant of the VRP where a heterogeneous fleet of vehicles is used for the distribution activities (Baldacci et al., 2008). Two variants of the HF-VRP with respect to the available fleet have been addressed in the literature. The HF-VRP with a limited number of vehicles, called the Heterogeneous VRP (HVRP) (Taillard, 1999) involves optimizing the vehicle routes with the available fixed fleet. On the other hand, the Fleet Size and Mix VRP (FSMVRP) (Golden et al., 1984), consists of determining the best fleet composition and routing when the number of vehicles of each type is unlimited. Therefore, the idea is not only to consider the routing of the vehicles, but also the fleet composition. Relating to the type of costs to be minimized, two different objective functions have been considered. They consist of minimizing the vehicle acquiring costs (fixed costs) and/or the dependent routing costs. The latter can be related to the total trip duration without service times or proportional to the total distance traveled. The HF-VRP with time windows (TW) appears when additional restrictions are introduced in the problem to force customers to be served by a vehicle in a predefined time interval $[E_i , L_i]$, where E_i and L_i are the earliest and latest times to start the service respectively. Thus, the HF-VRP variants with time windows are denoted by adding TW to the acronym of the specific problem. For a recent literature survey on the HF-VRP we refer to Koç et al. (2016).

This paper tackles the HVRP with TW (HVRPTW) which was first addressed and solved by Paraskevopoulos et al. (2008). The objective is to find the optimum set of routes that minimize the sum of the total fixed and dependent routing costs. Mathematical formulations of this problem are provided by Paraskevopoulos et al. (2008) and by Jiang et al. (2014). The HVRPTW is an NP-Hard problem, which means that there is no algorithm that solves it in a polynomial time and justifies the use of heuristics and metaheuristics approaches to solve the problem.

The contribution of this paper is to propose a solution approach consisting of a hybridized ant colony system (ACS) with local search, called memetic ACS (Mem-ACS) to solve the HVRPTW. The Mem-ACS seeks to increase the quality of the solutions by the use of a Variable Neighborhood Tabu Search (VNTS) for exploiting the local search. This approach combines two state-of-the-art metaheuristic algorithms (ACS and VNTS) which have been successfully applied separately on a large variety of VRPs. Furthermore, to the best knowledge of the authors, it does not exist in the scientific literature any VRP application which combines both algorithms. In addition, in order to handle both feasible and infeasible solutions, a holding list (HL) containing the unserved customers, is introduced in the algorithm. This idea was first introduced by Lau et al. (2003) and subsequently implemented by other authors such as Jiang et al. (2014), Lim and Zhang (2007) or Molina et al. (2016).

The paper is structured as follows. A short review of the HF-VRPTW literature is presented in Section 2. Section 3 presents a detailed description of the main components of the Mem-ACS algorithm. The parameter settings and results obtained from benchmark instances from the scientific literature are reported in Section 4 and finally, the conclusions and recommendations for further work are given in Section 5.

2. Related works

Regarding the literature of FSMVRPTW, a variety of solution approaches have been proposed in the scientific literature since its introduction by Liu & Shen (1999), who proposed several insertion-based savings heuristics for solving the problem. They tested the heuristics introducing a set of heterogeneous fleets with an unlimited number of vehicles for the set of 168 benchmark instances derived from the Solomon (1987) instances. Later, Dullaert et al. (2002) developed three insertion-based heuristics incorporating the modified savings calculations based on Golden et al. (1984). The authors tested their algorithms using different objective functions. Dell'Amico et al. (2007) developed a constructive insertion heuristic and a metaheuristic algorithm for solving the problem. The heuristic is described as an insertion-based parallel approach while the metaheuristic adopted the ruin and recreate paradigm for the improvement of the solutions. Belfiore & Fávero (2007) proposed a scatter search approach that was applied to solve the benchmark instances proposed by Liu & Shen (1999). The results of the algorithm were compared with the heuristics proposed by Liu & Shen (1999) and Dullaert et al. (2002) presenting better results with respect to the quality of the solutions. Bräysy et al. (2008) proposed a three-phase multi-restart deterministic annealing metaheuristic for solving the FSMVRPTW. They also tested the algorithm on the 168 benchmark instances introduced by Liu & Shen (1999) obtaining 151 new best-known solutions. One year later, Bräysy et al. (2009) presented a new hybrid metaheuristic that combines threshold-accepting and guided local search for solving large-scale FSMVRPTW up to 1000 customers. The results showed that the method is competitive with the previous approach. Repoussis & Tarantilis (2010) presented a solution approach that utilizes the concept of an adaptive memory programming solution framework equipped with a semi-parallel construction heuristic, a re-construction mechanism, an iterated tabu search (TS) algorithm and frequency based long term memory structures. The algorithm improved the results over most FSMVRPTW instances illustrating the efficiency and effectiveness of the proposed method. Vidal et al. (2014) proposed a new component-based heuristic framework and a unified hybrid genetic search to address a large variety of VRP variants in which FSMVRPTW is included. They also tested the algorithm on FSMVRPTW instances outperforming the mean results obtained by Repoussis & Tarantilis (2010) and Bräysy et al. (2008) .

According to the literature survey on the HF-VRP, HVRP has received less attention than the FSMVRP due to the significant difference in the number of published papers. Note that it is easy to find feasible solutions to the FSMVRP since there is always a feasible insertion of customers to vehicles as they are unlimited. However, HVRP becomes more complex as all customers must be assigned to a vehicle route satisfying all problem constraints. Moreover, the incorporation of TW restrictions to the FSMVRP or to the HVRP increases the difficulty of obtaining feasible solutions.

Paraskevopoulos et al. (2008) introduced the HVRPTW and proposed a two-phase solution framework based on a semi-parallel construction heuristic and a VNTS algorithm. The algorithm solved both HVRPTW and FSMVRPTW. Later, Koç et al. (2015) proposed a hybrid evolutionary algorithm to also solve both problems. The algorithm combines advanced procedures and several metaheuristics principles from Vidal et al. (2014), such as adaptive large scale neighborhood search and population search. They obtained some new solutions on benchmark instances outperforming all previous algorithms. Jiang et al. (2014) defined the VRP with heterogeneous fleet and TWs to generalize the variants existing in the scientific literature and developed a two-phase TS algorithm. They considered a hierarchical objective function where the total number of served customers is considered as the primary objective, minimizing the total number of vehicles and total distance traveled as secondary and tertiary respectively.

The previous researches show that Evolutionary Algorithms and VNTS are appropriate methods to solve HVRPTW. Memetic algorithms (MAs) are defined in the scientific literature as a population based methods which are hybridized with a local search procedure to intensify the search. They have been successfully applied for different combinatorial optimization problems. However, in the scientific literature, almost all existing memetic algorithms for solving different VRP variants follow the framework of genetic algorithms and very rarely the framework of ACS algorithms. [e.g. Tavakkoli-Moghaddam et al. (2006); Prins (2009); Cattaruzza et al. (2014); El Fallahi et al. (2008); Mendoza et al. (2010)]. To the best of our knowledge there is little research in this area. Yu et al. (2009) proposed an improved ant colony optimization (ACO), with a new strategy to update the increased pheromone and a mutation operation, to solve VRP. Mavrovouniotis & Yang (2011) used an adaptive local search operator inside an ACO algorithm with an enhanced population-list to solve the dynamic travelling salesman problem. Abdulkader et al. (2015) proposed a hybridized algorithm which combines local search with an existent ant colony algorithm to solve the multi-compartment VRP. They tested the effectiveness of the algorithm on new benchmark problems.

3. Methodological approach

Memetic algorithms (MAs) were first presented in Moscato (1989) and constitute a class of metaheuristics that combine population-based global search approaches with local search procedures (Bansal et al., 2013; Salmeron et al., 2017). In the framework of MAs, a global search component is responsible for performing the exploration of the search space and local search techniques are used for further exploration (exploitation) of the regions found to be promising, which is helpful for attaining more accurate solutions (Acampora et al., 2011; Salmeron et al., 2019).

Examples of population based algorithms include evolutionary algorithms, such as genetic algorithms (Goldberg, 1989), evolution strategy (Beyer & Schwefel, 2002) and differential evolution (Price et al., 2006), but also ACO (Dorigo & Birattari, 2010), particle swarm optimization (Kennedy, 2006) and artificial bee colony (Karaboga, 2005) among others.

