

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/125255/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Masmoudi, Mohamed Amine, Hosny, Manar, Demir, Emrah and Pesch, Erwin 2020. Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem. *Journal of Heuristics* 26 (1) , pp. 83-118. 10.1007/s10732-019-09424-x

Publishers page: <http://dx.doi.org/10.1007/s10732-019-09424-x>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Hybrid Adaptive Large Neighborhood Search Algorithm for the Mixed Fleet Heterogeneous Dial-a-Ride Problem

Mohamed Amine Masmoudi . Manar Hosny . Emrah Demir . Erwin Pesch

Abstract The mixed fleet heterogeneous dial-a-ride problem (MF-HDARP) consists of designing vehicle routes for a set of users by using a mixed fleet including both heterogeneous conventional and alternative fuel vehicles. In addition, a vehicle is allowed to refuel from a fuel station to eliminate the risk of running out of fuel during its service. We propose an efficient hybrid Adaptive Large Neighborhood Search (hybrid ALNS) algorithm for the MF-HDARP. The computational experiments show that the algorithm produces high quality solutions on our generated instances and on HDARP benchmarks instances. Computational experiments also highlight that the newest components added to the standard ALNS algorithm enhance intensification and diversification during the search process.

Keywords Dial-a-ride problem . Alternative fuel station . Adaptive Large Neighborhood Search algorithm . Mixed vehicle fleet

1. Introduction

Everyday millions of people travel to different locations using various public commuting services. Unfortunately, people who suffer from physical disabilities often do not benefit from public services due to accessibility and mobility complications. To improve these services, researchers in this field have introduced the reduced mobility transportation problem, which seeks to plan vehicle routes to improve the disabled persons' mobility. This problem is also known as the Dial-a-Ride Problem (DARP).

As a general practice, Conventional Vehicles (CVs) with unlimited fuel supply are considered in all DARPs (see e.g. Muelas et al., 2013; Braekers et al., 2014). However, CVs are among the main contributors to harmful emissions, such as greenhouse gases (GHGs) and air pollutants (U.S. EIA 2013). In an attempt to reduce the harmful environmental impacts and for saving the limited energy resources, many organizations today resort to incorporating Alternative Fuel Vehicles (AFVs) in their fleet, including flexible fuel vehicles and fuel cell vehicles (US DOE, 2018). AFVs operate on different types of alternative fuels, such as biodiesel, propane, ethanol, and hydrogen. As an example from practice, in Stockholm City, the transport of persons is conducted by heterogeneous AFVs using different fuel types (Ethanol ED95, biogas, and biodiesels). Moreover, in Canada, several companies such as Société de Transport de Montréal (STM) in Montréal and the Réseau de Transport de la Capitale (RTC) in Québec, use different types of AFVs added to their existing CVs' fleet.

From a research perspective, the use of AFVs has recently attracted attention in the field of the Vehicle Routing Problem (VRP) (see, e.g., Erdoğan and Miller-Hooks, 2012; Adler and Mirchandani, 2016; Andelmin and Bartolini, 2017; Yavuz, 2017), giving rise to a new VRP variant known as the Green Vehicle Routing Problem (GVRP) with refueling, since refueling of AFVs during their service route is a main concern as will be explained shortly. However, to our knowledge, the use of AFVs in the DARP applications has not been previously considered in the literature. Thus, this paper promotes using a mixed fleet of CVs and AFVs with tank refueling within the context of the DARP.

AFVs can be classified into two main categories: dedicated AFVs and dual-fuel AFVs. Dedicated AFVs use only alternative fuel, such as CNG and propane. On the other hand, dual-fuel AFVs come into two main types: bi-fuel and flexible-fuel. Bi-fuel vehicles can operate with either alternative or conventional fuel; i.e., a special tank and fuel system is provided for each of these types, but the vehicle can operate on only one of them at a time. In contrast, a flexible-fuel vehicle has one fuel tank that can be filled with either type of fuel. The most common fuel used in flexible-fuel vehicles is a blend of gasoline and ethanol (DVRPC, 2011). In our case, we adopt a fleet of flexible-fuel vehicles (i.e., a fleet of alternative fuel vehicle using biodiesel).

One important point to note when using AFVs in VRP applications, though, is that the amount of alternative fuel in the vehicle tank is limited, in contrast to the traditional gasoline or diesel fuel, where the amount of fuel in the vehicle tank is assumed to be enough to travel for longer distances. In addition, the traditional gasoline or diesel refueling stations are usually plentiful, while Alternative Fuel Stations (AFSs) are usually scarce and often unevenly distributed across urban areas (Erdoğan and Miller-Hooks, 2012; Yavuz, 2017). Therefore, the main difference between the GVRP with refueling and the traditional VRP is the consideration of refueling requirement during route planning. Failing to address this issue, may cause vehicles to run out of fuel or may cause unnecessary detours from the pre-planned routes to reach an AFS (Erdoğan and Miller-Hooks, 2012). Thus, many papers in the GVRP literature have considered the limitation in the driving range of AFVs as well as the need for refueling in specialized stations (see, e.g., Erdoğan and Miller-Hooks, 2012; Koç and Karaoglan, 2015). In these papers, the aim is efficient route planning that considers both customers' visits as well as frequent visits to refueling stations during the planning period. Our research is in line with these studied GVRP variants, where we incorporate AFVs as well as refueling stations in the route for serving customers in the DARP.

