A study on the heterogeneous fleet of alternative fuel vehicles: Reducing CO₂ emissions by means of biodiesel fuel

Mohamed Amine Masmoudi[*]

Geneva School of Business Administration
University of Applied Sciences Western Switzerland (HES-SO)
1227 Carouge, Switzerland
E-mail: mohamed.masmoudi@hesge.ch

Manar Hosny

Computer Science Department,
College of Computer and Information Sciences (CCIS),
King Saud University (KSU), Riyadh, Saudi Arabia
E-mail: mifawzi@ksu.edu.sa

Emrah Demir

Panalpina Centre for Manufacturing and Logistics Research,
Cardiff Business School, Cardiff University, Cardiff CF10 3EU, United Kingdom
E-mail: demire@cardiff.ac.uk

Naoufel Cheikhrouhou

Geneva School of Business Administration
University of Applied Sciences Western Switzerland (HES-SO)
1227 Carouge, Switzerland
E-mail: naoufel.cheikhrouhou@hesge.ch

[*] Corresponding author:

Email address: mohamed.masmoudi@hesge.ch (Mohamed Amine Masmoudi)
Phone: +41(0)225463530

A study on the heterogeneous fleet of alternative fuel vehicles: Reducing CO₂ emissions by means of biodiesel fuel

Abstract

In the context of home healthcare services, patients may need to be visited multiple times by different healthcare specialists who may use a fleet of heterogeneous vehicles. In addition, some of these visits may need to be synchronized with each other for performing a treatment at the same time. We call this problem the Heterogeneous Fleet Vehicle Routing Problem with Synchronized visits (HF-VRPS). It consists of planning a set of routes for a set of light duty vehicles running on alternative fuels. We propose three population-based hybrid Artificial Bee Colony metaheuristic algorithms for the HF-VRPS. These algorithms are tested on newly generated instances and on a set of homogeneous VRPS instances from the literature. Besides producing quality solutions, our experimental results illustrate the trade-offs between important factors, such as CO_2 emissions and driver wage. The computational results also demonstrate the advantages of adopting a heterogeneous fleet rather than a homogeneous one for the use in home healthcare services.

***Keywords*:** Vehicle routing; heterogenous fleet; Synchronization; Alternative fuel; Artificial Bee Colony metaheuristic algorithm

1. Introduction

Home healthcare (HHC) services for the elderly and the disabled has become a vital topic of research, due to the expected increase in the aging population. According to the Administration on Aging (AoA), the US department of Health and Human Services estimates that the number of people aged 65 and over will increase from 14.5% in the year 2014 to 21.7% by 2040. This has led to an increasing trend in developing ambulatory HHC services that can provide care and treatment for those people in the comfort of their homes.

In practice, HHC patients need multiple and synchronized visits from different healthcare specialists. For example, a patient with motor disability may require a visit by more than one health care specialists (or nurses) at the same time to provide an ideal care, which may include medical treatment, measuring blood pressure, giving injections, personal hygiene, physical therapy, and so on. It should be noted, though, that in the majority of HHC services, there usually exist different levels of qualifications among nurses, which reflect their ability to perform certain tasks. For example, there are professional nurses that can perform sophisticated treatments and nursing care, while assistant nurses can only perform some small jobs, such as cleaning, dressing or washing, and they cannot give for example, prescription medications or injections. Our work does not explicitly consider matching the appropriate nurses' skills/qualifications for each patient based on their specific needs or preferences, as studied in the majority of home healthcare research (see e.g., Cappanera and Scutellà, 2014; Breakers et al., 2016). Rather, we assume here that the healthcare specialists are all highly qualified nurses, who are allowed to perform services even if requiring lower skills. To sum up, the main objective of our work is planning a set of routes for a qualified set of nurses to provide high quality healthcare and social services for a set of patients in their homes, located in different geographical locations, within certain pre-specified time windows.

From a research perspective, HHC services have given rise to a new family of the Vehicle Routing Problem (VRP) called the VRP with Synchronization (VRPS). Besides the usual constraints of the standard VRP, the VRPS requires full synchronization between the vehicles (i.e., in terms of time, location, payload or similar aspects). Recent studies of the VRPS include, for example, Afifi et al. (2016); Haddadene et al. (2016); and Redjem and Marcon (2016). In our study, we, therefore, consider several important aspects that distinguish our work from existing research in the domain of vehicle routing. We call this new variant of the VRP, the Heterogeneous Fleet VRP with Synchronized visits (HF-VRPS). In VRPS studies, conventional Light Duty Vehicles (LDVs) (i.e., passengers' cars) with gasoline or diesel fuel are used to serve the patients' requests. However, these types of fuel are a main source of emissions, such as, greenhouse gases (GHGs) and air pollutants. Aiming to overcome such environmental problems, we consider Alternative Fuel Vehicles (AFVs), using different types of fuel, such as, biodiesel, ethanol, propane, or hydrogen.

Real-life HHC applications usually require the utilization of various types of vehicles. For example, IMAD Geneva is a social profit organization in Geneva, Switzerland. It is part of the network of care instituted by the cantonal law on the network of care and home care. This organization provides home care services by a set of nurses (more than 10,000 nurses in 2016) for HHC patients (more than 16,000 in 2016) whose health require temporary or sustainable care or have difficulties in performing activities of daily living. IMAD Geneva offers for their employees different modes of transportation, such as, 268 electric bicycles, 172 conventional bicycles, 13 electric quadracycles, 35 hybrid passengers' cars (small and medium cars), 11 electric passengers cars, 4 vans (large cars), 50 conventional passengers' cars (mini, small, medium, and large cars) and other available private/public types of vehicles. Planning efficient routes that satisfy the demand is one of the main concerns of IMAD Geneva. Therefore, it is necessary for this profit organization to minimize the distance as well as the cost of fuel consumption, while complying with the time constraints of each patient. More information about IMAD Geneva and some relevant statistical and activities reports can be found in IMAD Geneva (2016).

Based on observations from real-life HHC applications as explained above, we adopt in our work a heterogeneous fleet of vehicles within the context of the VRPS. We consider a limited fleet size of AFVs with diverse features. More specifically, the heterogeneous fleet in our study incorporates different vehicles types that use different fuel types. Apparently, considering heterogeneous vehicles is more complex and realistic than assuming a homogeneous fleet, as previously assumed for most of the VRPS applications.

According to the U.S EPA (2017), LDVs can be classified into several categories, namely mini cars, small cars, medium cars, large cars, executive cars and sport utility cars. In our study, we only consider three main vehicle types that can be adopted by HHC companies. These include small cars (Mercedes-Benz E250-bluetec, 2014), medium cars (BMW 328d, 2017) and large cars (Audi A8 L, 2017). These specific vehicle types are chosen based on the classification found in U.S EPA (2016, 2017), and their capacity to work with an alternative fuel source, such as biodiesels.

In practical settings, it is also important to take into consideration the different routes, equipment, and services that can be carried out by the healthcare providers according the needs of the patients. In fact, providing the necessary material and medical equipment for the patients' visits in their homes is one of the decisions related to HCC resource planning (Benzarti et al. 2013). Thus, different types of vehicles with different trunk sizes can be considered in planning the appropriate vehicles. For instance, some nurses may

perform only very few visits that require only simple care (e.g., injections, dressings, and personal hygiene), during their working day. A small passenger car is sufficient in this case, since the nurse does not need to carry heavy or bulky equipment to the patient's home. However, based on some observations and statistical/activities reports from IMAD Geneva, nurses may need large medical equipment (e.g., wheelchair, oxygen tank). Typically, medium to large cars are considered in this case. In addition, as affirmed by the activity report of IMAD Geneva (2016), some nurses may need to move for long distances covering many visits that may use several medical equipment for one or more patients, as well as some visits that need additional social services, such as meal delivery or shopping. Large passengers' cars will be the best option in these cases, since using small/medium passenger cars may lead to the dissatisfaction of patients. Finally, medium and large passengers' cars, despite their relative high price, usually do not need frequent maintenance services (Yavuz et al, 2015), which is beneficial for the companies on the long run. Thus, considering a fleet composed of mini, medium and large passengers' cars can be considered as the best option to satisfy the diverse needs of patients as well as reducing the routing costs, including the cost of fuel.

The use of alternative fuel in routing problems is usually studied under the category of the Green Vehicle Routing Problem (GVRP) (Erdoğan and Miller-Hooks, 2012). However, the majority of GVRP studies consider a constant fuel consumption rate. Nevertheless, this assumption is not practical, since fuel consumption depends on many factors, such as speed, fuel type, vehicle load, road slope, etc. In Bektaş and Laporte (2011), the authors introduced the Pollution-Routing Problem (PRP) which adopts a fuel consumption function that is inspired from the Comprehensive Modal Emission Model (CMEM) developed by Barth et al. (2005). We have adopted the CMEM function to calculate the fuel consumption rate of the AFVs by considering biodiesel as a substitute to diesel.

The main advantage of using the CMEM is twofold; first, the CMEM depends on the fuel type; second, biodiesel is viewed as alternative fuel that can be used in conventional diesel engines, either on its own or mixed with diesel (Verma and Sharma, 2016). In addition, the properties and suitability of biodiesel does not require any adjustments in the engine of a specific vehicle due to the use of biodiesel instead of (or blended with) petroleum-based diesel. Thus, the new diesel-powered vehicles are designed to operate on biodiesel without modification (Beiter and Tian, 2016). We note, though, that due to the different types of blends of biodiesel with petroleum diesel, we particularly consider two common types of blends that are approved for use in diesel engines by major manufacturers (i.e., Volkswagen, Renaud, Audi, and Mercedes-Benz). For additional information on biodiesel fuels, the readers are referred to Xue et al. (2011) and Mahmudul et al. (2017).

To sum up, the HF-VRPS is a combination of the standard VRPS, the heterogeneous VRP and the PRP. This problem variant is particularly relevant to HCC companies that operate a fleet of AFVs to serve the disabled and the elderly. As far as we know, our research is the first attempt in the literature to handle such extensive variant. The HF-VRPS is NP-hard since it is an extension of the classical VRP and more complex than the traditional VRPS. For this reason, we have developed metaheuristic algorithms in an attempt to obtain good-quality solutions within relatively short computational time to solve large size instances of the HF-VRPS.

The remainder of this paper is organized as follows. Section 2 starts with state-of-the-art VRPS studies and the relevant literature to the investigated problem. Section 3 presents a summary of the scientific

contribution of this paper. Section 4 starts with the description of the new problem variant HF-VRPS followed by its mathematical formulation. Section 5 contains the proposed hybrid algorithms; Hybrid Artificial Bee Colony with Demon Algorithm (ABC-DA), Hybrid Artificial Bee Colony with Old Bachelor Acceptance (ABC-OBA), and Hybrid Artificial Bee Colony with Record-to-Record Travel (ABC-RRT). Section 6 contains the computational results with a comparison to existing algorithms. Section 7 provides conclusions.

2. Literature Review

The VRPS is a variant of the VRP that has many applications in HHC routing and scheduling. For a complete overview of different variants of VRPs, the readers can find comprehensive surveys in Bräysy and Gendreau (2004a, b) and Montoya-Torres et al. (2015). This section contains a brief literature review on the VRPS and the relevant PRP studies.

2.1. The vehicle routing problem with synchronization (VRPS)

The VRPS is a promising research area that was first introduced by Bredström and Rönnqvist (2008) with HHC application services for the elderly. The authors proposed a heuristic algorithm for the VRPS by considering a multi-criteria objective function, minimizing preferences, travel time, and maximal workload difference. In another study, Kergosien et al., (2009) proposed a mixed integer programming (MIP) formulation to deal with multiple Traveling Salesman Problems (TSPs) with time windows and synchronization in homecare context. In another study, Afifi et al. (2016) developed a simulated annealing algorithm improved by several known iterative local search operators. The algorithm is tested on the benchmark instances of Bredström and Rönnqvist (2008) by considering travel time and preferences as parts of the objective function. To the best of our knowledge, Afifi et al. (2016) provided the best results for the VRPS benchmark instances. Recently, Haddadene et al. (2016) proposed an MIP formulation and developed a hybrid method combining a greedy randomized adaptive search procedure and iterative local search to solve the VRPTW with synchronizations and precedence constraints.