The rapidly growing research interest in MAs is demonstrated by the significant increase in the number of research publications on MAs. Although a large portion of MAs are evolutionary algorithms combined with local search techniques, there are numerous MAs in the literature that are derived from other metaheuristics such as ACO [e.g. Sharifipour et al. (2018); Mavrovouniotis & Yang (2011); Fathi et al. (2014)], ACS [e.g. Chen et al. (2013)], particle swarm optimization [e.g. Wang et al. (2012); Zhang et al. (2016)], artificial immune systems [e.g. Yang et al (2008)], artificial bee colony [e.g. Bansal et al. (2013)] and so on.

In this section, we describe our proposed solution approach consisting of a hybridized ACS with local search, called Mem-ACS, to solve the HVRPTW. This decision is based on the remarkable success of swarm intelligent algorithms, such as ACO and ACS, in combinatorial optimization problems (Mavrovouniotis & Yang 2011; Abdulkader et al. 2015). Moreover, the combination of a probabilistic, adaptive construction heuristic by artificial ants with local search algorithms may result in solutions of higher quality. ACO and ACS algorithms are such adaptive construction heuristics, in the sense that a colony of ants modifies the solution representation assigning higher pheromone trail strength to connections which are contained in better solutions. During the solution construction ants preferably select couplings which have high pheromone strength and by combining such couplings they generate promising starting solutions for the local search algorithm. An additional advantage of using ant algorithms is that, by generating good initial solutions, the subsequent local search needs less iteration to reach a local optimum. Thus, for a given time limit many more local searches can be run than by starting from randomly generated solutions (Stützle & Dorigo 1999). Thus, the flexibility of building solutions in ACS algorithms allows its application to VRPs especially when a limited heterogeneous vehicle fleet and TW restrictions are considered. In contrast, genetic algorithms usually may lead to obtain unfeasible solutions (due to capacity or TW restrictions) and therefore, additional techniques have to be integrated within the algorithm.

In addition, according to Dorigo & Birattari (2010) ACO and ACS algorithms are very interesting approaches to solve problems that can be translated into seeking the minimum cost of a graph structure. Moreover, they are algorithms where local searches are very easy to incorporate, since they intrinsically consider the use of local heuristics.

On the other hand, the local search aspect is performed by a VNTS which combines a VNS scheme with the use of TS to indicate promising regions not yet explored. Some authors [e.g. Repoussis et al., 2006; Paraskevopoulos et al. (2008); Molina et al. 2016] have shown that VNTS provides excellent results when applied to different HF-VRP variants.

The remainder of this section introduces the main components of the Mem-ACS algorithm. Section 3.1 presents a general description of the ACS approach. Section 3.2 describes the VNTS algorithm used in the local search process and an illustrative example is presented in section 3.3 to help make the methodology clearer.

3.1 Ant Colony System

The Ant Colony System (ACS) algorithm was first presented by Dorigo & Gambardella (1997) and is a probabilistic metaheuristic inspired by the behavior of ant colonies for solving combinatorial optimization problems. ACS defines a maximum number of iterations (*iter_max*) and a population of ants (M), which participates in the two main phases of the algorithm; the ant´s route construction and the pheromone update.

3.1.1 Route construction

In order to tackle the HVRPTW, an individual ant simulates a fleet of vehicles, whose routes are built sequentially one by one. Two lists containing the heterogeneous vehicles (V_k, $k=1...K$) and the unassigned customers (C_s, $s=1...N$) are needed to be implemented in the algorithm for every route construction. The HL is considered as the vehicle $K+1$.

An ant randomly chooses a vehicle from V_k and initializes the route by a "seed" criterion based on the furthest customer from Cs. Then, at each construction step, an un-routed customer is chosen to be inserted in the current route according to the following pseudo-random proportional rule. If a random number q that is uniformly distributed over [0,1] is less than q_0, equation (1) is used to choose the next customer from Cs, where P_{ijuk} represents the probability to choose a customer u to be inserted between customers i and j in the current route k. Otherwise, the customer is selected by a fitness proportionate selection, also known as roulette wheel selection, according to the probability distribution given in equation 1. Therefore, with probability q_0, the best move described is performed, while with probability (1-q_0) different arcs are explored. Thus, customers are iteratively inserted at any position on the route until none can be inserted due to capacity or TW restrictions. Then, the route is added to the partially final solution and the vehicle and served customers are removed from V_k and Cs respectively. Next, the ant randomly chooses another vehicle from V_k and the overall procedure is repeated until all customers are assigned to a vehicle route. Finally, the customers who were not visited by a vehicle are stored in a HL. The HL is an infeasible customer list, similar to a ‘‘phantom’’ route, that plays an important role, as it will participate in the local search process inducing an extended neighborhood search space for every inter-route operator of the VNTS (Lau et al., 2003).

$$P_{ijuk} = \begin{cases} \dfrac{(\tau_{iu})^{\alpha} \cdot (\eta_{ijuk})^{\beta}}{\sum_{w \in J(i)} (\tau_{iw})^{\alpha} \cdot (\eta_{ijwk})^{\beta}}, & if\ u \in J(i) \\ 0, & otherwise \end{cases} \tag{1}$$

P_{ijuk} takes into account two different values; the level of pheromone on the arc (i,u), denoted as τ_{iu} and η_{ijuk} that represents the attractiveness of the move. $J(i)$ represents the feasible neighborhood for customer i, which is formed by the set of un-routed customers from C_s that are directly accessible from node i without violating capacity and TWs restrictions.

For the HVRPTW, the attractiveness of the move (η_{ijuk}) is given by a greedy function (Φ_{ijuk}) that measures the cost of inserting a customer u between other two customers i and j served by vehicle k. Our approach used the greedy function proposed by Paraskevopoulos et al. (2008) to calculate the η_{ijuk} as follow:

$$\eta_{ijuk} = \Phi_{ijuk} = \alpha_1 \cdot \left(C^0_{iju} + C^1_{iju}\right) + \alpha_2 \cdot C^1_{iju} + \alpha_3 \cdot C^3_{iju} + \alpha_4 \cdot C^4_{iju} \tag{2}$$

The first metric (Eq. 3a) is a measure of the coverage of the TW for the selected customer u and represents the closeness between the vehicle arrival time at customer u (a_u) and its earliest service time (e_u). Furthermore, the compatibility of the customer time window in the specific insertion position is introduced in the second metric (Eq. 3b). It represents the time gap between the latest service time (l_u) and the time of the vehicle arrival at customer u, which depends on the service time in customer i (s_i) and the travel time to customer u (t_{iu}). When a customer u is inserted between two consecutive customers (i,j) in a route, a driving time increase is produced and is given by the third metric (Eq. 3c). Metric $C_{ij,u}^3$ (Eq. 3d) considers the time difference between the vehicle arrival time at customer j , before and after the insertion of customer u into the current route. This metric represents the time gap that has to be pushed forward in customer j to insert u (Ioannou et al., 2001). Metric $C_{ij,u}^4$ (Eq. 3e) gives priority to the insertion of customers with large demands (D_u) on the route and maximizes the utilization of the vehicle capacity (Q_k) (Paraskevopoulos et al. ,2008). Finally, the α weights in Eq. (2) define the relative contribution of each individual metric to the overall selection (α_1+ α_2 + α_3+ α_4=1)

$$C^0_{iju} = a_u - e_u \tag{3a}$$
$$C^1_{iju} = l_u - (a_i + s_i + t_{iu}) \tag{3b}$$
$$C^2_{iju} = t_{iu} + t_{uj} - t_{ij} \tag{3c}$$
$$C^3_{iju} = \left[l_j - \left(a_i + s_i + t_{ij}\right)\right] - \left[l_j - \left(a_u + s_u + t_{uj}\right)\right] \tag{3d}$$
$$C^4_{iju} = Q_k - \left(\sum_{i \in N \cap u} D_i \cdot \sum_{j \in N} x_{ijk}\right) - D_u \tag{3e}$$

3.1.2 Pheromone initialization and update

At the beginning of the execution of the Mem-ACS, following the idea of the works proposed by Belmecheri et al. (2009) and Ellabib et al. (2007), the initial value of pheromones is obtained applying a heuristic approach. This work used the semi-parallel construction heuristic proposed by Paraskevopoulos et al. (2008) for solving the HVRPTW. The initial pheromone value, defined as τ_0, is shown in Eq. 4, where N is the total number of customers to be served and L is the total distance travelled by the vehicles in the solution obtained by the semi-parallel construction heuristic.