However, in the context of the DARP, the refueling requirement may cause service disruptions and may lead to the dissatisfaction of customers. Therefore, careful planning of these visits should be considered with this need in mind. We also note that the concept of refueling in the field of transportation of the disabled and the elderly is indeed applied in practice. For example, the company "Transport Adapté du Québec Métro Inc. (TAQM)" in Quebec, Canada (Thériault, 2005) offers services for the exclusive use of people with limited mobility. For this company, refueling of vehicles is an important requirement, due to the long distances frequently traveled to their destinations in different regions (e.g., health centers, special education institutions, working institutions, etc.). As a

result, it is mandatory to refuel the vehicles at the start of the day, during the trip, or at the end of the working day. The order of the customers' visits and the choice of an access point for each AFS are highly affected by an inappropriate planning. For example, a vehicle may frequently spend time during its journey for searching the nearest AFS due to restricted refueling infrastructure. Planning efficient routes that satisfy the customers' demands is one of the main concerns of TAQM. Therefore, it is necessary for the company to minimize the distance needed to reach AFS, while complying with the time constraints of each user.

Regarding the fuel consumption of AFVs, there are several studies concerning emission (fuel consumption) models (e.g., Demir et al., 2014) that report the major effect of the vehicle's type on fuel consumption. Thus, to enrich our research, we consider two types of emission models to calculate the fuel consumption rate: the Comprehensive Modal Emissions Model (CMEM) of Barth and Boriboonsomsin (2009) for the CVs and National Atmospheric Emissions Inventory (NAEI) model (NAEI, 2012) for the AFVs.

In addition, the considered mixed fleet in our problem is heterogeneous in terms of their capacity of carrying people (i.e., they are vehicles with different capacity resources like passenger seats, stretchers and wheelchairs). Thus, the problem that we consider belongs to the Heterogeneous Dial-a-Ride Problem (HDARP) with CVs' category as studied by Braekers et al. (2014), and Masmoudi et al. (2016, 2017). We call this specific problem as the Mixed Fleet HDARP (MF-HDARP).

2. Literature review

This section presents a brief literature review related to our problem. First, we review the most recent studies in the HDARP. Second, green vehicle routing problem papers that explicitly consider alternative fuel vehicles with refueling are presented. Third, VRPs that consider a mixed fleet of vehicles are reviewed. Finally, we discuss the main differences between our problem and the related studies followed by the main scientific contributions of our research.

2.1. The heterogeneous dial-a-ride problem

The reduced mobility people transportation is often complicated by the presence of several types of users with special needed equipment, such as a patient seat, a wheelchair and a stretcher (Wong and Bell 2006). The DARP with heterogeneous users and/or vehicles (called HDARP) (Parragh, 2011) is a generalization of the DARP, but it has not been extensively studied in the literature. Wong and Bell (2006) tackled the DARP with two types of vehicles (equipped with wheelchairs) and two types of users for elderly and disabled people's transportation. Xiang et al. (2006) developed a heuristic algorithm to solve a more practical version of the DARP with several types of users and vehicles. In a later study, Parragh et al. (2012) introduced a variant of the HDARP, in which the requirements of assistants and their lunch break constraints are considered.

In the study of Braekers et al. (2014), multiple depots of heterogeneous vehicles and users are considered to reduce the total routing costs. More recently, Masmoudi et al. (2017) solved the standard HDARP using a hybrid Genetic Algorithm (GA). To our knowledge, their hybrid GA provides the

best-known results on these instances so far, and outperforms current state-of-the-art algorithms for the standard DARP and HDARP. In another study, Masmoudi et al. (2016) augmented the multi depots and coffee break concepts on the standard HDARP. The authors developed two hybrid metaheuristic approaches, namely hybrid Bees Algorithm with Simulated Annealing (BA-SA) and hybrid Bees Algorithm with Deterministic Annealing (BA-DA), as well as ALNS algorithm.

Later, Braekers and Kovacs (2016) extended the single period HDARP to a multi-period DARP and considered a limited number of drivers to serve each user over a predefined number of days. More recently, Masmoudi et al. (2018b) proposed a new DARP variant by considering a fleet of homogeneous Electric Vehicles (EVs) instead of the CVs. In their problem, the EVs are allowed to be recharged by swapping their depleted battery by a full one from any battery-swap station. To solve this problem, three Evolutionary Variable Neighborhood Search (EVO-VNS) variants are proposed. In their methods, the VNS is embedded with some features adopted from population-based methods, such as crossover of the GA, to diversify the search, and the Shuffled Frog-Leaping Algorithm (SFLA) to create the initial solution for each VNS iteration. The proposed algorithms are compared on new randomly generated instances with up to eight vehicles and 96 requests. These instances are based on the benchmark HDARP instances of Masmoudi et al. (2017) and on an artificial data set with different characteristics, containing up to 15 vehicles and 100 users. The results show that the proposed approaches provide high quality solutions on the new generated instances. In addition, they demonstrate that the hybridization of several features of population based methods with VNS outperforms the traditional VNS. To the best of our knowledge, this is the only study that incorporates a fleet of EVs in the DARP.

For other variants of (H)DARP, interested readers can find several real concepts related to this problem in the applications studied by Zhang et al. (2015), Liu et al.(2015), Lim et al. (2016), and Amirgholy and Gonzales (2016). Interested readers are also referred to surveys on the DARP by Molenbruch et al. (2017) and Ho et al. (2018).