In general, metaheuristics have shown their effectiveness in solving a variety of VRPS. For example, Liu et al. (2013) proposed two metaheuristic approaches, GA (Genetic Algorithm) and TS (Tabu Search) to solve a simultaneous pickup and medicine delivery problem with time windows. Recently, Decerle et al. (2017) studied the home healthcare routing and scheduling problem with route balancing. A Memetic Algorithm (MA) embedded with several local search operators is developed to evaluate the multi-objective method on the VRPTW benchmark instances of Solomon (1987). The results show that the MA is able to find a good trade-off between the minimization of total traveled distances, patients' soft time windows and shared visits, and maximal distance difference between routes.

Although metaheuristics are popular in this field, some studies are based on mixed integer linear programming models or exact methods, and solved using commercial solvers like CPLEX. Nevertheless, these models are only capable of solving small sized instances.

Interested readers can find more details on the VRPS in Liu et al. (2013), Mankowska et al. (2014) and Ceselli et al. (2014). Survey papers on the variants of synchronization constraints can also be found in Drexl (2012) and Fikar and Hirsch (2017).

2.2. The pollution-routing problem (PRP)

A realistic variant of the Green Vehicle Routing Problem (GVRP) that has attracted the interest of researchers in recent years is the PRP, where many aspects (e.g., vehicle speed, driver's wage, fuel type, etc.) are explicitly considered (see, e.g., Bektaş and Laporte, 2011; Demir et al., 2012). To solve the PRP, an Adaptive Large Neighborhood Search (ALNS) metaheuristic algorithm along with Speed Optimization (SOP) algorithm was proposed by Demir et al. (2012). Later, Franceschetti et al. (2017) proposed an enhanced ALNS algorithm for the time-dependent version of the PRP, where traffic congestion is considered. In a recent study, Eshtehadi et al. (2017) investigated robust models for the solution of the PRP with demand and travel time uncertainty. Several interesting and related studies on the PRP and other related green vehicle routing problems can be found in (e.g., Dabia et al., 2016; Qian and Eglese, 2016; Fukasawa et al., 2016; Qiu et al., 2017). Survey papers that focus on energy consumption in the context of green vehicle routing, can be found in Demir et al. (2014, 2015).

The GVRP is one important variant of the VRP that specifically necessitates a heterogeneous fleet of vehicles; firstly, because vehicles may produce a variety of emissions; and secondly, because logistic agencies may acquire heterogeneous vehicle types involving various categories of engines (fuel, electric, or hybrid), recent and old models, as well as vehicles from different brands. Latest studies about the heterogeneous GVRP, where CO_2 emissions reduction are considered, can be found in the work of Kwon et al. (2013), Juan et al. (2014), Kopfer et al. (2014), and Koç et al. (2014).

As can be observed from our brief review in this section, in previous studies, many realistic conditions and constraints, which are related to the specificities of HHC services, have been simply ignored. These include, alternative fuel vehicles, heterogeneous fleet, realistic fuel consumption rate, etc. To the best of our knowledge, a fleet of heterogeneous vehicles composed of alternative fuel vehicles, as well as a realistic fuel consumption rate, are considered here for the first time in the context of the VRPS. In addition, previous studies on the VRPS have limitations on the size of instances, where only small and medium instances are tested with up to 16 nurses (vehicles) and 73 visits (see e.g., Afifi et al., 2016 and Haddadene et al., 2016).

It can be seen from the reviewed literature that a few solving techniques that were proposed for the VRPS were based only on linear programming (Kergosien et al., 2009), exact methods (Dohn et al., 2009; Rasmussen et al., 2012) and modified (meta)heuristics (Mankowska et al., 2014; Redjem and Marcon, 2016, Afifi et al., 2016). These methods lacked the use of efficient hybrid metaheuristic methods (except the hybrid GRASP-ILS method of Haddadene et al., 2016). In fact, Salas et al. (2016) and Masmoudi et al. (2016) suggest that the hybridization of population-based metaheuristics with advanced local search mechanism (i.e., single solution-based metaheuristics) is effective to solve such complex variants of the VRP. We, therefore, propose three hybrid population-based metaheuristics based on ABC algorithm.

3. Scientific contributions of the paper

This work studies the joint impact of using a heterogeneous fleet of biodiesel vehicles (passengers' cars) and routing on emissions in the VRPS. The contributions of this work are as follows. *i*) a Heterogeneous

Fleet VRP with Synchronized visits (HF-VRPS) is introduced, which extends the traditional VRPS of Bredström and Rönnqvist (2008), to allow for the use of alternative fuel passengers' cars with biodiesels instead of the traditional conventional vehicles with diesel, *ii*) a mathematical formulation of the problem is given by considering the PRP objective function introduced by Bektaş and Laporte (2011) that accounts for fuel consumption, CO_2 emissions, and different types of vehicles, *iii*) three effective hybrid metaheuristic methods based on Artificial Bee Colony Algorithm (ABC), namely hybrid Artificial Bee Colony with Demon algorithm (ABC-DA), hybrid Artificial Bee Colony with Old Bachelor Acceptance (ABC-OBA), and hybrid Artificial Bee Colony with Record-to-Record Travel (ABC-RRT) algorithms are proposed to solve the HF-VRPS.

The use of ABC for the HF-VRPS is motivated by its favorable performance in the field of combinatorial optimization, such as vehicle routing and scheduling problems. An additional advantage of our hybrid approach is that it combines the benefits of ABC in terms of diversification as well as the benefits of DA, OBA and RRT in terms of intensification., *iv*) a new set of instances based on the benchmark instances of the VRPS of Bredström and Rönnqvist (2008) is introduced, *v*) from the numerical experiments, we demonstrate that our algorithms provide good-quality solutions on both new and existing benchmark instances. In addition, we show that our hybrid ABC-DA, ABC-OBA, and ABC-RRT, clearly outperform the DA, OBA, RRT, and ABC as standalone algorithms, and *vi*) we provide managerial insights on the trade-offs between important factors, such as travel cost, fuel consumption and CO_2 emissions. Our analysis also demonstrates the benefit of using a heterogeneous fleet in our particular application.

4. Problem Definition

The proposed HF-VRPS model is different from other VRPS studies by also considering the CO_2 emissions in the objective function of the well-known PRP of Bektaş and Laporte (2011). In the following subsections, we provide the problem definition as well as the calculation of energy consumption, and the mathematical model of the HF-VRPS, which is inspired from the mathematical formulation of Bredström and Rönnqvist (2008).

4.1. Problem definition

The HF-VRPS can be formally described as follows. Let $G = (V, A)$ be a directed graph with a node-set $V = \{N \cup C\}$, where $N = \{1, \dots, |N|\}$ is the set of visits to patients and $C = \{0, n + 1\}$ corresponds to the origin and destination depot, respectively. Let $A = \{(i, j): i, j \in V, i \neq j\}$ be the set of arcs connecting each pair of nodes. Each arc (i, j) in set A has associated a travel distance d_{ij}. A limited fleet of heterogeneous vehicles $K = \{1, \dots, |K|\}$ (index k) is assumed to be available at the initial depot 0, to be used by the K nurses to carry out all the daily visits to patients. Each visit i must start within the patient's time window preference $[a_i, b_i]$ where a_i and b_i are respectively the earliest starting time of the service and the latest starting time. The time window $[a_0, b_0] = [a_{n+i}, b_{n+i}]$ is the available time for all vehicles. Some patients may need multiple visits from their professional nurse(s). In this case, the patient needs to be visited by two distinct vehicles. These two visits must be synchronized. We use $(i, j) \in P^{synch}$ to denote the set of visits

that are synchronized, where i and j are associated with the same patient, i.e., the two visits must be conducted in the same time. In addition, a service time duration s_i is associated with each visit i ($\forall\ i \in N$).

Similar to the relevant studies in the field of PRP (Demir et al., 2012), we use the Comprehensive Modal Emission Model (CMEM) of Barth et al. (2005) to estimate the fuel consumption and CO_2 emissions. The fuel consumption rate FR_{ij} over the course of an arc (i,j) is calculated as: $FR_{ij} = \frac{\xi}{e.\text{D}}(rFD + \frac{P_{ij}}{q.q_{zf}})$, where P_{ij} represents the mechanical power: $P_{ij} = (\frac{1}{2}.C_d^k.\ u.R.\overline{(v^t)}^2 + m.\ g.(\sin(\sigma) + C_c.\cos(\sigma))).\ \bar{v}^t$. We also consider a discretized speed function with T non-decreasing speed levels $\bar{v}^t$ ($t=1,\dots,T$). A list of values for the common parameters (Demir et al., 2012) for all vehicle types, and specific parameters for each vehicle type (Mercedes-Benz E250-bluetec, 2014; BMW 328d, 2017; and Audi A8 L, 2017) are given in Tables 1 and 2, respectively.

The HF-VRPS consists of determining a set of routes satisfying the patients' requirements while minimizing a function comprising fuel costs, emissions, and driver wages. A solution should satisfy the following assumptions: i) all patients must be served, ii) each service of a patient must start within the specified time window, iii) each vehicle trip must start and end at the same depot and its accumulated travel time cannot exceed a maximum imposed duration L_{max}, iv) each visit is served by exactly one vehicle (nurse), v) the synchronized couple of visits must start simultaneously, and vi) the vehicle speed over the course of an arc should be optimized.

Let x_{ij}^k be a binary variable equal to 1, if and only if the vehicle k goes from i to j and 0 otherwise. z_i^k is a continuous variable that indicates the starting time of visit i if this latter is visited by vehicle k. Let y_{ij}^{tk} be a binary variable equal to 1 if and only if the vehicle k goes from i to j at speed level $t=1,\dots,T$. The total time spent by a vehicle k on a route in which $j \in N$ is the last node before returning to the depot, is denoted by o_j^k.

Table 1
Parameters used in the HF-VRPS

Notation	Description	Value
g	Gravitational constant (meter/second^2)	9.81
u	Air density(kilogram/meter^3)	1.2041
C_c	Coefficient rolling friction	0.01
ξ	Fuel-to-air mass ratio	1
e	Heating value of typical biodiesel fuel (kilojoules per gram)	41.5*
D	Factor converting the fuel rate (grams per second to liters per second)	737
∂	Efficiency parameter for biodiesel engines	0.90*
q_{zf}	Drive train efficiency	0.45
σ	Road angle	0
τ	Acceleration (m/second^2)	0
v^l	Lower vehicle speed (kilometer/hour)	20
v^s	Upper vehicle speed (kilometer/hour)	90
f_c	Fuel and CO_2 emissions cost per liter (£)	1.4
f_d	Driver wage (£/ second)	0.0022

Source: Sivaramakrishnan and Ravikumar (2011) and US EPA (2016)

Table 2
Vehicle specific parameters

Notation	Description	Small (k=1)	Medium (k=2)	Large (k=3)
M^k	Vehicle mass (kilogram)	1,592	1,845	2,095
R^k	Frontal surface area of the vehicle(meter^2)	2.12	2.51	2.35

C_d^k	Coefficient of aerodynamic drag	0.28	0.25	0.34
F^k	Engine speed (revolution per second)	66.66	63.33	65.50
H^k	Engine friction factor (kilojoules per revolution per liter)	0.38	0.50	0.85
D^k	Engine displacement (liters)	2.0	2.1	3.0