$$\tau_0 = \frac{1}{N*L} \tag{4}$$

The ACS method uses two types of pheromone updates: local and global. Each time a solution is constructed by an ant, the local update is performed by modifying the pheromone level of the arcs (i,j) of the obtained solution (S) as shown in Eq. 5a, where the parameter ρ is introduced to regulate the reduction of pheromone on the arcs. On the other hand, once all ants have computed their route (i.e. at the end of each iteration), the global update is only performed by the ant, that produced the best solution ($S*$) so far. The trail pheromone of the arcs (i,j) are updated as shown in Eq. 5b, where $\Delta\tau_{ij}^{bs} = (N*L_{best})^{-1}$.

$$\tau_{ij} = (1-\rho)\cdot\ \tau_{ij} + \rho\cdot\tau_0 \quad if\ (i,j) \in S \tag{5a}$$
$$\tau_{ij} = (1-\rho)\cdot\ \tau_{ij} + \rho\cdot\Delta\tau_{ij}^{bs} \quad \forall(i,j) \in S^* \tag{5b}$$

3.2 VNTS approach for the local search

VNTS is a hybrid approach that introduces the use of TS in the local search procedure of a Variable Neighborhood Search (VNS) scheme in order to explore the solution space in a more effective manner. VNS was introduced by Mladenovic and Hansen (1997) and is a technique that applies a systematic change of neighborhoods within a local search.

The VNTS starts by defining a set of neighborhood structures N_λ (λ = 1… λ_{max}), where N_λ is the λ_{th} neighborhood. The iterative process starts from an initial solution s obtained by an ant in the ACS algorithm. Then, TS is performed to determine a new solution s´ in N_λ. If $f(s')$ is better than the best solution $f(s)$, then s is replaced by s', and the search returns to N_1, otherwise the search explores the next neighborhood $N_{\lambda+1}$. This is repeated until all neighborhood structures are examined ($\lambda = \lambda_{max}$). Thus, distant neighborhoods of a solution are explored following a discontinuous trajectory during the search jumping from one solution to another if an improvement occurs. The VNTS finishes when a local minimum solution with respect to all neighborhood structures is reached (Hansen et al., 2010).

The combination of VNS with TS results in a high efficiency of the search, which is relevant to find high quality solutions in a HVRPTW. First, the VNS scheme avoids getting stuck in poor quality solutions since a local optimum for a given neighborhood structure is not necessarily so for another. Moreover, the TS provides an intensification of the local search allowing non improving moves and preventing the search from returning to recently visited solutions.

The TS algorithm implemented in this work has been adopted by some authors in the literature. It provides a balance between diversification and intensification search strategies. The use of a small tabu list size results in a more effective intensification search since it allows cycling of small periods. On the other hand, a large tabu list size will allow a diversified search of more distant neighbors, which may result in escaping from a current local optimum solution (Paraskevopoulos et al., 2008).

Figure 1 shows the TS mechanism. Initially, the size of the tabu list (TT) is set equal to a lower value (TT_{min}). The diversification scheme is provided by increasing at each iteration the TT in one unit up to an upper bound TT_{max} while no improvement is observed. On the other hand, intensification of the search is performed when a new best solution is found. For this purpose, the TT is reinitialized to TT_{min} removing the oldest solutions from the tabu list. For a detailed description, we refer to the papers presented by Paraskevopoulos et al. (2008) and Molina et al. (2009).

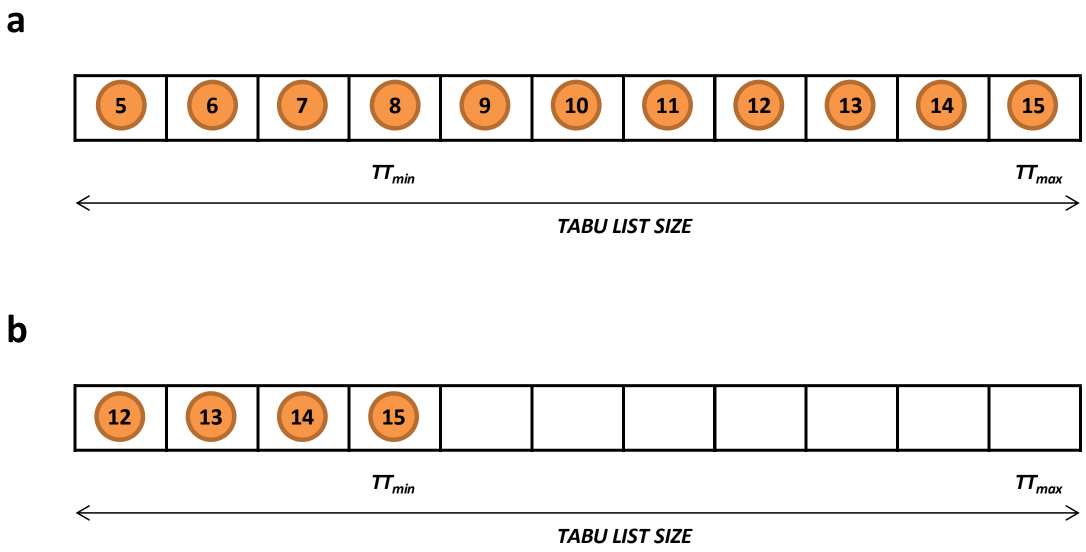

Figure 1 (a) A tabu list scheme, with TT_{min} and TT_{max} values, while no improvement of the current best solution is observed. (b) a tabu list scheme just when a new best solution is found.

This research uses seven popular and effective neighborhood structures (λ_{max}=7) which have been used in VRPs. The set of neighborhoods are used in the following sequence: Relocate (inter-route), Exchange (inter-route), GENI, 2-Opt, Relocate (intra-route), Exchange (intra-route), Double insertion, CROSS-exchange and GENI-Exchange.

- Relocate: this operator removes a customer from a current position and inserts it into a new position. It is applied on pairs of routes (inter-route) and on single routes (intra-route).
- Exchange: this operator simultaneously exchanges two customers. It is applied on pairs of routes (inter-route) and on single routes (intra-route).
- GENI-Insertion: this neighborhood is applied only on pairs of routes and is only composed of the Generalized Insertion (GENI) operator. It basically consists of removing a customer from a route and inserting it into any two customers, not necessarily consecutive, of another route. For further information, we refer to Gendreau et al. (1998).
- GENI-Exchange: this operator simultaneously removes two customers from different routes and exchanges them but performs the insertions through a GENI operator in the best feasible position of the route.
- 2-Opt: this operator is applied only on single routes (intra-route). It removes two non-adjacent edges from a solution and reconnects the remaining parts constructing two new arcs in a route. This operation reverses the direction of one part of the solution. The main idea is to build a new route by crossing over itself and reordering it, maintaining the tour structure.
- Double insertion: this operator inserts a segment of two consecutive customers from one route to another.
- CROSS-Exchange: this operator simultaneously swaps segments of customers between two routes. Sets of 1-2, 2-2, 1-3, 2-3 and 3-3 swaps are defined and executed in the listed order. The orientation of the segments and of the routes is preserved. These moves usually increase the chances of finding high-quality solutions and avoid becoming trapped in a local minimum.

Figure 2 shows the procedures used within the steps of the VNTS. It illustrates the total cost of the solutions (y-axes) versus the search coordinate (x-axes). An ant generates a starting solution for the VNTS (point A). Then, the neighborhood structure N_1 is selected and TS is performed to determine a new solution in N_1 (point B). It can be seen that TS intensifies the search in N_1 allowing non-improving moves and preventing the search getting trapped in local optima. As a new best solution is reached, the search returns to N_1 and the process is repeated. Then, if a new best solution is not found, the neighborhood structure N_2 is selected and TS is executed again. Since a local optimum for a given neighborhood structure is not necessarily so for another, the search continues and a new solution is obtained in N_2 (point C). As mentioned before, when a new best solution is reached, the search returns to N_1, otherwise the search will explore the next neighborhood. Therefore, it can be observed that the change of neighborhoods clearly leads to escape from local optima and intensifies the search in order to reach under-explored areas of the solution space.