2.2. The green vehicle routing problem with refueling

The problem presented in this research is related to alternative fuel vehicles, fuel stations, and green vehicle routing problems with refueling using a limited fuel tank.

During the last few years, research in the field of logistics and operations research has been extended by considering environmental impacts and costs related to both people and industrial transportation activities. Within this domain, the Green Vehicle Routing Problem (GVRP), which considers the fuel tank capacity limitation, has received an increasing attention recently. Erdoğan and Miller-Hooks (2012) were the first authors to introduce the GVRP, where refueling and a fleet of biodiesel-powered alternative fuel vehicles are considered. A constant fuel consumption rate is used in order to decide when the vehicle should be refueled. The authors proposed a mixed-integer linear model to minimize the travel distance considering AFSs as well as a finite driving autonomy, where the number of tours and their limited duration are respected. A constant fuel consumption is considered in Koç and Karaoglan (2015). They suggested a mixed integer programming formulation

and proposed a more sophisticated branch-and-cut algorithm, in which various adequate inequalities taken from the literature are incorporated, in order to improve the lower bound. They also used a simulated annealing metaheuristic to acquire the upper bound. The algorithm was evaluated by testing it on benchmark GVRP instances of Erdoğan and Miller-Hooks (2012). The results indicate that there is a possibility to optimally solve 22 out of 40 instances with 20 customers during a short computation time. An exact solution approach based on a set partitioning formulation by adding a new valid inequality is proposed by Andelmin and Bartolini (2017) to solve the GVRP. Yavuz (2017) developed an Iterated Beam Search (IBS) algorithm for the GVRP with refueling of a homogeneous service fleet by allowing multiple AFS visits. Other studies on the GVRP with refueling can be found in Adler and Mirchandani (2016) and Xiao and Konak (2017). Interested readers are referred to the survey paper of Demir (2018) for GVRP varieties.

2.3. Mixed fleet vehicle routing problem

In recent years, using a mixed fleet of vehicles in different VRP variants has attracted the attention of researchers, since it is more practical and relevant. However these studies are still limited in the literature and only few works have addressed this concept. Sassi et al. (2014) used a mixed fleet of vehicles composed of EVs and CVs in the context of Electric Vehicle Routing Problem (EVRP), which is considered as an extension of the GVRP (Schneider et al., 2014). Goeke and Schneider (2015) considered also a mixed fleet of EVs and CVs for the EVRP. The authors developed a realistic energy consumption model for the EVs based on the CMEM of Barth and Boriboonsomsin (2009), designed for the traditional CVs. ALNS method is proposed to solve this problem. The experiments show that this method is able to find good results on the proposed problem and on the benchmark VRP with time windows and the EVRP.

More recently, Hiermann et al. (2019) proposed a new EVRP variant called Hybrid Heterogeneous Electric Fleet routing problem with Time Windows and recharging stations (H^2E -FTW), where they consider a mixed fleet of vehicles, composed of Battery Electric Vehicles (BEVs), Plug-in Hybrid Vehicles (PHEV) and CVs, as well as multiple vehicle types from each class with different battery sizes, capacity and fuel consumption and/or electric energy per mile. To solve this problem, the authors proposed an efficient sophisticated hybrid genetic algorithm based on layered route evaluation procedures. The algorithm was tested on a variety of benchmark instances of the E-FSMFTW (Electric Fleet Size and Mix vehicle routing problem with Time Windows and recharging stations) and the E-VRPTW and was able to obtain better or equal average results than existing algorithms on these problems. In addition, 119, 11, and 19 new best solutions were found for the E-FSMF, the E-VRPTW, and the E-VRPTW with partial recharging. The authors also investigated how fuel and energy cost can impact the decision regarding fleet composition. They concluded that a mixed fleet can reduce costs in most operational scenarios, in comparison to the use of a single vehicle type.

2.4. Summary and discussion

The different characteristics of the MF-HDARP can be frequently encountered in practice. Our MF-HDARP is both similar to and different from HDARP, mixed fleet VRP, and GVRP, as explained next.

First, based on the HDARP literature summarized above, we can observe that using a fleet of CVs (Parragh, 2011; Braekers et al., 2014; Masmoudi et al., 2016, 2017), or using EVs (Masmoudi et al., 2018b) is considered separately. However, in real-world applications most companies operate different modes of transportations. Thus, in our research we adopt a mixed fleet of CVs and AFVs with tank refueling.

Second, there is a limited number of studies of DARP that use EVs (DARP-EV). Specifically, we are only aware of one paper (Masmoudi et al., 2018b) that studies this problem. In addition, this study has some limitations in terms of using only one mode of transportation, since the fleet is composed of homogeneous EVs (i.e., all vehicles have the same type of resources). On the other hand, our research is different than the DARP-EV of Masmoudi et al. (2018b) in that we use both a mixed fleet of CVs and AFVs, and the fleet has different capacity resources (i.e., heterogeneous fleet). Thus our problem is more complex than the traditional DARP-EV. It is also worth mentioning that using AFVs with fuel tank instead of EVs is beneficial for companies that operate CVs with gasoline or diesel. This is because biodiesel is an alternative fuel that can be adopted in conventional engines that use diesel, either separately or by blending it with diesel (Verma and Sharma, 2016). Moreover, using biodiesel in either form does not require any engine adjustment of the CV (Masmoudi et al., 2018a). Thus, newly manufactured diesel-powered vehicles can run on biodiesel without any alterations or special requirements. This allows logistics companies to transform some of their CVs to AFVs using biodiesels, without having to replace their fleets.