4.2. Mathematical formulation

$$min \quad \sum_{k \in K} \sum_{(i,j) \in A} \lambda f_c H^k F^k D^k d_{ij} \sum_{t=1}^{T} y_{ij}^{tk} / \overline{v}^t \tag{1}$$

$$+ \sum_{k \in K} \sum_{(i,j) \in A} \lambda f_c \gamma M^k \alpha_{ij} d_{ij} x_{ij}^k \tag{2}$$

$$+ \sum_{k \in K} \sum_{(i,j) \in A} \lambda f_c \beta \gamma d_{ij} \sum_{t=1}^{T} (\overline{v}^t)^2 / y_{ij}^{tk} \tag{3}$$

$$+ \sum_{k \in K} \sum_{j \in N} f_d o_j^k \tag{4}$$

subject to

$$\sum_{k \in K} \sum_{j:(i,j) \in A} x_{ij}^k = 1 \qquad \forall \in V \setminus \{n+1\} \tag{5}$$

$$\sum_{j:(0,j) \in A} x_{0j}^k = \sum_{j:(j,n+1) \in A} x_{j(n+1)}^k = 1 \qquad \forall k \in K \tag{6}$$

$$\sum_{j:(i,j) \in A} x_{ij}^k = \sum_{j:(j,i) \in A} x_{ji}^k \qquad \forall i \in V \setminus \{n+1\}, \forall k \in K \tag{7}$$

$$z_i^k - z_j^k + s_i + \sum_{t=1}^{T} d_{ij} y_{ij}^{tk} / \overline{v}^t \le M_{ij}(1 - x_{ij}^k) \qquad \forall i \in V, \forall j \in N, i \ne j, \forall k \in K \tag{8}$$

$$a_i \le z_i^k \le b_i \qquad \forall i \in N, \forall k \in K \tag{9}$$

$$z_j^k + s_j - o_j^k + \sum_{t=1}^{T} d_{j0} y_{j0}^{tk} / \overline{v}^t \le L_{ij}(1 - x_{j0}^k) \qquad \forall i \in V, \forall k \in K \tag{10}$$

$$\sum_{t=1}^{T} y_{ij}^{tk} = x_{ij}^k \qquad \forall (i,j) \in A, \forall k \in K \tag{11}$$

$$\sum_{k \in K} z_i^k = \sum_{k \in K} z_j^k \qquad \forall (i,j) \in P^{synch} \tag{12}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in A, \forall k \in K. \tag{13}$$

$$y_{ij}^{tk} \in \{0,1\} \qquad \forall (i,j) \in A, t = 1, ..., T, \forall k \in K. \tag{14}$$

The objective function is derived from Bektaş and Laporte (2011) and contains four components. The two first terms (1) and (2) calculate the costs incurred by the engine module and the weight module of the vehicle, term (3) computes the cost induced by variations in speed. Finally, term (4) calculates the total driver wage, where, $\lambda = \xi / (eD)$, $\gamma = 1/(1000 q_{zf} \partial)$ and $\alpha = \tau + .g.(\sin(\sigma) + C_r.\cos(\sigma))$ (Bektaş and Laporte, 2011). Constraints (5) ensure that each vehicle leaves from the depot and returns to the depot. Constraints (6) guarantee that vehicles enter and leave given nodes. Constraints (7) ensure the continuity of the routes. Constraints (8), (9) and (10) impose the time windows, where $M_{ij} = max\{0, z_i^k + s_i + d_{ij}/v^l - a_j\}$ and L_{ij} is a big number. Constraints (11) guarantee that only one speed level is chosen for each arc (i,j). Constraints (12) ensure the synchronization between couples of visits. Finally, constraints (13) and (14) define the domain set of decision variables. Before we conclude this section we note that the problem defined by the mathematical model above is hard to solve to optimality, since this entails solving several sub-problems together, namely the heterogeneous VRP and the synchronized visits, both of them are already difficult to solve to optimality. In fact, commercial solvers can provide exact solutions for only small instances having just few customers. The best alternative in such cases is to use metaheuristics, hence we propose three hybrid variants of ABC to solve this problem, as described in the next section.

5. Hybrid Artificial Bee Colony Algorithms for the HF-VRPS

This section presents three different variants of the Artificial Bee Colony (ABC) algorithm to solve the HF-VRPS. These variants are Hybrid Artificial Bee Colony with Demon Algorithm (ABC-DA), Hybrid Artificial Bee Colony with Old Bachelor Acceptance (ABC-OBA), and Hybrid Artificial Bee Colony with Record-to-Record Travel (ABC-RRT). The DA, OBA, and RRT are well-known variants of the Simulated Annealing (SA) metaheuristic algorithm. In fact, we have chosen these algorithms to evaluate which of them tackles our problem more effectively, when embedded within the ABC algorithm. To the best of our knowledge, these three algorithms, as well as the ABC algorithm, are not studied in the literature to solve the VRPS.

The ABC is inspired by the natural foraging behavior of honey bees to find (near-)optimal solutions. It was first proposed by Karaboga (2005) and has been successfully applied to various optimization problems including the VRP (see, e.g., Zhang et al., 2014; and Marinakis and Marinaki, 2014). In many studies, the ABC was already shown to be quite effective compared to other population-based algorithms (Szeto et al., 2011). The ABC is also suitable for wider exploration of the search space, but it is poor for deeper exploitation of promising areas, as discussed in Liu and Liu (2013), making it interesting to investigate how to improve the convergence of the ABC algorithm. For this purpose, we propose the hybridization of the basic ABC with other intensification (i.e., single solution-based) search techniques. This differs from the traditional ABC which relies on single-step neighboring moves only to improve the selected solutions. In addition, we apply the traditional Boltzmann function of Kirkpatrick et al. (1984) to occasionally accept a worse solution instead of always accepting better solutions to escape local optima.

As in the original ABC metaheuristic of Karaboga (2005), our algorithm contains three main phases: employed, onlooker and scout bees stages. The main steps of the ABC are shown in Algorithm 1. First, an effective heuristic is used to generate the initial population of size P. Then, the algorithm runs for a number of iterations in the three phases, each one is carried out repeatedly in a loop until the stopping criterion is reached.

In the first phase (employed bees), for each solution x_{sol} of the initial population, a new solution x_{sol}' is generated using a local search operator {I1, I2, I3 or I4}. The algorithm then checks whether the new solution is feasible. If the new solution improves the current solution (i.e., the number of vehicles $k_v(x_{sol}')$ is equal or less than the number of available vehicles k^v and the objective function of (x_{sol}') is smaller than that of (x_{sol}), it replaces the current solution and $Trival_{sol}$ is set to zero. $Trival_{sol}$ represents the number of times that the current solution cannot provide a better solution. On the other hand, if the new solution does not improve the current solution (i.e., in case $k_v(x_{sol}') \leq k^v$ and the objective function of (x_{sol}') is not smaller than that of (x_{sol})), the new solution may be accepted subject to the SA acceptance condition defined as $e^{\Delta/T_{iter}}$. This is implemented by randomly generating a number $0 < \beta < 1$ and replacing x_{sol} with x_{sol}' when $\beta < e^{\Delta/T_{iter}}$, where $\Delta = f(x_{sol}) - f(x_{sol}')$ and T_{iter} is the current temperature at a given ABC iteration $iter$, which is initialized to $T_{max}(T_{iter}=T_{max})$. We note that this method is inspired from Braekers et al. (2014) and is different from other state-of-the-art heuristics for the VRPS. Finally, if the new solution is infeasible or not accepted, we retain the old solution x_{sol} and $Trival_{sol}$ is incremented by one. In each iteration of this phase, the new (or retained) solution x_{sol} is stored in a list L.

In the second phase (onlooker bees phase), unlike to the first phase, we first select from the population the participating solutions using roulette wheel selection, where the probability $p(sol)$ of selecting a solution sol is calculated with the following formula: $p(sol) = \frac{fit_{sol}}{\sum_j^P fit_j}$, where fit_{sol} represents the objective function of solution sol. Then, for each solution from the P solutions selected in the onlooker phase, we apply an effective metaheuristic to improve the solution. In this case three metaheuristics are used: DA, OBA and RRT; each one of them is hybridized separately with the ABC. In fact, this part of our ABC constitutes the novelty of our approach, where we enhance the intensification around the onlooker solutions to further improve their quality. At the end of each iteration of the onlooker bees phase (similar to the employed bees phase), the improved (or retained) solution x_{sol} is stored in a list L.

In the first two phases, if a solution x_{sol} cannot be improved by local search or by a metaheuristic (DA, OBA or RRT) during a predefined number of attempts, called $Limit$, the solution is assumed to be abandoned. This marks the start of the third phase (the scout bees). In this phase, a new solution is created using a construction heuristic that replaces each abandoned solution and the value of $Limit$ is then reset to zero. The best solution x_{best} from the stored solutions in list L is then selected, and if x_{best} is better than the global best solution x_{best}^*, x_{best} becomes a new (global) best solution.

At each ABC iteration, the temperature value of our acceptance criterion is reduced using the formula: $T_{iter} = \delta * T_{iter-1}$, where δ is a constant equal to 0.999. Finally, the best solution x_{best}^* is returned from the hybrid ABC, when no improvement of the best solution is achieved after ten consecutive iterations.

At the end of the hybrid ABC algorithm, the Speed Optimization algorithm (SOP) of Demir et al. (2012) is run on the resulting final solution to optimize the speed values on each arc of the route in the solution. This is intended to further improve the objective function value. The detailed description of the SOP algorithm can be found in Demir et al. (2012).

Algorithm 1: The hybrid Artificial Bees Colony Algorithm

Initialization; Generate the initial population P using a set of construction heuristics x_{sol} with $sol=1,\ldots,P$; k^v the number of available vehicles; $f(x_{best}^*) = \infty$ and $Trival_{sol}=0$ and $L=\emptyset$; $iter = 1$; $T_{iter} = T_{max}$

Repeat

    // ****Employed bees phase****

    For $sol=1$ to P

        Apply a randomly selected local search operator from {I1, I2, I3 or I4} to the current solution x_{sol} to obtain a new solution x_{sol}';

        If x_{sol}' is feasible **Then**

            If (x_{sol}' is better than x_{sol}) **Or** (accepted by the SA acceptance criterion) **Then**

                $x_{sol} \leftarrow x_{sol}'$;

                $Trival_{sol} \leftarrow 0$;

            Else //Retain the old solution x_{sol}

                $x_{sol} \leftarrow x_{sol}$;

                $Trival_{sol} = Trival_{sol} + 1$;

        Else //Retain the old solution x_{sol}

            $Trival_{sol} = Trival_{sol} + 1$;

        Memorize the solution x_{sol} in list L;

    End For

    //****Onlooker bees phase****

    Select P solutions from the employed bees using Roulette Wheel selection

    For $sol=1$ to P

        Perform DA, OBA or RRT metaheuristic on the current solution x_{sol} to obtain new solution x_{sol}';

        If x_{sol}' is feasible **Then**

            If (x_{sol}' is better than x_{sol}) **Or** (accepted by the SA acceptance criterion) **Then**

                $x_{sol} \leftarrow x_{sol}'$;

$$Trival_{sol} \leftarrow 0;$$
Else //Retain the old solution x_{sol}
$$x_{sol} \leftarrow x_{sol};$$
$$Trival_{sol} = Trival_{sol} + 1;$$
Else //Retain the old solution x_{sol}
$$x_{sol} \leftarrow x_{sol};$$
$$Trival_i = Trival_i + 1;$$
End If
Memorize the solution x_{sol} in list L;
End For
//*** Scout bees phase***
For $sol = 1$ to P
    If $Trival_{sol} \geq$ Limit **Then**
        Replace x_{sol} by a selected construction heuristic solution;
        $Trival_{sol} = 0;$
    End If
End For
$$T_{iter} = \delta * T_{iter-1};$$
$$iter = iter + 1;$$
Select the best solution (x_{best}) from the stored solutions in L
    If $k_v(x_{best}) \leq k^v$ **And** $f(x_{best}) < f(x_{best}^*)$ **Then**
        $x_{best}^* \leftarrow x_{best};$
Until No improvement of x_{best}^* after ten consecutive iterations
Return: x_{best}^*
Apply SOP algorithm to x_{best}^*

5.1. A construction heuristic

A simple and efficient heuristic is proposed to generate the population of initial solutions with size P that contains the synchronized visits. In fact, the proposed two-phase heuristic is inspired from the one introduced by Solomon (1987). The first phase consists of constructing a number of routes based on the number of available vehicles and inserting the synchronized visits one by one based on the earliest starting time (a_i). The second phase consists of inserting the remaining patients in the existing routes in the following manner: randomly select a patient's visit i and add it in the best position that respects the time windows and maximum route duration in already existing routes. If there is no feasible insertion of patient i in any existing route without violating the time windows and the maximum route duration, a new route is generated. This procedure is repeated until all visits are served. The construction heuristic is applied P times to generate the initial population.