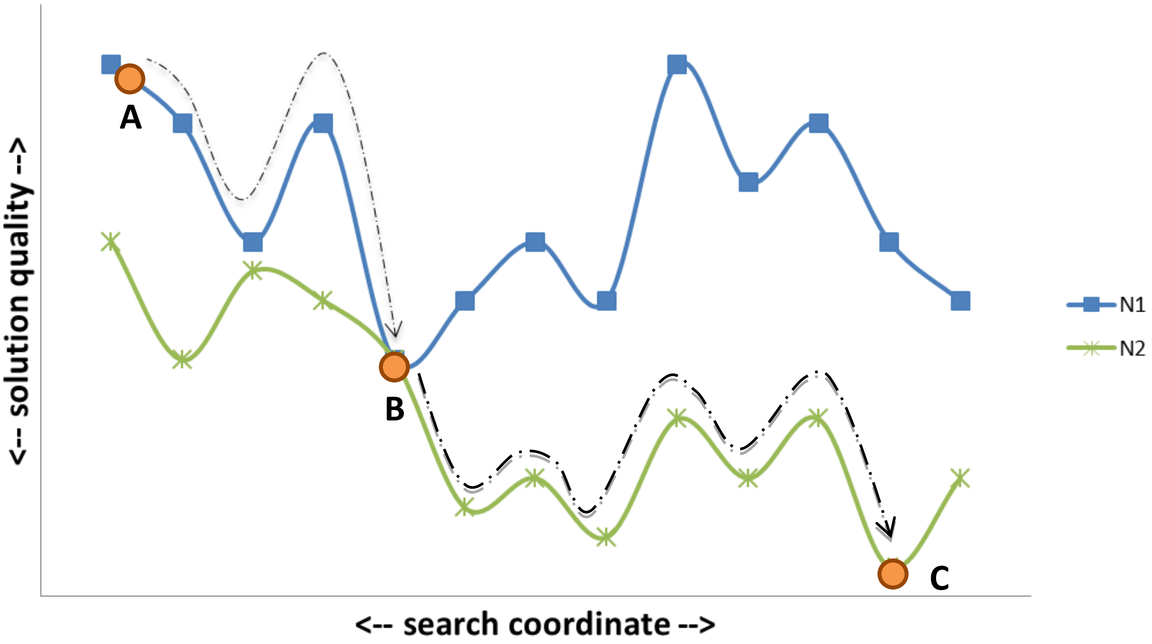

Figure 2. Intensification mechanism in VNTS algorithm

The route construction phase of the ACS algorithm may result in obtaining solutions that may not necessarily include all customers to be served. For that purpose, a HL containing the list of the unserved customers is included in the solution scheme. Therefore, in order to obtain feasible solutions and to avoid getting trapped in local minima, the HL incorporates some additional moves at every iteration and neighborhood structure of the VNTS:

- *Relocate from HL*: this involves transferring a customer from the HL to an existing route. This move is examined in every iteration of the local search process of a N_λ. This type of moves is prioritized in the local search in order to find feasible solutions.
- *Relocate to HL*: this involves transferring a customer from an existing route to the HL. This move occurs when all feasible relocate (inter-route) moves in a local search process are tabu. In this case the HL favors the procedure to search for better solutions by going through the infeasible solution space (Lim and Zhang, 2007).
- *Exchange with HL*: this involves exchanging one customer from an existing route with another in the HL. This type of move only occurs when Exchange (inter-route) neighborhood is used.

3.3 An illustrative example

In this section, an illustrative example is presented in order to make the Mem-ACS more transparent and clear. We consider the six node network of Figure 3(a), with two different vehicles at the depot to serve customers 1, 2, 3, 4 and 5. Each customer has a load demanded of 4, 2, 2, 2 and 3 tons respectively, as shown in figure 3(a). The fleet is heterogeneous, as it is characterized by vehicles of different capacities (7 tons for vehicle-1 and 6 tons for vehicle-2). Time windows restrictions are not considered to make the example more illustrative.

The first phase of the ACS is the ant´s route construction. Thus, an ant randomly chooses a vehicle (e.g. vehicle-1) and initializes the route with the furthest customer from the depot (customer 2). Next, the route is constructed by incrementally inserting customers at any position on the route until none can be inserted due to capacity restrictions. As explained in section 3.1, the decision making about inserting customers is based on a probabilistic rule taking into account both the attractiveness of the move and the pheromone information. In this manner, the ant builds the route-1 (D-1-2-D), as shown in figure-3(a). Note that no customer can be additionally inserted in this route since it would not satisfy capacity restrictions. Then, the ant randomly chooses another vehicle from the depot (vehicle-2) and the process is repeated, obtaining the route-2 (D-5-3-D). As there are not any additional vehicle in the depot, customer 4 is placed in the HL and an initial solution is obtained. Next, the trail pheromone of the arcs of the vehicle routes are updated for the following ant´s route constructions.

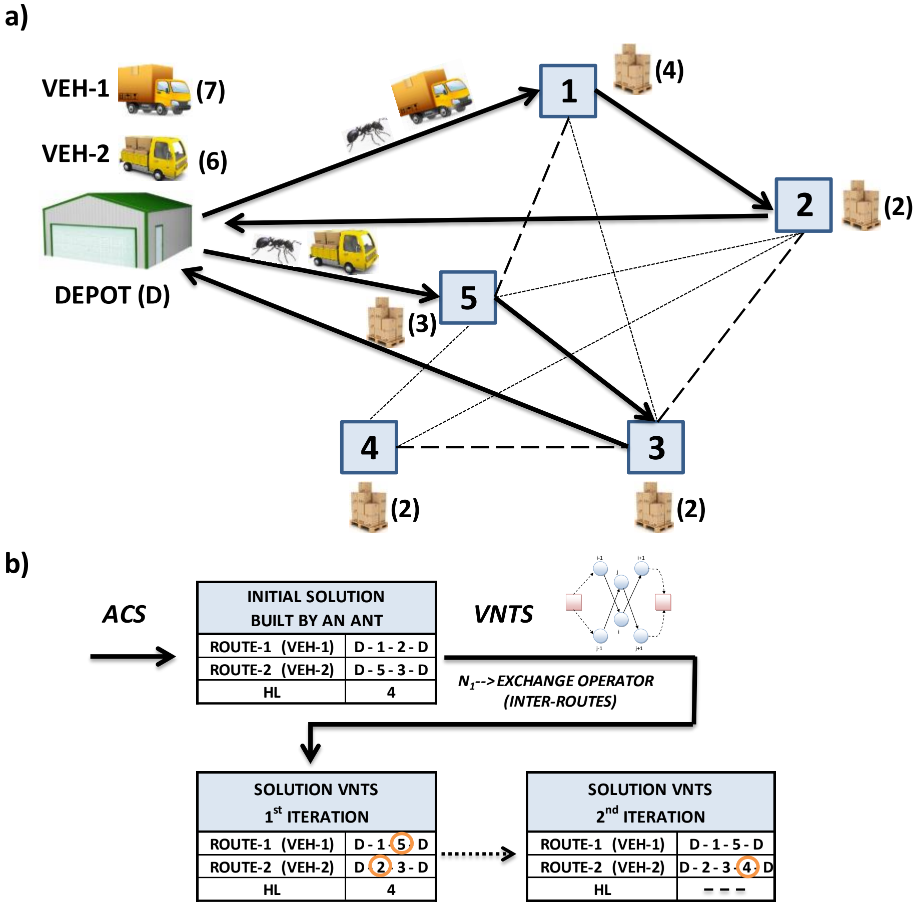

Figure 3 (a) Graph representing the HVRP with five customers solved by the ACS. The pheromone trails have different intensities: strong (—), medium (- - -) and weak (···). (b) Example of transferring a customer from the HL to an existing route in the VNTS.

Once an initial solution is obtained, the local search phase is performed by a VNTS algorithm, as shown in figure-3(b). The first neighborhood structure (N_1) is selected (e.g. exchange inter-route operator) and a TS is performed. At each iteration, the best solution of N_1 which is not included in the tabu list or satisfies the aspiration criterion is selected as the new current solution. Following this procedure, in the first iteration, customers 5 and 2 are simultaneously exchanged, as shown in figure-3(b).

In addition, at every iteration of the TS, two different types of moves are also considered. Firstly, moves that consist in transferring a customer from the HL to an existing route are also examined and prioritized in order to find feasible solutions. Thus, in the second iteration, customer 4, which is placed in the HL, is transferred to the route-2 since the new solution increases the number of served customers and satisfies all problem restrictions. This type of moves is always analyzed as it does not depend of the neighborhood structure selected. Secondly, moves that involve exchanging one customer from an existing route with another in the HL are also examined in this neighborhood structure.

Since a new best solution is reached, the search again returns to N_1 and continues as explained in section 3.2.