Third, using a mixed fleet of vehicles is very limited in the literature and applied only in some EVRP variants (Sassi et al., 2014; Goeke and Schneider, 2015; and Hiermann et al., 2019). It has not been applied in any DARP or HDARP variants. Thus, we believe that it deserves to be studied in the context HDARP. Moreover, as previously mentioned, our research is different than the existing mixed fleet applications in that we use different capacity resources inside the vehicle, instead of only one resource type, as in the majority of mixed fleet vehicle routing problems. Finally, most studied GVRP with tank refueling (see, e.g., Erdoğan and Miller-Hooks, 2012; Koç and Karaoglan, 2015; Andelmin and Bartolini, 2017; Yavuz, 2017) use a constant fuel consumption rate. However, in recent studies of the EVRP and the Pollution Routing Problem (PRP), the fuel(energy) consumption rate is not linear and depends on several factors, such as the speed, load,...etc (see, e.g., Demir et al., 2012, 2014; Franceschetti et al., 2017; Androutsopoulos and Zografos, 2017; Toro et al., 2017; Salehi et al., 2017). Similar to the mentioned works, we apply a fuel consumption rate function. Moreover, to enrich our research, we consider two emission models to calculate the fuel consumption rate: the CMEM and National Atmospheric Emissions Inventory (NAEI) (NAEI, 2012) model. The CMEM model is applied to CVs (Demir et al., 2012), while the NAEI model is implemented on the AFVs. In fact, the

main advantage of using the NEAI in our research, especially for the AFVs, is that the NAEI model is calculated based on both total fuel consumption data as well as fuel properties. Moreover, since it is obvious that fuel consumption is the origin of CO₂e (Demir et al., 2012), the amount of fuel consumption can be immediately converted into that of CO₂e through multiplying it with a certain coefficient. To the best of our knowledge, these two models have not been applied on any DARP variant yet, as well as in the EVRP and the mixed fleet vehicle routing problem.

To sum up, our MF-HDARP is considered as a combination of several aspects from the existing HDARP, mixed fleet vehicle routing problem and the GVRP with refueling. To our knowledge, this rich problem variant has not been previously tackled in the literature.

2.5. The main scientific contributions and structure of the paper

The contributions of this research are as follows: *i)* we introduce a Mixed Fleet Heterogeneous Dial a ride Problem (MF-HDARP) and provide a mathematical formulation of the purposed problem. *ii)* A hybrid Adaptive Large Neighborhood Search (ALNS) algorithm is developed to solve the MF-HDARP. The motivation for adopting ALNS for solving the MF-HDARP is its successful performance on related vehicle routing problems including (H-)DARP, in addition to its robustness in efficiently solving problem instances with different characteristics. *iii)* We hybridize the traditional ALNS with several diversification and intensification mechanisms to improve its performance. Specifically, we add an exploration mechanism to avoid local optima using crossover operators, and an intensification mechanism using a local search procedure. In addition, several special characteristics and algorithmic improvements have been developed to achieve good performance on the MF-HDARP, as will be discussed later. *iv)* We introduce a new large size data set instances with different characteristics having up to 200 requests. *v)* Extensive computational experiments show that our algorithm is able to produce good-quality solutions, on both existing and new benchmark instances. And finally, *vi)* we assess the effect of hybridizing the ALNS with the different new components (i.e., crossover and local search operators), and draw some insights on the performance of these components.

The rest of the paper is organized as follows. Section 3 provides the problem definition. Section 4 describes our proposed algorithm, and Section 5 reports the computational experiments. Conclusions are summarized in Section 6.

3. Problem definition

The MF-HDARP can be formally described as follows. Consider a graph $G = (V', A)$ with node set V' and arc set $= \{(i, j): i, j \in V', i \neq j\}$ where V' is further partitioned into subsets N and F' ($V' = N \cup F'$); $N = \{1, \dots, 2n\}$ corresponds to n users to be serviced, where $P = \{1, \dots, n\}$ and $D = \{n + 1, \dots, 2n\}$ are the sets of nodes corresponding to pickup and delivery locations, respectively. Let F the set of refueling stations. F' is the set of vertices in F . The depot is a special node that belongs to the set F' . It is assumed that the depot is a refueling station, where vehicle routes must start and end. In

addition, the depot node is duplicated, where the starting node is denoted by d and the ending node is denoted by e . There is a non-negative travel cost c_{ij} , travel speed v_{ij} and a non-negative distance d_{ij} associated with each arc (i, j) from set A . We assume that vehicles travel each arc (i, j) with different speeds between v^l and v^w , and the number of stops that can be made for refueling is unlimited. When refueling takes place, it is assumed that the tank is refilled to its maximum capacity. The time window to visit any refueling node is set as $[0, T]$, where T is the length of the planning horizon. Moreover, a mixed fleet with a fixed size of heterogeneous vehicles, which is composed of m_{AF} AFVs and m_{CV} CVs, are available to serve the n users.