5.2. Demon algorithm (*DA*)

This section details the first metaheuristic DA that is hybridized with the ABC to improve the selected solutions of the second phase of Algorithm 1. The DA was first introduced by Creutz (1983) and has been used to solve a number of optimization problems (see, e.g., Chandran et al., 2003; Alahmadi et al., 2014). The difference between the DA and the standard SA algorithm is in the method of acceptance of a solution at each step, which is based on the demon (credit) value.

The reason that we chose the DA is that the DA is a relatively simple but efficient algorithm. It only requires the tuning of a single parameter, which is the credit value (*Dem*). The framework of the improved DA is shown in Algorithm 2. First, we select the current solution x and the best solution x_{best} from the P solutions (onlooker bees) that were chosen using Roulette Wheel Selection from the employed bees.

Let Dem be the initial Demon value. The algorithm runs for n_{DA} consecutives iterations. During the search, a neighborhood search operator {N1, N2, N3 or N4} is applied on the current solution x to obtain a new solution x'. The new solution x' is accepted if it is feasible and the difference in fitness (ΔE) is less than or equal to the credit value Dem. The credit value Dem is then updated. If the solution value has improved (i.e., $\Delta E < 0$), the credit value will be increased; otherwise the credit value will be decreased.

Moreover, we have slightly changed our proposed DA compared to the traditional DA to cope with the special characteristics of HF-VRPS. We further apply a local search as an intensification phase to improve the solution obtained by the DA in the previous step. The local search works as follow; we select in a random order a local search operator from {I1, I2, I3 or I4}. If an improvement is found, then the algorithm continues the search; otherwise, the search continues with another selected local search operator. The local search stops when the last operator does not yield an improvement. The solution is then checked as to whether a new global best solution x_{best} has been achieved, by considering the minimum number of additional vehicles and the objective function.

Furthermore, a diversification process $Div(x_{best})$, inspired from Wei et al (2015), is proposed to create a new initial solution from the current best solution x_{best}. This procedure is intended to diversify the search by taking advantage of the best-obtained solution. The idea is that if the current best solution is not improved after n_{div} consecutive iterations, the current solution x is restored with the current x_{best}. The procedure of diversification consists of randomly selecting s patients from x_{best} and inserting them in list L. Then, the algorithm tries to reinsert the patients from list L one-by-one in other routes at their best positions, while respecting the feasibility of the solution. If a patient cannot be added, a random route is chosen and the patients of this route are deleted and put in the list L (one patient at a time) until the given patient is inserted in this route. A new list L is then obtained (if found) and the same insertion procedure is applied until all patients are inserted (i.e., L becomes empty). According to Pisinger and Ropke (2007) deleting a large number of patients may have a considerable effect on the solution. Therefore, we apply the following technique to select the number of patients to be deleted: if the number of patients in the instance is less than 50, s is chosen randomly between [5, 10]; otherwise, s is chosen randomly between [5, 20].

Algorithm 2: The Demon Algorithm

Initialize Demon $Dem = Dem_{max}$; k^v the number of available vehicles and $x = x_{best}$ = the current solution selected from the onlooker bees solutions;

  Repeat

    Perform a neighborhood search operator {N1, N2, N3 or N4} on x to obtain a new solution x'

      If x' is feasible **Then**

        $\Delta E = f(x') - f(x)$;

        If ($\Delta E \leq Dem$) **Then**

        $x \leftarrow x'$ and $Dem = Dem - \Delta E$;

        End if

      End if

    Perform a local search operator {I1, I2, I3 or I4} on x;

    If x is feasible **Then**

      If $k_v(x_{best}) \leq k^v$ **And** $f(x) < f(x_{best})$ **Then** $x \leftarrow x_{best}$;

      End if

    End if

    If x_{best} is not improved after n_{div} consecutive iterations

      $x \leftarrow Div(x_{best})$;

    End If

Until the number of steps n_{DA} is reached

Return: x_{best}

5.3. Old bachelor acceptance algorithm (OBA)

The second metaheuristic that is hybridized with ABC to improve the selected solutions in the second phase of Algorithm 1 is the OBA algorithm. The OBA was introduced by Hu et al. (1995), but was not widely used to solve optimization problems (see, e.g. Hu et al., 1995 and Lari et al., 2008). In our approach, we applied several modifications on the OBA to improve its performance for the HF-VRPS. The framework of our OBA algorithm is shown in Algorithm 3 and it is similar to our proposed DA. Two main differences are in the acceptance of the new solution x' and in the update of the acceptance function. The first is that, if x' is feasible and is better than the current solution (i.e., the objective function of x' is less than the objective of the current solution plus the current threshold value Th_{iter} (initialized to zero)), it is accepted and becomes the new current solution. In addition, in this case, age is restored to zero, where age is the number of iterations that passed since the last-accepted move (initialized to zero). Otherwise, age is incremented. The second difference is that the threshold value Q_i is updated dynamically based on the history of the search. This is done based on the formula: $Th_{iter+1} = \left(\left({age}/{a} \right)^b - 1 \right) * \Delta * (1 - {iter}/{n_{OBA}})^c$, where n_{OBA} represents the number of iterations, Δ represent the granularity of the update. Moreover, a, b and c represent the multiplicative factor in growth rate, power law and coefficient in dumping of threshold magnitude.

Algorithm 3: The Old Bachelor Acceptance Algorithm

Initialize $Th_0 = 0$; $age = 0$; k^v the number of available vehicles; and $x = x_{best}$ = the current solution selected from the onlooker bees solutions

Repeat

        Perform a neighborhood search operator {N1, N2, N3 or N4} on x to obtain a new solution x'

       If x' is feasible **Then**

             If $f(x') < (f(x) + Th_{iter})$ **Then**

                  $age = 0$ and $x \leftarrow x'$;

             Else

                  $age = age + 1$;

             End if

       End if

       $Th_{iter+1} = \left(\left(\frac{age}{a} \right)^b - 1 \right) * \Delta * (1 - \frac{iter}{n_{OBA}})^c$;

       Perform a local search operator {I1, I2, I3 or I4} on x;

       If x is feasible **Then**

             If $k_v(x_{best}) \leq k^v$ **And** $f(x) < f(x_{best})$ **Then** $x \leftarrow x_{best}$;

             End if

       End if

       If x_{best} is not improved after n_{div} consecutive iterations

             $x \leftarrow Div(x_{best})$;

     End If

   Until the number of steps n_{OBA} is reached

Return: x_{best}

5.4. Record-to-record travel algorithm (RRT)

This section describes the third proposed metaheuristic that is hybridized with the ABC, namely Record-to-Record Travel (RRT) algorithm. The RRT algorithm was first introduced by Dueck (1993) and was shown to be able to produce high quality solutions for a variety of VRP extensions (see, e.g., Li et al., 2007

and Derigs et al., 2013). The RRT algorithm, like the DA and the OBA, has the advantage of ease of implementation. In addition, it has only one parameter called Dev, which controls the acceptance of degrading solutions.

The framework of our RRT algorithm is presented in Algorithm 4. The first difference compared to previous algorithms consists of the following; if the objective function of x' is smaller than that of the current value of Rec minus the deviation Dev ($Rec - Dev$), x' becomes the new current solution. The second difference is that, during the search, the value of Rec is updated based on the objective value of the new solution x'.

Algorithm 4: The Record-to-Record Travel Algorithm

Initialize Deviation $Dev > 0$; k^v the number of available vehicles and $x = x_{best}$ = the current solution selected from the onlooker bees solutions and $Rec = f(x)$

      Repeat

            Perform a neighborhood search operator {N1, N2, N3 or N4} on x to obtain a new solution x'

            If x' is feasible **Then**

                  If $f(x') < (Rec - Dev)$ **Then** $x \leftarrow x'$;

                  If $f(x') > Rec$ **Then** $Rec \leftarrow f(x')$;

            End if

            Perform a local search operator {I1, I2, I3 or I4} on x

            If x is feasible **Then**

                  If $k_v(x_{best}) \leq k^v$ **And** $f(x) < f(x_{best})$ **Then** $x \leftarrow x_{best}$;

                  End if

            End if

            If x_{best} is not improved after n_{div} consecutive iterations

                  $x \leftarrow Div(x_{best})$;

            End If

      Until the number of steps n_{RRT} is reached

Return: x_{best}

5.5. Neighborhood search operators

Neighborhood search is an important step in any metaheuristic search in order to generate new solutions that both retain the good parts of previous solutions and perturb the previous solutions sufficiently to diversify the search. In our ABC-DA, ABC-OBA and ABC-RRT algorithms we apply several neighborhood search operators that are inspired from the literature as explained below.

Swap Neighborhood (N1) (N2): This operator is based on Masmoudi et al. (2016) but we adapt it to the characteristics of the HF-VRPS. In this move we first select two routes at random from a solution, and then we select a sequence h of consecutive patients from each route. The nodes included in the sequence h, are deleted from their original routes. After this, the nodes are re-inserted in the second route. As usual, the insertion is done in the best possible insertion position, while respecting the feasibility. If no feasible insertion is found, the lowest cost insertion is selected. When applying this neighborhood in our algorithms, the value of h is chosen randomly between 2 **(N1)** and 3 **(N2)**.

Cross-Exchange (N3): This operator is based on Taillard et al. (1997). It is similar to (N1) and (N2) but simply exchanges two sub-routes from two routes selected at random, while preserving the order of nodes in the sub-routes. This move is intended for more diversification of the.

Remove-route insert-one-by-one (N4): This neighborhood is inspired from Braekers et al. (2014). The role of this operator is to try to reduce the number of vehicles. First, we select a route having the smallest number

of patients. If more than such route exists, one of them is selected randomly. Then, we remove the patients from this route one at a time and reinsert each one in another route. If a patient cannot be inserted in another route, it is kept in its original route.

5.6. Local Search operators

For further intensification around the solutions, we apply local search using several well-known operators. We apply two well-known intra-route operators: the 2-opt (I1) operator adopted from Lin (1965), and the relocate operator (I2) adopted from Savelsbergh (1992). Moreover, we apply two inter-route operators: the 2-opt* operator (I3) of Potvin and Rousseau (1995), and finally the relocate operator (I4) of Savelsbergh (1992).

5.7. Evaluation function

During the search, a new solution is generated and must be evaluated to check its feasibility. Infeasible solutions are not allowed. They are penalized following the evaluation function $f(x) = c(x) + \beta d(x) + \gamma w(x) + \tau s(x)$. The term $c(x)$ represents the fuel consumption and CO_2 emission of solution x. As defined in Cordeau et al. (2001), the terms $d(x)$ and $w(x)$ represent the duration and time window, calculated as $d(x) = \sum_{i \in V} max\{(a_i \text{-} B_i^k), \ 0\}$ and $w(x) = \sum_{j \in V} max\{((B_n^k \text{-} B_0^k) \text{-} L_{max}), \ 0\}$ ($\forall \ k \in K$), respectively. In addition, the term $s(x)$ represents the synchronization of visits constraint violation and is defined by Bredström and Rönnqvist (2008) as $\sum_{(i,j) \in Psynch} max\{(\sum_{k \in K} B_i^k - B_j^k), 0\}$. The penalty parameters β, γ and τ are adjusted dynamically during the search. We note that the solution x becomes a new best solution only if $d(x) = w(x) = s(x) = 0$.

6. Computational Experiments

In this section, we detail the experimental results of our proposed algorithms. All algorithms are implemented in C language and performed on a configuration Intel processor 4 GHz, and 2 GB of RAM operating Windows 7 with 32 bits.