Algorithm 1: Memetic Ant colony system algorithm

```
Initialization of parameters (α1, α2, α3, α4, iter_max, M, ρ, α, β, q0, iter_tabu_max, TTmin, TTmax);
S ← Semi-parallel_Insertion_Heuristic (α1, α2, α3, α4), τ(N,N)← Pheromone_matrix();
τ0← Calculate_pheromone_value (S); Δτ^bs← τ0;
τ(N,N)← Initialize_pheromone_matrix (τ0);
iterations=1;
   While (iterations ≤ iter_max) AND (CPU time consumed ≤ γ) do:
    ant=1;
       While (ant ≤ M) AND (CPU time consumed ≤ γ) do:
          S' ← Ant_Route_Construction (α, β, q0, α1, α2, α3, α4);
          τ(N,N)← Local_Update_pheromone_matrix (S', ρ, τ0);
          S'' ← VNTS (S', iter_tabu_max, TTmin, TTmax);
          If  f (S'') improves f (S) then
                S ← S´´;  Δτ^bs← Calculate_pheromone_value (S'');
          ant=ant+1;
        EndWhile
    τ(N,N)← Global_Update_pheromone_matrix (S, ρ, Δτ^bs);
    iterations=iterations+1;
   EndWhile
```

Algorithm 2: Ant Route Construction Procedure (*α, β, q₀,* α₁, α₂, α₃, α₄)

```
Initialize available vehicle list Vk, k=1,2,…,K;
Initialize customer list Cs, s=1,2,…,N;
While (Cs ≠0) AND (Vk ≠ 0) do:
   k←PickatRandom (Vk), Vk← RemoveVehicle(k);
    seed←FindSeedCustomer (Cs, k);
    If (seed≠ 0):
       rk ← InitializeRoute(k),  rk←InsertSeedCustomer(seed), Cs ← RemoveCustomer(seed), Select_cust ←True;
          While (Cs ≠0) AND (Select_cust) do:
           Select_cust ←False;
             q←ChooseRandomNumber();
             For all customers u of  Cs do:
               For all insertion positions i,j of  rk do:
                   ηi,j,u,k ←GreedyFunction(i,j,u,k, α1, α2, α3, α4);
                   Pi,j,u,k ←Calculate_Probabilities( ηi,j,u,k, τ(N,N));
               EndFor
            EndFor
            If (Pi,j,u,k ≠ 0):
                If (q<= q0):
                   n←Maximum_Value(Pi,j,u,k);
                Else:
                   n←Roulette_Wheel_Selection (Pi,j,u,k);
                EndIf
                rk ← InsertAT(i, j,n,k); Cs ← RemoveCustomer(n)
               Select_cust ←True
            EndIf
         EndWhile
    EndIf
 EndWhile
 If (Cs ≠0) do:
```

$r_k \leftarrow$ *Insert_AT_HoldingList*(C_s);
EndIf

<u>Algorithm 3: VNTS Procedure</u> (S', *iter_tabu_max*, TT_{min}, TT_{max})

Define a set of neighborhood structures N_λ, $\lambda=1, 2, \ldots, \lambda_{max}$;
$S \leftarrow S'$;
$\lambda=1$;
While ($\lambda \leq \lambda_{\max}$) do:
    $S'' \leftarrow$ *Tabu_Search* (S, λ, *iter_tabu_max*, TT_{min}, TT_{max});
    If $f(S'')$ improves $f(S)$ then
        $S \leftarrow S''$; $\lambda \leftarrow 1$;
    Else
        $\lambda = \lambda + 1$;
    EndIf
EndWhile
Return (S)

4. Experimental approach

This section describes the computational experiments carried out to validate the effectiveness of the memetic algorithm presented in Section 3. The Mem-ACS algorithm was programmed in C++ and run on a 3.30 GHz Intel® Core(TM) i5-2400 CPU. First, section 4.1 describes the benchmark problem instances used for the experiments. Next, section 4.2 describes the testing of the algorithm to identify the best parameter values, while section 4.3 presents the computational results of the proposed algorithm for the HVRPTW.

4.1 The benchmark problem instances

The benchmark data sets used in our computational experiments for the evaluation of the Mem-ACS algorithm, are those proposed by Paraskevopoulos et al. (2008) and Jiang et al. (2014). Paraskevopoulos et al. (2008) used the benchmark data sets proposed by Liu and Shen (1999) and extended a fixed fleet for each problem with the best-known solutions of these authors. In Liu and Shen (1999) test cases, different vehicles types, which differ in capacities and costs, are added to the classical Solomon (1987) instances with 100 customers to solve the FSMVRPTW. In total there are 24 instances grouped into 6 types of data sets. Customers are randomly distributed in instances of type R, clustered in type C and semi-clustered in instances of type RC. Problem sets shown by R1, C1 and RC1 have a short scheduling horizon and small vehicle capacities, contrary to R2, C2 and RC2. The total cost of a route is obtained by the sum of the fixed vehicle cost and of the total en-route time, which includes travelling and waiting times. The second set consists of the HVRPTW instances presented by Jiang et al. (2014), which, in a similar way to Liu and Shen (1999), introduced a set of heterogeneous fleets with different fixed costs, variable costs and latest returning times to the depot into the Solomon (1987) test cases.

4.2 Parameter setting for the Mem-ACS

This section addresses the best values selected for the parameters of the Mem-ACS algorithm proposed in this work. Memetic algorithms use a set of parameters which affects the performance of the algorithm and, therefore, it plays an important role for obtaining solutions of high quality. The tests used to do the parameter setting have been randomly selected among the Paraskevopoulos et al. (2008) instances; for this purpose, instances R101, C102, RC103, R201, C202 and RC204 have been selected.

When a certain parameter was tested, the algorithm was run maintaining identical settings for the other parameters. In a first phase, experiments have been done with the following parameter settings: the value parameter α=1 was fixed and three parameters: $\beta \in \{1, 1.5, 2, 2.5, 3\}$, $\rho \in \{0.2, 0.25, 0.3\}$ and $q_0 \in \{0.8, 0.85, 0.9\}$ were analyzed on the above-mentioned instances. In all runs the ACS parameters were set as: *iter_max*=10 and M=50, while the local search parameters were fixed to *iter_tabu_max*=30, TT_{min}=10 and TT_{max}=30, as recommended in Paraskevopoulos et al. (2008). The latter values were held constant for the rest of the experiments. The results obtained with this analysis indicate that different parameter combinations vary the performance of the algorithm and the best obtained values are α=1, β=2, q_0=0.8 and ρ=0.2. Consequently, we will adopt this setting in the remainder of this section.

In order to investigate the relative contribution of the rest of parameters of the evolutionary algorithm, experiments were conducted on the values of *iter_max* and *M*. The first defines a termination criterion of the algorithm limiting the total number of iterations, while the second indicates the population of ants. Both control parameters are expressed as a fractional size of the number of nodes (*N*). The value of *iter_max* was set to 10, 20 and 30% indicating the degree of the evolutionary search while the value of *M* was set to 10, 25, 50 and 75%. As a result, the algorithm was run for each instance with 12 different parameter combinations.

Table 1 presents the results obtained from the parameter setting. It shows the percentage deviation between the solution obtained by the Mem-ACS and the best-known solution (BKS) value, averaged over the six instances. The experiments showed that both parameters have an impact on the solution quality, obtaining the best results when *iter_max* =20%*N and *M*=50%*N. Finally, Table 2 presents the best parameter values obtained in the parameter setting.

Table 1. Average percentage deviations from the best-known solutions with varying *iter_max* and *M*

M	*Iter_max*		
	10% *N*	20% *N*	30% *N*
10% *N*	0.44	0.35	0.33
25% *N*	0.32	0.29	0.26
50% *N*	0.42	**0.20**	0.21
75% *N*	0.48	0.25	0.23

Table 2. Parameter settings for the Mem-ACS algorithm

ACS parameters						*VNTS parameters*	
α	β	ρ	q_0	*Iter_max*	*M*	*iter_tabu_max*	TT_{min}-TT_{max}
1	2	0.2	0.8	20% *N*	50% *N*	30	10-30

4.3 Computational results

This section presents the computational results of the Mem-ACS obtained from the HVRPTW benchmark instances. The algorithm was run ten times for each instance with a maximum running time of 20 minutes. The algorithm finished when it reached the maximum number of iterations or the time limit was exceeded.