Each CV(AFV) has a capacity $Q^{r,CV}(Q^{r,AFV})$ that gives the amount of resource r available on each CV(AFS), where each type of resource is dedicated to: the accompanying person of the handicapped $Q^{0,k}$, handicapped person's seat $Q^{1,CV}(Q^{1,AFV})$, stretcher $Q^{2,CV}(Q^{2,AFV})$ and wheelchair $Q^{3,CV}(Q^{3,AFV})$. Each CV(AFV) contains a fuel tank capacity $H^{CV}(H^{AFV})$, which is consumed and reduced at a fuel rate FR on each traveled arc (i, j) . Each user is associated with a demand requirement q_i^r for each resource r , and time window $[T_i^-, T_i^+]$, where T_i^- and T_i^+ represent the earliest and latest visiting time, respectively. A maximum user/patient ride time L_{max} is implicitly considered to provide the highest service quality. In addition, a service time s_i is imposed when visiting each node ($\forall i \in N$), and a refueling time s_f is considered when the AFV visits a refueling station node ($\forall f \in F'$).

As studied in Demir et al. (2012), for the fuel consumption rate of CVs, the constant fuel rate FR required over the course in each arc (i, j) can be calculated as: $FR_{ij} = \frac{\xi}{\kappa \cdot \lambda} (eND + \frac{P_{ij}}{\eta \cdot \eta_f})$, where P_{ij} represents the mechanical power $P_{ij} = (\frac{1}{2} c_d u A v^2 + m g (\sin(\alpha_{ij}) + c_r \cos(\alpha_{ij}))) v_{ij}$. All parameters along with typical values are summarized in Table12 in the Appendix.

For AFVs, the fuel consumption rate with an average speed v can be calculated as: $FR_{ij}(v) = \rho(\sigma_1 + \sigma_2 v + \sigma_3 v^2 + \sigma_4 v^3 + \sigma_5 v^4 + \sigma_6 v^5 + \sigma_7 v^6)/v$, where $\rho = 0.037$, $\sigma_1 = 10537.515$, $\sigma_2 = 220.217$, $\sigma_3 = 54.175$, $\sigma_4 = -2.404$, $\sigma_5 = 0.043$, $\sigma_6 = 0$ and $\sigma_7 = 0$. A detailed explanation of each parameter can be found on NAEI (2012).

Based on the heterogeneous dial a ride problem formulation of Parragh (2011) and the electric vehicle routing problem with mixed fleet formulation of Goeke and Schneider (2015), we provide below a mixed-integer programming formulation for the MF-HDARP. The MF-HDARP can be formulated as follows: x_{ij}^{CV} is a binary variable that is equal to 1 if arc (i, j) is traveled by the CV and 0 otherwise. Similarly, x_{ij}^{AFV} is a binary variable that is equal to 1 if the arc (i, j) is traveled by the AFV and 0 otherwise. B_i is a continuous variable that represents the time when the service starts at node i . The continuous variable $Q_i^{r,CV}(Q_i^{r,AFV})$ represents the load of resource r on the CV(AFV), immediately after visiting node i . The continuous variables l_i represents the ride time of user $i \in P$ on any vehicle (CV and AFV). Finally, the continuous variable o_i represents the tank fuel level of the AFV, when visiting node i .

$$\text{minimize } \sum_{(i,j) \in A} c_{ij} (x_{ij}^{AF} + x_{ij}^{CV}) \quad (1)$$

subject to

$$\sum_{j \in N} (x_{ij}^{AF} + x_{ij}^{CV}) = 1 \quad \forall i \in P \quad (2)$$

$$\sum_{j \in V'} x_{ij}^{AF} - \sum_{j \in V'} x_{n+i,j}^{AF} = 0 \quad \forall i \in P \quad (3)$$

$$\sum_{j \in V} x_{ij}^{CV} - \sum_{j \in V} x_{n+i,i}^{CV} = 0 \quad \forall i \in P \quad (4)$$

$$\sum_{j \in V'} x_{ij}^{AF} \leq 1 \quad \forall i \in F' \quad (5)$$

$$\sum_{j \in V \setminus \{d\}} x_{ij}^{AF} - \sum_{j \in V \setminus \{e\}} x_{ji}^{AF} = 0 \quad \forall i \in V' \quad (6)$$

$$\sum_{j \in V \setminus \{d\}} x_{ij}^{CV} - \sum_{j \in V \setminus \{e\}} x_{ji}^{CV} = 0 \quad \forall i \in V \quad (7)$$

$$\sum_{j \in V'} x_{dj}^{AF} \leq m_{AF} \quad (8)$$

$$\sum_{j \in V} x_{dj}^{CV} \leq m_{CV} \quad (9)$$

$$\sum_{i \in V', i \neq d} x_{di}^{AF} = \sum_{j \in V', j \neq d} x_{je}^{AF} \quad (10)$$

$$\sum_{i \in V', i \neq d} x_{di}^{CV} = \sum_{j \in V', j \neq e} x_{je}^{CV} \quad (11)$$

$$Q_j^{rAF} \geq q_j^r x_{ij}^{AF} + Q_i^{rAF} - Q^{r,CV} (1 - x_{ij}^{AF}) \quad \forall i \in V \setminus \{d, e\}, \forall j \in V \setminus \{d\}, r \in \{0, 1, 2, 3\} \quad (12)$$