6.1. Data and experimental setting

To evaluate the effectiveness of algorithms, we use benchmark instances of the VRPS proposed by Bredström and Rönnqvist (2008), which are based on the modification of instances generated by Eveborn (2006). These benchmark instances contain between 4-16 vehicles (nurses) and 18-73 patients. These instances are divided into three categories based on the number of patients and vehicles: group (*A1*) has 18 patients and 4 vehicles; group (*A2*) has 45 patients and 10 vehicles; and group (*A3*) has 80 patients and 16 vehicles. In addition, in each group, the number of synchronized visits constitutes around 10% of the numbers of patients, thus approximately ranging between two to eight visits. To calculate the distance (d_{ij}) between any two locations i and j, we use the Euclidean distance based on the coordinates of locations i and j and the length of time windows are classified as small, medium and large.

For large sized instances, a new group (*A4*) based on the benchmark instances of Eveborn (2006) is introduced. The original data set contains 140 patients and 28 vehicles. As a result, our fourth group (*A4*) contains between 80-140 patients and 16-28 vehicles. We follow the same idea of Bredström and Rönnqvist (2008) and Cordeau et al. (2001) to create the characteristics of this group. The coordinates of patients are randomly generated in a in a specific square area (i.e., $[-10, 10]^2$). The lengths of time windows are generated randomly and range from small to large. We generate first a uniform random value a_i in the interval $[60, T\text{-}60]$ and then a uniform random number l_i is chosen in the interval $[a_i+15, a_i+30]$, $[a_i+15, a_i+45]$ and $[a_i+30, a_i+90]$ for the small, medium and large time windows. The patient's service durations are also generated randomly between 10 mins and 30 mins. In addition, about 10% of the number of patients represents the number of synchronized visits, and the maximum duration of working days are assumed identical to that in Bredström and Rönnqvist (2008).

In addition, we assume that at the beginning of the working day, the AFVs are fully refueled. Thus, we eliminate that the vehicle needs to visit AFSs to be refueled. In fact, in the field of VRPS, the patients are located in a small area, which is unlike the GVRP, where the vehicle needs to be refueled at any AFS node, especially for long trips. The vehicles are assigned as follows; first, we start with an overall vehicle number K that is associated with the number of CVs found in the VRPS instances of Bredström and Rönnqvist (2008), and then a small car replaces the first CV, a medium car replaces the second CV, a large car replaces the third CV, a small car replaces the fourth CV, and so on. For more details, the readers can find these data sets and the detailed results of each of our algorithms in http://hfvrps.e-monsite.com.

6.2. Parameters setting

This section provides the experiments conducted to set the parameters of our algorithms. We have chosen the parameters based on both recommendations from the literature and preliminary experiments. We note that the sensitivity analysis for the parameters is done on the basic version of each algorithm, i.e., that does not apply the SOP algorithm at the end of the method. Moreover, it should be noted that since our hybrid methods are compared with standalone methods (i.e., DA, OBA and RRT without hybridization with ABC), there are some differences in the setting of parameters between these two versions.

For the DA, there are several parameters that need to be set. We selected the following initial values: n_{DA}=50,000 iterations and Dem=1,500. The analysis is then carried out on a small data set containing 20 instances (from *A1*, *A2*, *A3* and *A4*) with various levels of the number of patients and time windows. We then tested different combinations of parameters using the initial credit values Dem={1,000, 1,500, 2,000}, and the number of iterations n_{DA}={10,000;…;100,000}. Each instance is run for five times. The average value obtained by each combination is reported in Table 3.

Table 3
Parameter testing for the standard DA

n_{DA}	10.000			20.000			**50.000**			100.000		
Value (Dem)	Avg	Best	CPU (min)	Avg	Best	CPU (min)	Avg	Best	CPU (min)	Avg	Best	CPU (min)
500	329.92	325.24	1.33	319.09	306.07	1.54	301.30	298.43	1.55	294.08	292.41	5.82
1.000	320.23	310.10	1.03	320.28	307.73	1.53	297.86	297.54	3.61	293.87	292.50	7.68
1.500	311.40	306.41	0.93	307.71	302.54	1.63	**293.97**	**292.51**	**2.57**	293.86	292.26	8.11
2.000	320.93	310.70	1.57	320.88	307.75	1.49	296.01	293.47	3.69	293.84	291.93	9.00

Table 3 shows the average and the best objective function values in five runs for the twenty instances used in this experiment, as well as the average CPU time in minutes (the column CPU). We can observe in the table, that using 100,000 iterations with different Dem values does not significantly improve the results than using 50,000 iterations, and it consumes considerably more processing time, which is sometimes larger than twice the time of 50,000 iterations. This can be explained by the fact that the solution obtained after 50,000 iterations is already of very good quality as evident by the results shown in the table. Therefore, the algorithm spends quite a long time after this trying to improve the solution by applying the local search operators as well as our diversification procedure $Div(x_{best})$. Applying these attempts may eventually lead to a slight improvement and indeed escaping the high quality local optimum, but this comes at the expense of considerably more processing time as previously mentioned. Accordingly, we can conclude from Table 3 that the values n_{DA}=50,000 iterations and Dem=1,500 appear to be appropriate in terms of both the solution quality and processing time. This also conforms with the observation of Masmoudi et al. (2016), who recommend using a smaller number of iterations, which slightly sacrifices solution quality in favor of a better processing time.

For the OBA algorithm, we initially used the suggested parameters proposed by Hu et al. (1995): a=1, b=2, c=0.5, Δ=0.093, and the number of iterations n_{OBA} is equal to 50,000 iterations. For the RRT algorithm, the deviation value Dev is equal to (0.01* current global Rec) based on the principal RRT algorithm of Dueck (1993), and the number of iterations n_{RRT} is set to 50,000 iterations.

We note that the number of iterations n_{OBA}, n_{RRT} and n_{DA} of 50,000 used in our standalone algorithms is reduced to only 1,000 iterations (n_{OBA}= n_{RRT}= n_{DA}=1,000) in our hybrid algorithms. The reduction in the number of iterations is intended to limit the computational time in our hybrid framework, as recommended in Masmoudi et al. (2016).

Moreover, we have conducted several experiments to fine-tune the parameters of the hybrid ABC algorithms. In the majority of proposed ABC algorithms, the population size P is equal to 20, 30, 40 or 50. However, the intensification phase (hybridization with DA, OBA and RRT) that we have incorporated in our ABC will need to run several times during each iteration. As a result, it will be computationally expensive. In this regard, we have chosen to test a small size of P (i.e., 10 and 20).

As mentioned by Karaboga (2005), the value of $Limit$ is an important component in the ABC. This value is generally calculated as $P * S$ (see, e.g., Yildiz, 2013; Kıran and Fındık, 2015), where S represents the dimension of the problem (i.e., is the number of users for the VRP. The $Limit$ value calculated this way is thus very large, especially when the number of patients is large. Therefore, we have again chosen to test small values of $Limit$ (i.e., 20, 30, 40, 50 and 60).

Table 4 presents the computational results of the sensitivity analysis on the parameters P and $Limit$. To test the influence of these parameters, we used the ABC-DA algorithm on a small data set containing 20 instances based on different characteristics, such as the number of patients and tight and wide time windows. All experiments were run ten times for each instance. In Table 4, the rows denoted "Best" and "Avg" represent the best and average solution values obtained with the ABC-DA algorithm, while the column "CPU" represents the average run time in minutes.

Table 4

Identification of the best parameter setting for the hybrid ABC-DA

P	**10**					20				
Limit	20	30	40	**50**	60	20	30	40	50	60
Best (ABC-DA)	314.03	313.36	313.54	**313.12**	314.06	313.72	313.78	313.58	313.28	313.23
Average	315.50	314.55	314.98	313.81	315.13	315.81	316.55	316.80	315.70	315.05
CPU (min)	9.29	9.65	10.13	9.41	11.70	15.60	16.39	16.53	19.50	21.20

We note that, the maximum population size *P* and the value of *Limit* have a significant impact on the solution quality. Comparing the results obtained by each method using different parameters, it can be observed that using the population size *P* of 10 and 20 produces good quality solutions for the majority of combinations of the *Limit* value. Our choice of the best parameters is based on the production of a high-quality solution within a shorter amount of CPU time.

The results shown in Table 4 thus confirm that the parameters setting (indicated in bold) *P*=10 and *Limit*=50 provides a good trade-off between solution quality and run time. We note that these values are applied for all hybrid algorithms.

6.3. Computational analysis

This section presents the detailed results obtained for the benchmark VRPS instances, the generated instances of the HF-VRPS, the impact of using speed optimization and the impact of the cost component.

6.3.1 Results on the VRPS benchmark instances

For this experiment, we used VRPS instances of Bredström and Rönnqvist (2008). Table 5 shows the results obtained for the VRPS. We compared these results with the optimal and best known solution values found in the literature. The objective function is to minimize the total travel time of the vehicles ($\sum_{k \in K} \sum_{(i,j) \in A} t_{ij} x_{ij}^k$). We ran our algorithms ten times as done in Afifi et al. (2016). In Table 5, the column "Opt" presents the optimal solution values where known. Column "Best" refers to the best published results obtained by the heuristic of Bredström and Rönnqvist (2008) and by the Simulated Annealing with Iterative Local Search (SA-ILS) method of Afifi et al. (2016). "BKS" refers to the best known solution, which is the minimum result value among the optimal solution ("Opt") and the best solution ("Best"). The columns "Best (%)" and "Avg (%)" are the percent deviations from the best known (average) solutions obtained by our hybrid algorithms in ten runs. Finally, we show the CPU time in seconds ("CPU(s)"). We note that the symbol "-" indicates that no result is obtained for the corresponding instance. The detailed results of this table are shown in the website.

It should be noted that we cannot compare the performance of the algorithms with respect to the computational time with that reported in Afifi et al. (2016) for the sake of a fair comparison with the previously published method. This is because a different machine has been used to run our algorithms than that used for the SA-ILS. Moreover, the speed factor of the configuration applied on our algorithms cannot be estimated by using Dongarra (2014) table, due to lack of relevant information in Dongarra (2014) and in Linpack (www.roylongbottom.org.uk). Thus, we report in Table 5 the computational time only for the record, and not for a direct comparison with the previously published results. In general, our algorithms provide good results in a reasonable computational time.

Table 5
Comparison of our three algorithms with the best published results on the instances of Bredström and Rönnqvist (2008)