Table 3 reports the best results obtained by the Mem-ACS algorithm in Paraskevopoulos et al. (2008) instances, compared to the BKS from the current state-of-the-art solution approaches for the HVRPTW. The algorithms reported in the literature are referenced using the following abbreviations: ReVNTS for Paraskevopoulos et al. (2008) and HEA for Koç et al. (2015). Bold numbers indicate that the algorithms have reached the best solution. In the last column, "=" shows the number of matches and "<" shows the number of new BKS obtained for each instance. Table 3 also presents the composition fleet (Mix) of the solutions and the gap between the Mem-ACS and the BKS. The gap is defined as the percentage increase of the objective function value with respect to the BKS.

Additionally, the average, minimum and maximum gap over all problem instances are also shown. The last rows indicate the number of times to run the algorithm, the type of processor used and the average time.

The results show that the Mem-ACS is competitive, generating ten BKSs (problems R104A, C102A, C103A, C104A, RC101A, RC102A, RC103A, RC104A, R204A and RC204A) and obtaining equivalent solutions on five problems (C101A, C201A, C202A, C203A and C204A). This may be attributed to the fact that the introduction of the VNTS for the local search can diversify the ant colony, explore new solution space and prevent the algorithm from being trapped in a local minimum. For the other nine instances, seven are at most 1% worse and only two of them are more than 1% worse.

Since the Mem-ACS presents an average gap of 0.08%, it was capable of finding 15 equivalent or better solutions in reasonable computational times, outperforming the BKSs on C1A and RC1A instances (see Table 5). Moreover, the algorithm obtains equivalent solutions in C2A instances. Total cost reductions range from -0.58% to -0.01% and the worst-case performance is 1.26%.

As presented in Table 3, the average computational time for the Mem-ACS is approximately 12.24 minutes for the Paraskevopoulos et al. (2008) benchmark instances. Therefore, the results leads to the conclusion that our method produces highly competitive quality solutions at dealing with HVRPTWs in reasonable time, especially when solving instances containing smaller vehicle capacities and cluster or random locations of customers.

On the other hand, Table 4 shows the results obtained by Mem-ACS compared to the BKSs obtained by Jiang et al. (2014). To the best of our knowledge, we are the first authors to apply a different solution approach in their instances. The first column of the table shows the instance category. Then, the fixed costs (FC), the variable costs (VC) and the total costs (TC) are showed. The gap between the Mem-ACS and the BKS is also presented. The last rows indicate the minimum, average and maximum gap over all problem instances and the average values for fixed and variable costs. The number of times to run the Mem-ACS, the type of processor used and the average time are also shown.

The Mem-ACS provides quality solutions with an average deviation of -1.17% and a worst-case performance of 0.00%. The algorithm found 55 new best solutions and matched one within an average computation time of 13.29 minutes.

Looking at the results obtained on the instances, on average the Mem-ACS yields 1.57% lower vehicle fixed costs than the BKS. Moreover, the Mem-ACS also decreases the average variable cost by 0.59%, compared with the BKS. The results indicate that our algorithm obtains better fleet compositions while reducing distribution costs within a modest computational effort.

Table 3. Results for Paraskevopoulos et al. (2008) HVRPTW instances

Instance	Fleet	BKS in the literature			Mem-ACS			BKS	
		Mix	TC	Reference	Mix	TC	Gap (%)	=	<
R101A	$A^{1} B^{11} C^{11} D^{1}$	$B^{10} C^{11} D^{1}$	**4583.99**	ReVNTS	$B^{10} C^{11} D^{1}$	4641.79	1.26	0	0
R102A	$A^{1} B^{4} C^{14} D^{2}$	$A^{1} B^{4} C^{13} D^{2}$	**4376.54**	HEA	$A^{1} B^{2} C^{14} D^{2}$	4422.93	1.06	0	0
R103A	$B^{7} C^{15}$	$B^{6} C^{15}$	**4195.05**	ReVNTS	$B^{6} C^{15}$	4198.80	0.09	0	0
R104A	$B^{9} C^{14}$	$B^{9} C^{13}$	4027.69	HEA	$B^{7} C^{14}$	**4004.51***	**-0.58**	0	1
C101A	$A^{1} B^{10}$	B^{10}	**8828.93**	ReVNTS, HEA	B^{10}	**8828.93**	**0.00**	1	0
C102A	A^{19}	A^{19}	7137.79	ReVNTS	A^{19}	**7119.35***	**-0.26**	0	1
C103A	A^{19}	A^{19}	7122.57	HEA	A^{19}	**7105.39***	**-0.24**	0	1
C104A	A^{19}	A^{19}	7083.74	HEA	A^{19}	**7081.51***	**-0.03**	0	1
RC101A	$A^{7} B^{7} C^{7}$	$A^{4} B^{7} C^{7}$	5266.36	HEA	$A^{4} B^{7} C^{7}$	**5257.67***	**-0.16**	0	1
RC102A	$A^{5} B^{6} C^{8}$	$A^{4} B^{5} C^{8}$	5099.55	HEA	$A^{2} B^{6} C^{8}$	**5072.33***	**-0.53**	0	1
RC103A	$A^{11} B^{2} C^{8}$	$A^{10} B^{2} C^{8}$	4991.29	HEA	$A^{10} B^{2} C^{8}$	**4990.94***	**-0.01**	0	1
RC104A	$A^{2} B^{13} C^{3} D^{1}$	$A^{2} B^{13} C^{3} D^{1}$	5016.97	HEA	$A^{2} B^{13} C^{3} D^{1}$	**5003.94***	**-0.26**	0	1
R201A	A^{5}	A^{5}	**3779.12**	ReVNTS	A^{5}	3789.55	0.28	0	0
R202A	A^{5}	A^{5}	**3578.91**	ReVNTS	A^{5}	3585.31	0.18	0	0
R203A	$A^{4} B^{1}$	$A^{4} B^{1}$	**3553.92**	HEA	$A^{4} B^{1}$	3573.78	0.56	0	0
R204A	A^{5}	A^{5}	3081.80	HEA	A^{5}	**3080.44***	**-0.04**	0	1
C201A	$A^{4} B^{1}$	$A^{4} B^{1}$	**6140.64**	ReVNTS, HEA	$A^{4} B^{1}$	**6140.64**	**0.00**	1	0
C202A	$A^{1} C^{3}$	$A^{1} C^{3}$	**7623.96**	HEA	$A^{1} C^{3}$	**7623.96**	**0.00**	1	0
C203A	$C^{2} D^{1}$	$C^{2} D^{1}$	**7303.37**	ReVNTS, HEA	$C^{2} D^{1}$	**7303.37**	**0.00**	1	0
C204A	A^{5}	A^{5}	**5680.46**	HEA	A^{5}	**5680.46**	**0.00**	1	0
RC201A	$C^{1} E^{3}$	$C^{1} E^{3}$	**5523.15**	ReVNTS	$C^{1} E^{3}$	5550.88	0.50	0	0
RC202A	$A^{1} C^{1} D^{1} E^{2}$	$A^{1} C^{1} D^{1} E^{2}$	**5132.08**	ReVNTS	$A^{1} C^{1} D^{1} E^{2}$	5148.90	0.33	0	0
RC203A	$A^{1} B^{1} C^{5}$	$A^{1} B^{1} C^{5}$	**4471.92**	HEA	$A^{1} B^{1} C^{5}$	4473.13	0.03	0	0
RC204A	$A^{14} B^{2}$	$A^{14} B^{2}$	4241.83	HEA	$A^{14} B^{2}$	**4234.46***	**-0.17**	0	1
Nº BKS								**5**	**10**
Average							**0.08**		
Max							**1.26**		
Min							**-0.58**		
Runs							**10**		
Processor							**Core i5 3.3 GHz**		
Avg Time (minutes)							**12.24**		