$$Q_j^{rCV} \geq q_j^r x_{ij}^{CV} + Q_i^{rCV} - Q^{r,CV} (1 - x_{ij}^{CV}) \quad \forall i \in V \setminus \{d, e\}, \forall j \in V \setminus \{d\}, r \in \{0, 1, 2, 3\} \quad (13)$$

$$0 \leq Q_j^{rAF} \leq Q^{r,AF} \quad \forall j \in V', r \in \{0, 1, 2, 3\} \quad (14)$$

$$0 \leq Q_j^{rCV} \leq Q^{r,CV} \quad \forall j \in V, r \in \{0, 1, 2, 3\} \quad (15)$$

$$Q_d^{rAF} = 0 \quad r \in \{0, 1, 2, 3\} \quad (16)$$

$$Q_d^{rCV} = 0 \quad r \in \{0, 1, 2, 3\} \quad (17)$$

$$B_j \geq B_i + (s_i + t_{ij}) x_{ij}^{AF} - (1 - x_{ij}^{AF}) \quad \forall i \in N, j \in V' \quad (18)$$

$$B_j \geq B_i + (s_i + t_{ij}) x_{ij}^{CV} - (1 - x_{ij}^{CV}) \quad \forall i \in N, j \in V \quad (19)$$

$$l_i = B_{n+i} - (B_i + s_i) \quad \forall i \in P \quad (20)$$

$$t_{i,n+i} \leq l_i \leq L_{\max} \quad \forall i \in P \quad (21)$$

$$T_i^- \leq B_i \leq T_i^+ \quad \forall i \in V' \quad (22)$$

$$o_j \leq o_i - FRd_{ij} x_{ij}^{AF} + H(1 - x_{ij}^{AF}) \quad \forall j \in N, \forall i \in V', i \neq j \quad (23)$$

$$o_j \geq \min\{FRd_{jd}, FR(d_{ji} + d_{id})\} \quad \forall j \in N, \forall i \in F' \quad (24)$$

$$o_j = H^{AF} \quad \forall j \in F' \quad (25)$$

$$B_e - B_d \leq T_{\max} \quad (26)$$

$$x_{ij}^{AF}, x_{ij}^{CV} \in \{0, 1\} \quad \forall k \in K, \forall i, j \in V', i \neq j. \quad (27)$$

The objective function (1) minimizes the total routing costs. Constraints (2)-(4) guarantee that each user is served exactly once and each pair of pickup and delivery is served by the same vehicle, while constraints (5) ensure that each recharging station can be used at most once. Constraints (6) and (7) define the arc flow conservation. Constraints (8) and (9) guarantee that the number used of alternative fuel vehicles and conventional vehicles, respectively, does not exceed the fleet size. Constraints (10) and (11) guarantee that each vehicle begins at the origin depot and finishes at the corresponding destination depot. Constraints (12)-(15) enforces the capacity condition. Constraints (16) and (17) ensure that a vehicle has an empty load when leaving the depot. Constraints (18) and

(19) specify the beginning of service at each node. Constraints (20) define the ride time of each user in each route, which is bounded by constraint (21). These constraints also ensure the precedence constraint between the pickup and the corresponding drop off nodes. Constraints (22) enforce the time windows. Constraints (23) keep track of the fuel level of the alternative fuel vehicle, which is determined by the sequence and type of visited nodes. That is, if i is a customer node and j is visited immediately after i ($x_{ij}^{AF} = 1$), the first term in constraints (23) will guarantee that the fuel level is reduced by a sufficient amount, when the alternative fuel vehicle arrives at j . The reduction in fuel level is based on the distance from i to j and the fuel consumption rate. Constraints (24) guarantee that the alternative fuel vehicle will not get stuck after visiting any customer in the route due to shortage in fuel. This is done by ensuring that there is enough fuel remaining to drive to the depot, either directly or by passing through a refueling station. Constraints (25) guarantee that the tank becomes full after visiting a refueling station. Constraints (26) enforce the time limit of the route, which is restricted by T_{max} . Finally, constraints (27) specify the binary decision variables.

The HDARP is an NP-hard problem (Parragh, 2011). Several researchers have attempted exact methods (e.g., Branch-and-Cut, Branch-and-Price-and-Cut) to solve small-sized instances to optimality, since commercial solvers cannot solve instances as small as 10 requests (Zhang et al. 2015; Liu et al. 2015). For example, in Masmoudi et al. (2018b), CPLEX 12.6.1 can only solve very few small size instances with two vehicles and 15 requests. However, even the exact methods developed in the literature can only solve few small instances to optimality in the majority of studied problems (Molenbruch et al. 2017). Therefore, most HDARP studies develop metaheuristic approaches to solve this problem (Breakers et al., 2014; Braekers and Kovacs, 2016; Masmoudi et al., 2016, 2017, 2018b). Also, the Electric Vehicle Routing Problem using Mixed Fleet (E-VRPMF) is an NP-hard problem (Goeke and Schneider, 2015). Again, most studies have developed metaheuristics to solve this problem (Sassi et al., 2014; Goeke and Schneider, 2015; Hiermann et al., 2019). Since the MF-HDARP is a combination of the traditional (H)DARP and the E-VRPMF, solving practical size instances of this problem requires utilizing heuristic and metaheuristic approaches, in order to provide acceptable solutions within reasonable processing times. In the next section our proposed metaheuristic approach for solving the MF-HDARP is introduced.