Inst.	Opt.	Heuristic Best	SA-ILS Best	CPU(s)	BKS	ABC-DA Best%	Avg%	CPU(s)	ABC-OBA Best%	Avg%	CPU(s)	ABC-RRT Best%	Avg%	CPU(s)
1S	3.55[a]	3.55	3.55	0.02	3.55	0.00	0.00	1.46	0.00	0.00	1.50	0.00	0.00	1.51
1M	3.55[b]	3.55	3.55	0.02	3.55	0.00	0.00	1.92	0.00	0.00	2.07	0.00	0.00	1.89
1L	3.39[b]	3.39	3.39	0.03	3.39	0.00	0.00	6.78	0.00	0.00	7.49	0.00	0.00	6.77
2S	4.27[a]	4.27	4.27	0.02	4.27	0.00	0.00	4.10	0.00	0.00	4.54	0.00	0.00	4.18
2M	3.58[b]	3.58	3.58	0.03	3.58	0.00	0.00	5.52	0.00	0.00	6.02	0.00	0.00	5.56
2L	3.42[b]	3.42	3.42	0.03	3.42	0.00	0.00	6.80	0.00	0.00	7.10	0.00	0.00	6.48
3S	3.63[a]	3.63	3.63	0.02	3.63	0.00	0.00	2.61	0.00	0.00	2.69	0.00	0.00	2.54
3M	3.33[b]	3.33	3.33	0.03	3.33	0.00	0.00	4.97	0.00	0.00	5.36	0.00	0.00	5.11
3L	3.29[b]	3.29	3.29	0.02	3.29	0.00	0.00	7.05	0.00	0.00	7.80	0.00	0.00	7.43
4S	6.14[a]	6.14	6.14	0.02	6.14	0.00	0.00	8.87	0.00	0.00	9.67	0.00	0.00	9.40
4M	5.67[b]	5.75	5.67	0.05	5.67	0.00	0.00	12.19	0.00	0.00	12.92	0.00	0.00	12.05
4L	5.13[b]	5.30	5.13	0.09	5.13	0.00	0.00	10.70	0.00	0.00	11.46	0.00	0.00	11.28
5S	3.93[a]	3.93	3.93	0.03	3.93	0.00	0.00	10.15	0.00	0.00	10.52	0.00	0.00	9.81
5M	3.53[a]	3.53	3.53	0.03	3.53	0.00	0.00	8.31	0.00	0.00	8.93	0.00	0.00	8.37
5L	3.34[b]	3.34	3.34	0.03	3.34	0.00	0.00	23.94	0.00	0.00	24.81	0.00	0.00	23.50
6S	8.14[b]	13.69	8.14	13.97	8.14	0.00	0.00	19.25	0.00	0.00	20.26	0.00	0.00	18.76
6M	–	12.80	7.70	26.68	7.70	0.00	0.00	31.13	0.00	0.00	31.88	0.00	0.00	32.14
6L	7.14[b]	11.87	7.14	15.86	7.14	0.00	0.00	46.86	0.00	0.00	49.78	0.00	0.00	48.05
7S	8.39[b]	15.06	8.39	15.08	8.39	0.00	0.00	63.18	0.00	0.48	66.11	0.00	0.24	63.40
7M	–	13.45	7.48	18.34	7.48	0.00	0.00	46.73	0.00	0.27	49.56	0.00	0.53	48.76
7L	–	11.52	6.88	15.92	6.88	0.00	0.29	20.19	0.00	0.29	20.86	0.00	0.73	20.37
8S	9.54[b]	–	9.54	25.13	9.54	0.00	0.31	64.16	0.00	0.00	70.89	0.00	0.42	64.48
8M	8.54[b]	–	8.54	15.01	8.54	0.00	0.35	37.54	0.00	0.59	58.50	0.00	0.12	53.57
8L	–	15.16	8.00	24.51	8.00	0.00	0.12	36.00	0.00	1.13	54.17	0.00	0.12	50.53
9S	–	–	11.93	150.52	11.93	0.00	0.25	218.29	0.00	0.25	221.00	0.00	0.08	228.14
9M	–	–	10.92	292.17	10.92	0.00	0.18	367.59	0.00	0.18	372.16	0.00	0.46	384.18
10S	–	16.24	8.60	16.10	8.60	0.00	0.23	155.91	0.00	0.70	168.00	0.00	0.58	173.43
Avg		*7.38*	*6.04*	*23.32*	*6.04*	*0.00*	*0.06*	*45.27*	*0.00*	*0.14*	*48.37*	*0.00*	*0.12*	*48.21*

[a] Results reported by Bredström and Rönnqvist (2008) executed on 2.67 MHz Xeon processor with 2 GB RAM
[b] Results reported by Afifi et al.(2016) the executed on Intel Xeon computer with 2.67 GHz

Table 5 clearly demonstrate that our hybrid algorithms can find all the best and optimal solutions in a reasonable computational time. The average deviation of ten runs from the best known (optimal) solution is 0.00% for the small set of instances of the data set, whereas the average deviation for the large set of instances is very small and equal to 0.06% (varying between 0.00% and 0.35%) for the hybrid ABC-DA, 0.14% (varying between 0.00% and 1.13%) for the hybrid ABC-OBA, and 0.12% (varying between 0.00% and 0.73%) for the hybrid ABC-RRT. This indicates that the hybrid algorithms are also stable in terms of finding high quality solutions in most of the runs. Overall, our algorithms are competitive and capable of obtaining good quality solutions.

6.3.2 Results on the HF-VRPS new instances

This section details the results of our algorithms on the HF-VRPS new instances. In each table in this section, we report the best and average solution values using our hybrid algorithms without SOP in columns "Best" and "Avg". The columns "Best+" and "Avg+" indicate the results obtained by each hybrid algorithm when using SOP. The column "BS" presents the best solution result found by any of the three developed algorithms (ABC-DA, ABC-OBA and ABC-RRT) using SOP for each corresponding instance. The columns denoted by "%" ("%"+) show the percent deviation of each of our algorithms without (with) SOP algorithm from the best solution "BS". The column "CPU" shows the average processing time in minutes without SOP.

We ran each algorithm on each instance five times. Table 6 reports the best results for the best run of our algorithms without and (with) SOP algorithm. Table 7 provides the average results of five runs.

Table 6
Comparison of our three algorithms with and without SOP algorithm on best results

Inst.	BS	ABC-DA					ABC-OBA					ABC-RRT				
		Best+	(%)+	Best	(%)	CPU (min)	Best+	(%)+	Best	(%)	CPU (min)	Best+	(%)+	Best	(%)	CPU (min)
A1	52.76	52.76	0.00	53.78	1.93	1.11	52.76	0.00	53.78	1.93	1.22	52.76	0.00	53.78	1.93	1.08
A2	152.21	152.21	0.00	155.41	2.11	3.45	152.21	0.00	155.42	2.11	3.36	152.21	0.00	155.41	2.11	3.38
A3	214.07	214.07	0.00	217.20	1.48	7.38	214.07	0.00	217.39	1.57	7.21	214.07	0.00	217.36	1.55	7.07
A4	401.88	402.93	0.26	409.34	1.87	11.70	403.06	0.29	409.84	1.97	12.64	402.97	0.27	409.25	1.84	11.92
Avg	*205.23*	*205.49*	*0.06*	*208.93*	*1.84*	*5.91*	*205.53*	*0.07*	*209.11*	*1.90*	*6.11*	*205.50*	*0.07*	*208.95*	*1.86*	*5.86*

Tables 6 and 7 (and their detailed versions on the website) indicate that our algorithms with SOP (columns "Best+" and "Avg+") are performing well in terms of obtaining best solutions for most of the instances. The ABC-DA, ABC-OBA and ABC-RRT algorithms are able to find the best solution at least once in the five runs for 31, 23 and 29 instances, respectively. The three algorithms were able to find the same best (average) solutions for 23 (11) out of 37 instances. Over the whole set of instances (last line in the tables), the average for the best run, was 0.06%(0.20%) for the ABC-DA, 0.07%(0.27%) for the ABC-OBA and 0.07%(0.22%) for the ABC-RRT. The processing time was also comparable for all algorithms with 5.91 minutes, 6.11 minutes, and 5.86 minutes for the three algorithms, respectively. It is also noted that SOP increases the processing time by less than 0.1 second.

Table 7
Comparison of our three algorithms with and without SOP algorithm on average results

Inst.	BS	ABC-DA				ABC-OBA				ABC-RRT			
		Avg+	(%)+	Avg	(%)	Avg+	(%)+	Avg	(%)	Avg+	(%)+	Avg	(%)
A1	52.76	52.76	0.00	53.81	1.98	52.76	0.00	53.81	1.98	52.76	0.00	53.81	1.98
A2	152.21	152.21	0.00	155.51	2.17	152.42	0.14	155.73	2.32	152.21	0.00	155.51	2.17
A3	214.07	214.70	0.29	217.70	1.71	214.89	0.38	217.96	1.83	214.64	0.27	217.82	1.76
A4	401.88	403.89	0.50	410.54	2.16	404.16	0.56	411.01	2.26	404.40	0.62	410.90	2.25
Avg	205.23	205.89	0.20	209.39	2.01	206.06	0.27	209.63	2.10	206.00	0.22	209.51	2.04

Tables 6 and 7 also show the versions of our algorithms without SOP algorithm in the columns "Best" and Avg". The average percentage for the best run for each algorithm over the whole instances was 1.84% (2.01%) for the hybrid ABC-DA, 1.90% (2.10%) for the hybrid ABC-OBA and 1.86% (2.04%) for the hybrid ABC-RRT. It is clear that the SOP improves the objective function of our hybrid algorithms and consequently the solution quality.

The results whether with or without SOP in general indicate that our algorithms benefited from the diversification and intensification strategies that we adopted in terms of both the processing speed as well as the solution quality.

For further evaluation of the benefit of varying the speed, we have chosen to test, for example, our ABC-DA algorithm with different values of constant speed on the best solution. In this experiment, the speed on all arcs was fixed at 50, 70 or 90 km/h. Table 8 (and its detailed results in the website) shows the results of this experiment. For example, the instance a-10-50-m1 contains ten vehicles and 50 visits with medium time windows. The last column of the table shows the results using the SOP algorithm.

Table 8

Effect of vehicle speed on solution quality

Inst.	50 km/h		70 km/h		90 km/h		BS
	Best	%	Best	%	Best	%	Best
A1	57.06	8.10	53.10	0.66	53.66	1.70	52.76
A2	165.42	8.77	153.51	0.86	155.07	1.88	152.21
A3	232.66	8.84	215.54	0.69	216.77	1.27	214.07
A4	434.63	7.87	405.84	0.73	409.03	1.52	402.93
Avg	*222.44*	*8.39*	*207.00*	*0.74*	*208.63*	*1.59*	*205.49*

Table 8 indicates that optimizing the vehicle's speed with our ABC-DA (last column) obtained the best results. On the other hand, fixing the speed at 90 km/h degrades the solution by only 1.59% on average, with a range between 0.85% and 2.27%. This is logical since when the driver's cost increases, it will be more economical to increase the speed in order to shorten the driving hours. In addition, we observe that fixing the speed at 50 km/h results in degradation of the solution by an average of 8.39%, with a range between 2.22% and 16.28%.

It is also important to evaluate the contribution of integrating the DA, OBA, and RRT algorithms with the ABC algorithm. Thus, we compared the hybrid methods with the simple (non-hybrid) DA, OBA, and RRT (using SOP algorithm). In addition, it is important to assess the importance of hybridization by comparing our methods against a standard (non-hybrid) ABC. The results of these experiments are reported in Tables 9 and 10, respectively. For the DA, OBA, and RRT algorithms, the parameters are summarized in Table 13 in Appendix.

Table 9

Comparison of ABC-DA, ABC-OBA, and ABC-RRT performance with standard DA, OBA, and RRT

Inst.	DA					OBA					RRT				
	Best+	%+	Avg+	%+	CPU (min)+	Best+	%+	Avg+	%+	CPU (min)+	Best+	%+	Avg+	%+	CPU (min)+
A1	52.76	0.00	52.76	0.00	0.24	52.76	0.00	52.76	0.00	0.36	52.76	0.00	52.76	0.00	0.25
A2	152.58	0.24	153.19	0.64	0.80	152.70	0.32	153.50	0.71	1.04	152.79	0.38	153.14	0.61	0.78
A3	215.51	0.67	216.91	1.03	1.81	215.60	0.71	217.36	1.15	2.09	215.51	0.67	216.85	1.03	1.66
A4	407.44	1.12	408.00	1.02	2.42	407.57	1.12	408.81	1.15	3.46	407.48	1.12	408.52	1.02	2.74
Avg	*207.07*	*0.51*	*207.72*	*0.67*	*1.32*	*207.16*	*0.54*	*208.11*	*0.75*	*1.74*	*207.14*	*0.54*	*207.82*	*0.66*	*1.36*

Table 9 shows the best (average) results obtained by each of our standalone algorithms. The columns "%" indicate the deviation from the best (average) solution value obtained by our DA, OBA and RRT algorithms, when compared with the hybrid algorithms. In Table 10, the columns "%ABC-DA", "%ABC-OBA" and "%ABC-RRT" present the percent deviation from the best solution that is found by the ABC from the best (average) solution obtained by our algorithms.

The results in Table 9 confirm that our methods outperform the standalone algorithms for both best and average solution quality. Specifically, the DA, OBA and RRT lag behind the hybrid ABC-DA, ABC-OBA, ABC-RRT by 0.51%, 0.54% and 0.54%, respectively. In addition, the average deviation values of the non-hybrid methods were 0.67%, 0.75% and 0.66%, respectively, when compared with their hybrid versions with the ABC. In addition, the detailed results of Table 9 in the website reveal that the hybrid strategies are considerably better in performance compared to the standalone methods, for the large sized instances with 80 to 140 patients (group *A*4). Our proposed hybrid methods ABC-DA, ABC-OBA and ABC-RRT improve the solutions with 1.12% compared to DA, OBA and RRT for the best run. In addition, the average deviation

from the average calculated for five runs was 1.02% for the DA (408.00 to 403.89), 1.15% for the OBA (408.81 to 404.16) and 1.02% for the RRT (408.52 to 404.40), in comparison with the hybrid methods. Furthermore, the standard DA, OBA and RRT algorithms could obtain only seven best solutions.