Table 4. Results for HVRPTW Jiang et al (2014) instances

Instance	BKS in the literature				Mem-ACS			Instance	BKS in the literature				Mem-ACS		
	F.C	V.C	T.C.	%Dev.	F.C	V.C	T.C.		F.C	V.C	T.C.	%Dev.	F.C	V.C	T.C.
HC101	670	1215.33	1885.33	0.00%	670	1215.33	1885.33	HR112	2540	1712.71	4252.71	-0.43%	2540	1694.62	*4234.62
HC102	670	1220.66	1890.66	-3.49%	670	1154.59	*1824.59	HR201	255	1510.74	1765.74	-7.05%	250	1391.31	*1641.31
HC103	650	1258.04	1908.04	-2.96%	650	1201.65	*1851.65	HR202	255	1281.81	1536.81	-5.35%	270	1184.66	*1454.66
HC104	670	1139.78	1809.78	-1.06%	670	1120.54	*1790.54	HR203	255	1082.39	1337.39	-1.07%	270	1053.08	*1323.08
HC105	670	1184.73	1854.73	-0.22%	670	1180.70	*1850.70	HR204	275	839.94	1114.94	-0.55%	250	858.79	*1108.79
HC106	670	1210.64	1880.64	-0.86%	670	1194.49	*1864.49	HR205	270	993.91	1263.91	-0.33%	270	989.74	*1259.74
HC107	670	1169.52	1839.52	-0.00%	670	1169.45	*1839.45	HR206	270	910.44	1180.44	-0.19%	225	953.20	*1178.20
HC108	670	1156.49	1826.49	-0.31%	670	1150.74	*1820.74	HR207	225	877.06	1102.06	-0.89%	225	867.23	*1092.23
HC109	670	1129.22	1799.22	-0.42%	670	1121.71	*1791.71	HR208	225	782.00	1007.00	-0.94%	225	772.52	*997.52
HC201	520	793.28	1313.28	-0.00%	520	793.25	*1313.25	HR209	225	894.04	1119.04	-0.18%	225	891.98	*1116.98
HC202	540	743.58	1283.58	-1.93%	500	758.83	*1258.83	HR210	275	1032.53	1307.53	-2.66%	225	1047.79	*1272.79
HC203	540	719.97	1259.97	-0.21%	540	717.36	*1257.36	HR211	225	785.22	1010.22	-0.74%	225	777.71	*1002.71
HC204	540	716.09	1256.09	-0.32%	500	752.08	*1252.08	HRC101	3330	2373.97	5703.97	-1.60%	3270	2342.57	*5612.57
HC205	520	805.84	1325.84	-0.45%	540	779.88	*1319.88	HRC102	3420	2136.02	5556.02	-0.12%	3360	2189.20	*5549.20
HC206	540	723.63	1263.63	-0.10%	540	722.39	*1262.39	HRC103	3510	1928.89	5438.89	-0.27%	3390	2034.35	*5424.35
HC207	520	787.35	1307.35	-3.52%	540	721.39	*1261.39	HRC104	3450	1881.41	5331.41	-0.47%	3390	1916.52	*5306.52
HC208	500	690.81	1190.81	-0.35%	500	686.67	*1186.67	HRC105	3570	2135.79	5705.79	-1.16%	3390	2249.81	*5639.81
HR101	3030	2095.52	5125.52	-3.04%	2760	2209.52	*4969.52	HRC106	3510	2018.42	5528.42	-1.65%	3390	2047.39	*5437.39
HR102	3040	1942.39	4982.39	-1.37%	3040	1874.35	*4914.35	HRC107	3450	2001.31	5451.31	-1.65%	3390	1971.36	*5361.36
HR103	2880	1781.22	4661.22	-0.29%	2790	1857.81	*4647.81	HRC108	3510	1812.31	5322.31	-0.78%	3450	1831.05	*5281.05
HR104	2940	1590.55	4530.55	-3.10%	2740	1650.16	*4390.16	HRC201	2900	1601.65	4501.65	-0.02%	2900	1600.72	*4500.72
HR105	2700	1870.39	4570.39	-1.01%	2700	1824.06	*4524.06	HRC202	2950	1458.53	4408.53	-0.86%	2950	1420.70	*4370.70
HR106	2610	1821.31	4431.31	-1.02%	2590	1796.13	*4386.13	HRC203	2900	1421.87	4321.87	-0.22%	2900	1412.26	*4312.26
HR107	2700	1691.14	4391.14	-0.20%	2680	1702.37	*4382.37	HRC204	2900	1406.65	4306.65	-1.12%	2950	1308.25	*4258.25
HR108	2580	1700.05	4280.05	-0.72%	2540	1709.14	*4249.14	HRC205	2900	1552.88	4452.88	-0.16%	2900	1545.61	*4445.61
HR109	2540	1799.86	4339.86	-0.68%	2540	1770.30	*4310.30	HRC206	2900	1519.09	4419.09	-0.68%	2900	1488.86	*4388.86
HR110	2540	1732.25	4272.25	-4.91%	2540	1522.31	*4062.31	HRC207	2950	1393.55	4343.55	-0.62%	2900	1416.65	*4316.65
HR111	2670	1696.32	4366.32	-0.87%	2650	1678.41	*4328.41	HRC208	2900	1376.15	4276.15	-0.34%	2900	1361.43	*4261.43
Min				-7.05%											
Average				-1.17%											
Max				0.00%											
Av. FC		1728.66		-1.57%		1701.61									
Av. VC		1376.92		-0.59%		1368.84									
Runs						10									

Processor	**Core i5 3.3 GHz**
Avg Time (minutes)	**13.29**

Table 5. Average results for Paraskevopoulos et al. (2008) HVRPTW instances.

Instance set	Mem-ACS		BKS
	TC	Gap(%)	TC
R1A (4)	4317.01	0.49%	**4295.82**
C1A (4)	**7533.79**	**-0.13%**	7543.26
RC1A (4)	**5081.22**	**-0.24%**	5093.54
R2A (4)	3507.27	0.25%	**3498.44**
C2A (4)	**6687.11**	**0.00%**	**6687.11**
RC2A (4)	4851.84	0.20%	**4842.25**

5. Conclusions

The HVRPTW is an important and practical problem in real-world applications. This paper proposes a new memetic ACS scheme (Mem-ACS) that incorporates VNTS as a local search procedure to solve the HVRPTW.

To the best of the author´s knowledge this is the first work that combines an ACS algorithm with a VNTS procedure for solving a combinatorial problem. The major idea underlying our method is the use of the ACS algorithm to get a diversified exploration over the search space while VNTS intensifies the search in promising regions.

ACS algorithms are adaptive construction heuristics in which a colony of ants generates starting solutions for the local search algorithm. These obtained solutions may not necessarily include all customers to be served when solving a HVRPTW. Therefore, a HL which holds temporarily unserved customers is integrated within the algorithm to introduce flexibility when tackling infeasible solutions.

In the subsequent local search phase, the VNTS provides an intensification of the search by the use of different neighborhood structures and by allowing non-improving moves in order to preventing getting trapped in local optima. In this phase, unserved customers placed in the HL may be transferred to specific positions in existing routes. For that purpose, these additional movements are analyzed and prioritized at every iteration and neighborhood structure of the VNTS. Other movements as exchanging customers from existing routes are also allowed in the HL, if a solution with a best objective function is obtained.

Computational experiments were carried out on HVRPTW benchmark instances from the scientific literature. The Mem-ACS was applied to the 24 HVRPTW instances of Paraskevopoulos et al. (2008). The results indicate that the Mem-ACS obtained 10 new best solutions and matched 5, producing high quality solutions. In the HVRPTW instances of Jiang et al (2014), the Mem-ACS improved 55 solutions out of 56, being able in some instances to obtain better fleet compositions to reduce the total costs. These results showed that the suggested method provides feasible solutions of acceptable quality to all types of HVRPTW instances. Moreover, the Mem-ACS is relevant when customers of a specific route planning cannot be served due to a limitation in the number of vehicles. In these situations, the Mem-ACS maximizes the vehicles' utilization, reduces the number of unserved customers in a route planning and also reduces the total costs.

The Mem-ACS could be extended to solve the HVRPTW with simultaneous pickup and delivery introducing slight modifications in the algorithm. This will be the subject of future research.

Acknowledgements

This research has been fully funded by the Andalusia Government through grants P10-TEP-6332.

References

Abdulkader, M. M., Gajpal, Y., & ElMekkawy, T. Y. (2015). Hybridized ant colony algorithm for the multi compartment vehicle routing problem. Applied Soft Computing, 37, 196-203.

Acampora, G., Gaeta, M., & Loia, V. (2011). Hierarchical optimization of personalized experiences for e-Learning systems through evolutionary models. Neural Computing and Applications, 20(5), 641-657.

Baldacci, R., Battarra, M., Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In: The vehicle routing problem: latest advances and new challenges. Springer, US, pp. 3–27.

Bansal, J. C., Sharma, H., Arya, K. V., & Nagar, A. (2013). Memetic search in artificial bee colony algorithm. Soft Computing, 17(10), 1911-1928.