4. Hybrid Adaptive Large Neighborhood Search algorithm for the MF-HDARP

ALNS was used in solving a variety of VRPs including (H)DARP (see, e.g., Ropke and Pisinger 2006a; Demir et al. 2012, 2014; Masmoudi et al., 2016; Žulj et al., 2018; Alinaghian and Shokouhi, 2018). However, when ALNS is applied to highly constrained problems, it may get trapped in local optima. Thus, we try here to improve the convergence of ALNS towards better solutions, by applying different intensification strategies around good solutions, and also encouraging diversification to unexplored regions of the search space.

In most of the ALNS algorithms applied in the literature, if the newly generated solution (after applying the removal and insertion operators), is not better than the current one, or is not accepted by

the well-known acceptance function of the SA algorithm, the ALNS restarts from a new solution that is re-generated using the removal and insertion operators on the same current solution. Nevertheless, in our approach, we do not retract to a formerly obtained solution. Instead, we construct a new solution utilizing the crossover operator of the well-known GA. The new solution is constructed by combining both the best solution identified so far and a new solution generated using a constructive heuristic. This newly generated solution is then set as the current solution. The idea is to allow the algorithm enough diversification power, since the new solution, which is approximately as good as the current best solution, will be placed in a different region of the search space, thanks to the power of the crossover.

In addition, in most studied ALNS applications, the best solution is updated only if the newly generated solution is better than the current best solution. In contrast, in our ALNS variant, we adopt an acceptance function that is applied for the best solution. In other words, if the solution is not worse than $\delta\%$ of the current best solution, the solution is accepted. This solution is then improved using a local search procedure. After this, the solution obtained is compared with the best solution to decide whether to accept or reject the new improved solution. Thus, our ALNS gives another chance for promising solutions to become new best solutions after being improved by the local search procedure, which adds more diversification power. On the other hand, applying local search to improve promising solutions is intended to further intensify the search and improve the quality of these solutions even more.

To sum up, all the aforementioned characteristics shape a new hybrid ALNS, which combines advantages of the intensification potential of the local search operators, the diversification potential of the crossover, and the flexibility of the acceptance function mechanism applied on the best solution in a novel way.

The structure of our ALNS algorithm shown in Algorithm 1 is based on that proposed by Masmoudi et al. (2016). The algorithm executes for a fixed time to find a best solution x and its routing cost $f(x)$. Let x the initial solution and x_{best} the current best solution. The temperature T is initialized to its maximum value T_{max} and the weights and scores of the removal and insertion operators are also initialized, but they are updated during the search.

At each ALNS iteration, combinations of operators (removal and insertion) (see subsection 4.5) are selected according to their past performance (see subsection 4.4). This is done as follows: in case x_{best} is improved in the last iteration, one removal and one insertion operator are applied. Otherwise, two removal operators are performed in a random order to destroy the partial solution, followed by one insertion operator to repair the solution. Our insertion operators insert unserved requests, if feasible. In case some requests cannot be served, due to constraints violation, one more conventional vehicle will be added to the solution. In this case, the best solution in terms of cost will be the solution having a fewer number of additional vehicles.

After removing recharging station node(s) and re-insertion of users, the current solution may become infeasible, due to fuel related constraints. In this case, the two relevant operators Remove

Station (RS) and Insert Station (IS) are applied in a random order to restore feasibility. Thus, a new solution x' is obtained. x' is accepted if it is better than the current solution x , or if it satisfies the SA acceptance criterion $e^{(f(x)-f(x')/T)}$. Otherwise, a new solution is created using a randomly selected crossover operator (O1, O2, or O3) (see subsection 4.2), which combines different characteristics inherited from the current x_{best} and a newly generated solution using the constructive heuristic (see subsection 4.1). When obtaining a new solution, we decide to accept or reject the solution. If the objective function of x' is better than that of x_{best} , x' becomes the new best solution. Else, if the new solution x' is not worse by more than 2% of the current best solution x_{best} , x' is enhanced by the local search strategy (see subsection 4.3). Then, the new solution may become the current best solution, only if it has better quality.

Algorithm 1: Hybrid Adaptive Large Neighborhood Search Algorithm

1. **Initialize:** The weights and scores of removal, insertion, and local search operators, $x_{best} = x$, $T = T_0$
 2. **While** the stopping criterion is not reached **Do**
 3. Select and apply the removal operator(s) on the current solution x ;
 4. Select and apply the insertion operators on x to obtain x' ;
 5. Perform the RS and IS operators on x' ;
 6. **If** x' is feasible **Then**
 7. **If** $f(x') < f(x)$ or $f(x')$ satisfies the acceptance criterion **Then**
 8. $x \leftarrow x'$;
 9. **Else If** $f(x') > f(x)$ **Then**
 10. $x_{new} \leftarrow$ newly constructed solution by the constructive heuristic
 11. $x \leftarrow$ Crossover(x_{best} , x_{new});
 12. **Else If** $f(x') < f(x_{best})$ **Then**
 13. $x_{best} \leftarrow x'$;
 14. $T_b = T$;
 15. **Else If** $f(x') < f(x_{best}) * 1.02$ **Then**
 16. $x' \leftarrow$ local search (x');
 17. **If** $f(x') < f(x_{best})$
 18. $x_{best} \leftarrow x'$;
 19. $T_b = T$;
 20. **End If**
 21. **End If**
 22. **End If**
 23. $T \leftarrow \alpha \times T$;
 24. **If** $T < 0.01$
 25. $T_b = 2 * T_b$;
 26. $T = \min\{T_b, T_{max}\}$;
 27. **End If**
 28. Adjust the weights and probabilities of the removal, insertion and local search operators
 29. **End While**
 30. **Return:** x_{best}
-

As in similar ALNS applications in the literature, the temperature is reduced after each iteration by multiplying T by a cooling factor α . If after the reduction, the temperature becomes less than 0.01, then, T_b is multiplied by two and the temperature T is set to T_b , where T_b is applied to record the

temperature when x_{best} is found. In order to avoid that the search restarts from scratch from a randomly generated solution, we limit the temperature T to T_{max} .