In Tables 9 and 10, we can see that the results of our standalone OBA algorithm and the hybrid ABC-OBA are slightly inferior to the other standalone and hybrid algorithms. This is probably due to the necessity of generating the appropriate parameter values for these two methods. In fact, the OBA method has many parameters that need careful tuning, which can affect the quality of solutions. Moreover, our DA(ABC-DA) and RRT(ABC-RRT) obtain similar results to each other.

Table 10
Comparison of ABC-DA, ABC-OBA, and ABC-RRT performance with ABC

Inst.				ABC					
	Best	%ABC-DA	%ABC-OBA	%ABC-RRT	Avg	%ABC-DA	%ABC-OBA	%ABC-RRT	CPU (min)
A1	52.76	0.00	0.00	0.00	52.79	0.06	0.06	0.06	0.06
A2	153.02	0.53	0.53	0.53	154.12	1.25	1.12	1.25	1.25
A3	216.57	1.17	1.17	1.17	218.26	1.66	1.57	1.69	1.69
A4	410.52	1.88	1.85	1.87	413.72	2.43	2.37	2.30	2.30
Avg	*208.22*	*1.33*	*1.31*	*1.32*	*209.72*	*1.86*	*1.78*	*1.81*	*1.81*

Similar to what was observed in Table 9, the results in Table 10 show that the hybrid methods are more efficient than the standard ABC algorithm. In other words, the standalone ABC has benefited from the integration of the DA, OBA or RRT algorithms.

6.3.3. Impact of using a heterogeneous fleet

This section investigates the impact of using a heterogeneous fleet of vehicles. The set of experiments in Table 11 includes the application of a heterogeneous fleet and a distinct vehicle type, namely small cars, medium cars and large cars only, that we have chosen to test in our ABC-DA algorithm. The column "Fuel%"("CO_2%") presents the deviation percentage from the number of liters of fuel consumed by the heterogeneous fleet (CO_2).

Table 11
The benefit of using a heterogeneous fleet of LDVs

Inst.	Only Small				Only Medium				Only Large				Heterogeneous	
	Fuel	Fuel%	CO_2	CO_2%	Fuel	Fuel%	CO_2	CO_2%	Fuel	Fuel%	CO_2	CO_2%	Fuel	CO_2
A1	6.72	-0.02	17.81	0.25	6.72	0.02	17.82	0.30	6.73	0.05	17.62	-0.81	6.72	17.77
A2	17.70	-0.05	46.90	0.25	17.72	0.06	46.95	0.37	17.73	0.14	46.45	-0.70	17.71	46.78
A3	26.06	-0.09	69.06	0.21	26.09	0.05	69.15	0.35	26.15	0.24	68.51	-0.60	26.08	68.91
A4	43.61	-0.18	115.56	0.09	43.75	0.13	115.93	0.40	43.85	0.37	114.89	-0.49	43.69	115.46
Avg	*23.52*	*-0.09*	*62.33*	*0.20*	*23.57*	*0.07*	*62.46*	*0.35*	*23.61*	*0.20*	*61.87*	*-0.65*	*23.55*	*62.23*

Table 11 shows that using a fleet of homogeneous small cars decreases the value of the consumed number of liters of fuel; while, compared to the scenario of using a heterogeneous fleet, the emitted CO_2 increases with 0.20%. On the other hand, the CO_2 emitted by the homogeneous large cars is reduced compared to the scenario with heterogeneous fleet with an average equal to 0.65%. Also, as shown in these results, our experimental structure recommends the use of medium size vehicles only in case the fleet that is applied is homogeneous. This is due to that the latter reduces both the fuel consumption as well as the CO_2 emission, compared to using homogeneous large cars and homogeneous small cars, respectively. As reported

in Table 11, low CO_2 solutions with reasonable fuel consumption are produced by a mixed fleet as well. In contrast to the scenario which applies only CVs, CO_2 can be diminished to 35.22% and 0.21% on the average total routing costs compared to the scenario of applying AFVs.

6.3.4. The impact of cost elements

In this experiment, the results of applying diversified cost constituents on the performance criteria are studied. We can note in Table 12 (and its details in the website) the best results of the five runs. Concerning each instance, we present the columns the total distance (DT), average fuel and CO_2 emissions cost (FC), driver cost (DC) and total cost (TC). Taking all instances into consideration, we can report that the driver cost makes up 83.94% of the total cost, while fuel and emissions costs amount to 16.06% of the total cost.

Table 12
Effect of cost components

Inst.	DT	FC	DC	TC
A1	100.65	9.41	43.35	52.76
A2	277.67	24.79	127.43	152.21
A3	376.42	36.52	177.56	214.07
A4	652.97	61.16	340.71	401.88
Avg	*351.93*	*32.97*	*172.26*	*205.23*

7. Conclusions

This paper introduces the HF-VRPS, where we assume a heterogeneous fleet of alternative fuel vehicles (i.e., passenger cars) composed of small, medium and large cars, instead of homogeneous conventional vehicles. We adopt the CMEM to estimate the fuel consumption and CO_2 emissions by considering biodiesel fuel for the vehicles instead of the traditional oil-diesel.

The main contribution of this work is developing such new rich variant of the VRP with synchronized visits that utilizes heterogeneous AFVs, due to their obvious benefits in terms of environmental impacts. In addition, we have presented a mathematical formulation for the problem, and developed three hybrid metaheuristic algorithms based on Artificial Bee Colony (ABC) for solving the HF-VRPS. We have done extensive computational experiments for tuning the parameters and for testing our algorithms on both benchmark data and newly generated instances of different sizes. On the VRPS benchmark instances, our proposed algorithms are comparable to the state-of-the-art algorithm in terms of solution quality, where we were able to find the optimal and best-known solutions for the VRPS. In addition, we have generated new test data of the HF-VRPS containing large size instances with up to 140 patients and 28 vehicles.

Overall, the results indicate reasonable run times for our algorithms as well as good scaling performance. The experimental results also confirm that the hybridization of the population-based (ABC) metaheuristic with the single-solution based methods (i.e., DA, OBA and RRT) has led to obtaining better solutions than using standalone methods (without hybridization), especially for large sized problems. We also believe that these proposed hybrid algorithms can be efficiently used to solve other complex VRP variants.

From a practical perspective, some interesting insights can be derived from the results of this study. First, we can observe that the total cost can be reduced more by using a heterogeneous fleet without speed optimization than using a homogeneous fleet with speed optimization. Second, it can be realized that optimizing the speed on each arc is not really essential, since it does not produce much better results than using a fixed speed. This is beneficial in practice, since it is easier for drivers to maintain a constant speed

during the entire trip, rather than attempting to change their speed on each segment of the trip. Finally, an important result of our study is that using a heterogeneous fleet is more advantageous than using a homogeneous one in this application, since it contributes to obtaining a tradeoff between the consumed fuel and emissions by considering two types of biodiesels blends (B20 and B5) for the different vehicles' types. This has an important implication in practice, since most companies do not usually operate just one type of vehicles, and they strive to minimize the cost of fuel consumption as well as CO_2 emission. Thus, based on the results of our study, not only homecare companies but also other profit and non-profit organizations that use conventional diesel vehicles should aim to convert their vehicles to AFVs using biodiesels blends (B20 and B5), since no modification in the diesel engines is required in this case. This solution can readily help to overcome the environmental effects and also fulfill new governmental regulations.

Finally, we should note that despite the richness of the studied variant in this paper, some limiting assumptions still exist. These include, considering specific vehicles' types and specific fuel blends in our model, which is needed to estimate the effect of fleet heterogeneity on fuel consumption and emissions. In addition, our model does not consider the possible need of refueling during the vehicle's journey. Future extensions may try to relax some of these assumptions. Other interesting perspectives include, for example, taking into consideration the consistency of the service providers for the patients based on their preferences, matching the specialties of the healthcare providers with the patients' needs, and extending the planning horizon to a multi-period case. In addition, different methodological variants can be attempted, such as exact methods (e.g., Branch-and-Cut, Branch-Cut-and-Price, etc.) for solving HF-VRPS instances of moderate size.

Acknowledgements

References

Administration on Aging (AoA). Available from: https://aoa.acl.gov/Aging_Statistics/Index.aspx.

Afifi, S., Dang, D. C., Moukrim, A., 2016. Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. *Optimization Letters*, *10*(3), 511-525.

Alahmadi, A. A., Alamri, T. M., Hosny, M. I., 2014. Time efficient demon algorithm for graph coloring with search cut-off property. In *Science and Information Conference (SAI), 2014* (pp. 254-259). IEEE.

Audi A8 L, 2017. Available from: http://www.baronsgroup.co.uk/audi/new-cars/a8.

Barth, M., Younglove, T., Scora, G., 2005. Development of a heavy-duty diesel modal emissions and fuel consumption model. *California Partners for Advanced Transit and Highways (PATH)*.

Beiter, P., Tian, T., 2016. *2015 Renewable Energy Data Book* (No. DOE/GO-102016-4904). National Renewable Energy Laboratory.

Bektaş, T., Laporte, G., 2011. The pollution-routing problem. *Transportation Research Part B: Methodological*, *45*(8), 1232-1250.

Benzarti, E., Sahin, E., Dallery, Y., 2013. Operations management applied to home care services: Analysis of the districting problem. *Decision Support Systems*, *55*(2), 587-598.

BMW 328d, 2017. Available from: https://www.autoblog.com/buy/2017-BMW-328d/.

Braekers, K., Caris, A., Janssens, G. K., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, *67*, 166-186.

Braekers, K., Hartl, R. F., Parragh, S. N., Tricoire, F., 2016. A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience. *European Journal of Operational Research*, *248*(2), 428-443.

Bräysy, O., Dullaert, W., Gendreau, M., 2004a. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*, *10*(6), 587-611.

Bräysy, O., Hasle, G., Dullaert, W., 2004b. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, *159*(3), 586-605.

Bredström, D., Rönnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European journal of operational research*, *191*(1), 19-31.

Cappanera, P., Scutellà, M. G., 2014. Joint assignment, scheduling, and routing models to home care optimization: a pattern-based approach. *Transportation Science*, *49*(4), 830-852.

Ceselli, A., Righini, G., Tresoldi, E., 2014. Combined location and routing problems for drug distribution. *Discrete Applied Mathematics*, *165*, 130-145.

Chandran, B., Golden, B., Wasil, E., 2003. A computational study of three demon algorithm variants for solving the traveling salesman problem. In *Computational Modeling and Problem Solving in the Networked World* (pp. 155-175). Springer US.

Cordeau, J. F., Laporte, G., Mercier, A., 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, *52*(8), 928-936.

Creutz, M., 1983. Microcanonical monte carlo simulation. *Physical Review Letters*, *50*(19), 1411.

Dabia, S., Demir, E., Woensel, T. V., 2016. An exact approach for a variant of the pollution-routing problem. *Transportation Science*, *51*(2), 607-628.

Decerle, J., Grunder, O., El Hassani, A. H., Barakat, O., 2017. A general model for the home health care routing and scheduling problem with route balancing. *IFAC-PapersOnLine*, *50*(1), 14662-14667.

Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346-359.

Demir, E., Bektaş, T., Laporte, G., 2014. A review of recent research on green road freight transportation. *European Journal of Operational Research*, *237*(3), 775-793.

Demir, E., Huang, Y., Scholts, S., Van Woensel, T., 2015. A selected review on the negative externalities of the freight transportation: Modeling and pricing. *Transportation Research Part E: Logistics and Transportation Review*, *77*, 95-114.

Derigs, U., Pullmann, M., Vogel, U., 2013. A short note on applying a simple LS/LNS-based metaheuristic to the rollon–rolloff vehicle routing problem. *Computers & Operations Research*, *40*(3), 867-872.