Belfiore, P. P., & Fávero, L. P. L. (2007). Scatter search for the fleet size and mix vehicle routing problem with time windows. Central European Journal of Operations Research, 15(4), 351-368.

Belmecheri, F., Prins, C., Yalaoui, F., & Amodeo, L. (2009). An ant colony optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. IFAC Proceedings Volumes, 42(4), 1550-1555.

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies–A comprehensive introduction. Natural computing, 1(1), 3-52.

Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Computers & Operations Research, 38(1), 140-151.

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

Bräysy, O., Porkka, P. P., Dullaert, W., Repoussis, P. P., & Tarantilis, C. D. (2009). A well-scalable metaheuristic for the fleet size and mix vehicle routing problem with time windows. Expert Systems with Applications, 36(4), 8460-8475.

Cattaruzza, D., Absi, N., Feillet, D., & Vidal, T. (2014). A memetic algorithm for the multi trip vehicle routing problem. European Journal of Operational Research, 236(3), 833-848.

Chen, X., Ong, Y. S., Feng, L., Lim, M. H., Chen, C., & Ho, C. S. (2013). Towards believable resource gathering behaviours in real-time strategy games with a memetic ant colony system. Procedia Computer Science, 24, 143-151.

Cordeau, J. F., Laporte, G., Savelsbergh, M. W., & Vigo, D. (2007). Vehicle routing. Handbooks in operations research and management science, 14, 367-428.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. Management science, 6(1), 80-91.

Dell'Amico, M., Monaci, M., Pagani, C., & Vigo, D. (2007). Heuristic approaches for the fleet size and mix vehicle routing problem with time windows. Transportation Science, 41(4), 516-526.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on evolutionary computation, 1(1), 53-66.

Dorigo, M., & Birattari, M. (2010). Ant colony optimization (pp. 36-39). Springer US.

Dullaert, W., Janssens, G. K., Sörensen, K., & Vernimmen, B. (2002). New heuristics for the fleet size and mix vehicle routing problem with time windows. Journal of the Operational Research Society, 53(1), 1232-1238.

El Fallahi, A., Prins, C., & Calvo, R. W. (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers & Operations Research, 35(5), 1725-1741.

Ellabib, I., Calamai, P., & Basir, O. (2007). Exchange strategies for multiple ant colony system. Information Sciences, 177(5), 1248-1264.

Fathi, M., Rodríguez, V., & Alvarez, M. J. (2014). A novel memetic ant colony optimization-based heuristic algorithm for solving the assembly line part feeding problem. The International Journal of Advanced Manufacturing Technology, 75(1-4), 629-643.

Gendreau, M., Hertz, A., Laporte, G., & Stan, M. (1998). A generalized insertion heuristic for the traveling salesman problem with time windows. Operations Research, 46(3), 330-335.

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA.

Golden, B. L., & Wong, R. T. (1981). Capacitated arc routing problems. Networks, 11(3), 305-315.

Golden B., Assad A., Levy L. and Gheysens F.G. (1984), The Fleet Size and Mix Vehicle Routing Problem, Computers and Operation Research, 11, pp. 49-66.

Hansen, P., & Mladenovic, N. (2003). A tutorial on variable neighborhood search. Groupe d'études et de recherche en analyse des décisions, HEC Montréal.

Hansen, P., Mladenović, N., & Pérez, J. A. M. (2010). Variable neighborhood search: methods and applications. Annals of Operations Research, 175(1), 367-407.

Hoff, A., Andersson, H., Christiansen, M., Hasle, G., & Løkketangen, A. (2010). Industrial aspects and literature survey: Fleet composition and routing. Computers & Operations Research, 37(12), 2041-2061.

Ioannou, G., Kritikos, M., & Prastacos, G., (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. Journal of the Operational Research Society, 52(5), 523-537.

Irnich, S., Schneider, M., & Vigo, D. (2014). Chapter 9: Four Variants of the Vehicle Routing Problem. In Vehicle Routing: Problems, Methods, and Applications, Second Edition (pp. 241-271). Society for Industrial and Applied Mathematics.

Jiang, J., Ng, K. M., Poh, K. L., & Teo, K. M. (2014). Vehicle routing problem with a heterogeneous fleet and time windows. Expert Systems with Applications, 41(8), 3748-3760.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Kennedy J (2006) Swarm intelligence. Handbook of Nature-Inspired and Innovative Computing, Springer, Boston, pp 187–219.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2015). A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. Computers & Operations Research, 64, 11-27.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. European Journal of Operational Research, 249(1), 1-21.

Lau, H. C., Sim, M., & Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. European Journal of Operational Research, 148, 559-569.

Lim, A., & Zhang, X. (2007). A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows. INFORMS Journal on Computing, 19(3), 443-457.

Liu F.H. & Shen S.Y. (1999). The fleet size and mix vehicle routing problem with time windows. Journal of the Operational Research Society, 50(7),721-732, 1999.

Mavrovouniotis, M., & Yang, S. (2011). A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. Soft Computing, 15(7), 1405-1425.

Mendoza, J. E., Castanier, B., Guéret, C., Medaglia, A. L., & Velasco, N. (2010). A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Computers & Operations Research, 37(11), 1886-1898.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research, 24(11), 1097-1100.

Molina, J. C., Eguia, I., & Racero, J. (2018). An optimization approach for designing routes in metrological control services: a case study. Flexible Services and Manufacturing Journal, 30(4), 924-952.

Molina, J. C., Eguia, I., & Racero, J. (2019). Reducing pollutant emissions in a waste collection vehicle routing problem using a variable neighborhood tabu search algorithm: a case study. TOP, 27(2), 253-287.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826, 1989.

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., & Prastacos, G. P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. Journal of Heuristics, 14(5), 425-455.

Price, K., Storn, R. M., & Lampinen, J. A. (2006). Differential evolution: a practical approach to global optimization. Springer Science & Business Media.

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

Repoussis, P., Paraskevopoulos, D., Tarantilis, C., & Ioannou, G. (2006). A reactive greedy randomized variable neighborhood tabu search for the vehicle routing problem with time windows. Hybrid Metaheuristics, 124-138.

Repoussis, P. P., & Tarantilis, C. D. (2010). Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming. Transportation Research Part C: Emerging Technologies, 18(5), 695-712.

Salmeron, J. L., Mansouri, T., Moghadam, M. R. S., & Mardani, A. (2019). Learning fuzzy cognitive maps with modified asexual reproduction optimisation algorithm. Knowledge-Based Systems, 163, 723-735.

Salmeron, J. L., Ruiz-Celma, A., & Mena, A. (2017). Learning FCMs with multi-local and balanced memetic algorithms for forecasting industrial drying processes. Neurocomputing, 232, 52-57.

Semet, F. (1995). A two-phase algorithm for the partial accessibility constrained vehicle routing problem. Annals of Operations Research, 61(1), 45-65.

Sharifipour, H., Shakeri, M., & Haghighi, H. (2018). Structural test data generation using a memetic ant colony optimization based on evolution strategies. Swarm and Evolutionary Computation, 40, 76-91.

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Operational Research 35(2), 254-265.

Stützle, T., & Dorigo, M. (1999). ACO algorithms for the quadratic assignment problem. New ideas in optimization, (C50), 33.

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

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S., (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research, 152(1), 148-158.

Tavakkoli-Moghaddam, R., Saremi, A. R., & Ziaee, M. S. (2006). A memetic algorithm for a vehicle routing problem with backhauls. Applied Mathematics and Computation, 181(2), 1049-1060.

Toth P & Vigo D (eds) (2014) Vehicle routing: problems, methods, and applications, vol 18. SIAM, Philadelphia

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research, 234(3), 658-673.

Wang, H., Moon, I., Yang, S., & Wang, D. (2012). A memetic particle swarm optimization algorithm for multimodal optimization problems. Information Sciences, 197, 38-52.

Yang, J. H., Sun, L., Lee, H. P., Qian, Y., & Liang, Y. C. (2008). Clonal selection based memetic algorithm for job shop scheduling problems. Journal of Bionic Engineering, 5(2), 111-119.

Yu, B., Yang, Z. Z., & Yao, B. (2009). An improved ant colony optimization for vehicle routing problem. European journal of operational research, 196(1), 171-176.

Zhang, C., Hei, X., Yang, D., & Wang, L. (2016). A memetic particle swarm optimization algorithm for community detection in complex networks. International Journal of Pattern Recognition and Artificial Intelligence, 30(02), 1659003.