Dohn, A., Kolind, E., Clausen, J., 2009. The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. *Computers & Operations Research*, *36*(4), 1145-1157.

Dongarra, J., 2014. Performance of Various Computers Using Standard Linear Equations Software, (Linpack Benchmark Technical Report, CS-89-85). University of Tennessee, Computer Science Department.

Drexl, M., 2012. Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. *Transportation Science*, *46*(3), 297-316.

Dueck, G., 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics*, *104*(1), 86-92.

Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *48*(1), 100-114.

Eshtehadi, R., Fathian, M., Demir, E., 2017. Robust solutions to the pollution-routing problem with demand and travel time uncertainty. *Transportation Research Part D: Transport and Environment*, *51*, 351-363.

Eveborn, P., Flisberg, P., Rönnqvist, M., 2006. Laps Care—an operational system for staff planning of home care. *European Journal of Operational Research*, *171*(3), 962-976.

Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: A review. *Computers & Operations Research*, *77*, 86-95.

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., Stobbe, M., 2017. A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research*, *259*(3), 972-991.

Fukasawa, R., He, Q., Song, Y., 2016. A disjunctive convex programming approach to the pollution-routing problem. *Transportation Research Part B: Methodological*, *94*, 61-79.

Haddadene, S. R. A., Labadie, N., Prodhon, C., 2016. A GRASP× ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems with Applications*, *66*, 274-294.

Hu, T. C., Kahng, A. B., Tsao, C. W. A., 1995. Old bachelor acceptance: A new class of non-monotone threshold accepting methods. *ORSA Journal on Computing*, *7*(4), 417-425.

IMAD Genava, Available from URL: http://www.imad-ge.ch/fr/index.php

IMAD Genava, 2016. Activity report. Available from: http://www.imad- ge.ch/ data/documents/imad_20170620 _SCOM_EDI_rapport_annuel_ext_2016b.pdf

Juan, A. A., Goentzel, J., Bektaş, T., 2014. Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations?. *Applied Soft Computing*, *21*, 84-94.

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.

Kergosien, Y., Lenté, C., Billaut, J. C., 2009. Home health care problem: An extended multiple traveling salesman problem. In *4th multidisciplinary international conference on scheduling: Theory and applications (MISTA'09), Dublin (Irland)* (pp. 10-12).

Kıran, M. S., Fındık, O., 2015. A directed artificial bee colony algorithm. *Applied Soft Computing*, *26*, 454-462.

Kirkpatrick, S., 1984. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34 (5-6), 975-986.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2014. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, *70*, 239-254.

Kopfer, H. W., Schönberger, J., Kopfer, H., 2014. Reducing greenhouse gas emissions of a heterogeneous vehicle fleet. *Flexible Services and Manufacturing Journal*, *26*(1-2), 221-248.

Kwon, Y. J., Choi, Y. J., Lee, D. H., 2013. Heterogeneous fixed fleet vehicle routing considering carbon emission. *Transportation Research Part D: Transport and Environment*, *23*, 81-89.

Lari, I., Ricca, F., Scozzari, A., 2008. Comparing different metaheuristic approaches for the median path problem with bounded length. *European Journal of Operational Research*, *190*(3), 587-597.

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

Lin, S., 1965. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44, 2245–2269

Liu, R., Xie, X., Augusto, V., Rodriguez, C., 2013. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, *230*(3), 475-486.

Liu, Y. F., Liu, S. Y., 2013. A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem. *Applied Soft Computing*, *13*(3), 1459-1463.

Mahmudul, H. M., Hagos, F. Y., Mamat, R., Adam, A. A., Ishak, W. F. W., Alenezi, R., 2017. Production, characterization and performance of biodiesel as an alternative fuel in diesel engines–A review. *Renewable and Sustainable Energy Reviews*, *72*, 497-509.

Mankowska, D. S., Meisel, F., Bierwirth, C., 2014. The home health care routing and scheduling problem with interdependent services. *Health Care Management Science*, *17*(1), 15-30.

Marinakis, Y., & Marinaki, M., 2014. A bumble bees mating optimization algorithm for the open vehicle routing problem. *Swarm and Evolutionary Computation*, *15*, 80-94.

Masmoudi, M. A., Hosny, M., Braekers, K., Dammak, A., 2016. Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. *Transportation Research Part E: Logistics and Transportation Review*, *96*, 60-80.

Mercedes-Benz E250-bluetec, 2014. Available from: http://www.guideautoweb.com/specifications/mercedes-benz/Classe-e/e250-bluetec-4matic-berline/2014/.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., Herazo-Padilla, N., 2015. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, *79*, 115-129.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*(8), 2403-2435.

Potvin, J. Y., Rousseau, J. M., 1995. An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, *46*(12), 1433-1446.

Qian, J., Eglese, R., 2016. Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, *248*(3), 840-848.

Qiu, Y., Qiao, J., Pardalos, P. M., 2017. A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega*, *68*, 49-61.

Rasmussen, M. S., Justesen, T., Dohn, A., Larsen, J., 2012. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, *219*(3), 598-610.

Redjem, R., Marcon, E., 2016. Operations management in the home care services: a heuristic for the caregivers' routing problem. *Flexible Services and Manufacturing Journal*, *28*(1-2), 280-303.

Salas, Y. J. C., Pérez, C. A. M., Bello, R., Oliveira, A. C., Chaves, A. A., Lorena, L. A., 2015. Clustering search and variable mesh algorithms for continuous optimization. *Expert Systems with Applications*, *42*(2), 789-795.

Savelsbergh, M. W., 1992. The vehicle routing problem with time windows: Minimizing route duration. *ORSA journal on Computing*, *4*(2), 146-154.

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

Sivaramakrishnan, K., Ravikumar, P., 2011. Determination of higher heating value of biodiesels. *International Journal of Engineering Science and Technology*, *3*(11), 7981-7987

Szeto, W. Y., Wu, Y., Ho, S. C., 2011. An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, *215*(1), 126-135.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F., Potvin, J. Y., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, *31*(2), 170-186.

US EPA, Department of Energy, 2016. Fuel economy guide. Brochure prepared by the Office of Energy Efficiency and Renewable Energy. Available from https://www.fueleconomy.gov/feg/pdfs/guides/FEG2016.pdf. Last accessed on 26/02/2018.

US EPA, Department of Energy, 2017. Fuel economy guide. Brochure prepared by the Office of Energy Efficiency and Renewable Energy. Available from https://www.fueleconomy.gov/feg/pdfs/guides/FEG2017.pdf.

Verma, P., Sharma, M. P., 2016. Review of process parameters for biodiesel production from different feedstocks. *Renewable and Sustainable Energy Reviews*, *62*, 1063-1071.

Wei, L., Zhang, Z., Zhang, D., Lim, A., 2015. A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, *243*(3), 798-814.

Xue, J., Grift, T. E., Hansen, A. C., 2011. Effect of biodiesel on engine performances and emissions. *Renewable and sustainable energy reviews*, *15*(2), 1098-1116.

Yavuz, M., Oztaysi, B., Onar, S. C., Kahraman, C., 2015. Multi-criteria evaluation of alternative-fuel vehicles via a hierarchical hesitant fuzzy linguistic model. *Expert Systems with Applications*, *42*(5), 2835-2848.

Yildiz, A. R., 2013. A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing. *Applied Soft Computing*, *13*(5), 2906-2912.

Zhang , S., Lee, C. K. M., Choy, K. L., Ho, W., Ip, W. H., 2014. Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment*, *31*, 85-99.

Appendix

Table 13 presents all parameters and their values used in our algorithms.

Table 13

Parameters used in our proposed algorithms

Alg.	Description of parameters	Best value	Source
DA	Number of iterations (n_{DA})	50,000	Our experimental results in Table 2
	Initial Demon value (Dem)	1,500	Our experimental results in Table 2
	Number of iterations to restore the best solution when it is not improved (n_{div})	100	Inspired from the experimental results of Wei et al. (2015)
OBA	Number of iterations (n_{OBA})	50,000	Based on the experimental results of the parameter tuning of DA (Table 2)
	Threshold update granularity (Δ)	0.093	Experimental results of Hu et al. (1995)
	Multiplicative factor in growth rate (a)	1	Experimental results of Hu et al. (1995)
	Power-law in growth rate (b)	2	Experimental results of Hu et al. (1995)
	Coefficient in damping (c)	0.5	Experimental results of Hu et al. (1995)
	Number of iterations to restore the best solution when it is not improved (n_{div})	100	Inspired from the experiment results of Wei et al. (2015)
RRT	Number of iterations (n_{RRT})	50,000	Based on the experimental results of the parameter tuning of DA (Table 2)
	Deviation value (Dev)	0.01* current global Record	principal RRT algorithm of Dueck (1993)
	Number of iterations to restore the best solution when it is not improved (n_{div})	100	Inspired from the experiment results of Wei et al. (2015)
ABC	Number of iterations (n_{ABC})	No improvement of the best solution after ten consecutive iterations	Inspired from Masmoudi et al. (2016)
	Population size (P)	25	Based on the literature Karaboga (2005) and Szeto et al. (2011)
	Number of iteration of the current solution not has been improved ($Limit$)	$P * S$	Based on the literature Karaboga (2005) and Szeto et al. (2011)
ABC -DA	Number of iterations (n_{ABC})	No improvement of the best solution after ten consecutive iterations	Inspired from Masmoudi et al. (2016)
	Population size (P)	10	Our intuition and our experiment results in Table 3
	Number of iteration of the current solution not has been improved ($Limit$)	50	Our intuition and our experiment results in Table 3
	Initial temperature (T_{max})	100	Experimental results of Demir et al. (2012)
	Cooling rate (α)	0.999	Experimental results of Demir et al. (2012)
	Number of iterations (n_{DA})	1,000	Our intuition and following the principle of hybridization framework in

			Masmoudi et al. (2016)
	Initial Demon value (Dem)	1,500	Based on the experimental results of the parameter tuning of DA (Table 2)
	Number of iterations to restore the best solution when it is not improved (n_{div})	100	Inspired from the experiment results of Wei et al. (2015)
ABC - OBA	Number of iterations (n_{ABC})	No improvement of the best solution after ten consecutive iterations	Inspired from Masmoudi et al. (2016)
	Population size (P)	10	Our intuition and our experiment results in Table 3
	Number of iteration of the current solution that has not been improved ($Limit$)	50	Our intuition and our experiment results in Table 3
	Initial temperature (T_{max})	100	Experimental results of Demir et al. (2012)
	Cooling rate (α)	0.999	Experimental results of Demir et al. (2012)
	Number of iterations (n_{OBA})	1,000	Our intuition and following the principle of hybridization framework in Masmoudi et al. (2016)
	Threshold update granularity (Δ)	0.093	Experimental results of Hu et al. (1995)
	Multiplicative factor in growth rate (a)	1	Experimental results of Hu et al. (1995)
	Power-law in growth rate (b)	2	Experimental results of Hu et al. (1995)
	Coefficient in damping (c)	0.5	Experimental results of Hu et al. (1995)
	Number of iterations to restore the best solution when it is not improved (n_{div})	100	Inspired from the experiment results of Wei et al. (2015)
ABC - RRT	Number of iterations (n_{ABC})	No improvement of the best solution after ten consecutive iterations	Inspired from Masmoudi et al. (2016)
	Population size (P)	10	Our intuition and our experiment results in Table 3
	Number of iteration of the current solution not has been improved ($Limit$)	50	Our intuition and our experiment results in Table 3
	Initial temperature (T_{max})	100	Experimental results of Demir et al. (2012)
	Cooling rate (α)	0.999	Experimental results of Demir et al. (2012)
	Number of iterations (n_{OBA})	1,000	Our intuition and following the principle of hybridization framework in Masmoudi et al. (2016)
	Deviation value (Dev)	0.01* current global Record	Principal RRT algorithm of Dueck (1993)
	Number of iterations to restore the best solution when it is not improved (n_{div})	100	Inspired from the experiment results of Wei et al. (2015)