4.1. An initial solution

The proposed heuristic for constructing the initial solution considers the AFS nodes in the planning of routes. A list L containing a set of users to be served is initialized to be inserted one by one in a set of empty CVs and AFVs routes. The following steps are run. A vehicle starts at the depot, visits a set of users, and then returns to the depot. While an AFV is still available, the insertion of users is performed by inserting a randomly selected user i from the list L in the best position of its pickup and delivery nodes in already existing routes, provided that the ride time, time windows, vehicle capacity and maximum route duration are respected. If a user i cannot be added in the route due to lack of fuel, the selected user is re-inserted along with a refueling station node, such that the nearest recharging station node to the already existing node $i-1$ is inserted. If the insertion of user i is not possible in already existing routes, a new route is added to the current solution and the same insertion procedure is applied. If the user cannot be assigned to any available AFV, the user is then inserted into a CV route until at most the predefined number of CVs and AFVs is constructed.

4.2. Diversification mechanism

To diversify the search, we develop three effective and simple crossover operators that are well-known in GA literature. The advantage of using a diversification procedure is to discover new regions of the search space that may not have been visited yet by the insertion and removal operators.

One-point crossover (O1): This crossover operator is inspired from Prins (2004). First, a random point p is selected, then the new solution acquires all the users and AFS nodes (if found) from the best solution x_{best} before the crossover point p . To complete the new solution, the remaining elements are inherited from x_{best} in the same order as they appear in a solution generated using our constructive heuristic beginning from the first route.

Two-point crossover (O2): This operator is the classical Order Crossover using two points proposed by Goldberg (1988). This operator arbitrarily selects two crossover points and transcribes the partial permutation between them moving from x_{best} into the new solution. While maintaining their relative ordering, the rest of elements are taken from a solution generated using our constructive heuristic.

Linear two-point crossover (O3): This operator is proposed by Sevaux and Dauzere-Peres (2003), which is similar to O2. The only difference is that the remaining users and refueling nodes are inherited from the solution generated using our constructive heuristic, starting with the first position from left to right of the order of users and recharging station nodes of each route in the solution.

4.3. Local Search operators

To further improve the quality of solutions, we apply several well-known local search operators. These include two intra-route operators: the 2-opt operator of Lin (1965) and the relocate operator of Savelsbergh (1992). Moreover, two inter-route operators are applied: the 2-opt* operator of Potvin and Rousseau (1995), and the relocate operator of Savelsbergh (1992). We note that the 2-opt* operator is

applied only between the CVs or between the AFVs routes (including the alternative fuel station nodes). To accept new solutions during local search, first improvement strategy is applied. This is done by generating all possible neighbors of the current solution, using the current local search operator, until an improving solution is located. If no improving solution is found, the next local search operator is applied. If all local search operators are tried and no improving solution is found, the procedure terminates and the current solution is returned. In addition, our selection of the local search operator (I1, I2, I3 and I4) is distinguished by using a roulette wheel selection mechanism, based on the performance score of the operator, instead of random selection, as described in subsection 4.4. This procedure can achieve a balance between the quality of the solution and run time.

After processing the neighborhood moves, some recharging stations nodes may become redundant in the solution, and the current solution may need recharging station node(s). Thus, two operators adopted from Schneider et al. (2014) are also used, namely remove and insert station operators as described next.

Remove station (RS): This operator checks at each route in a solution each pair of nodes (i, j) . If the refueling level in the tank at node i is enough to visit directly node j , the refueling station node between them is then deleted.

Insert station (IS): This operator considers all nodes (i, j) of each AFV route $(\forall i, j \in N)$, such that if the remaining fuel level in the tank of the vehicle k at node i is not enough to visit directly node j , an AFS node is inserted. The insertion is done by finding the nearest refueling station to the current node i . We note that by each visit of refueling station, the tank of the vehicle k is refueled to its maximum level.

4.4. Adaptive weight adjustment procedure

Our ALNS uses five removal operators, four re-insertion and four local search operators. We select an appropriate operator at each iteration, using a roulette wheel selection mechanism. As in Ropke and Pisinger (2006a), the probability of choosing operator d at iteration t , is defined by $P_d^{t+1} = P_d^1(1 - r_p) + r_p \pi_i / \omega_i$, where r_p is the roulette wheel parameter, π_i is the score of an operator i , and ω_i is the number of times the operator i has been used in the last 100 iterations. Moreover, the score of an operator is increased according to the following criteria: i) the score is increased by π_1 , if the existing operator finds a new best solution; ii) the score is enhanced by π_2 , if it locates a better solution than the current; iii) if the current operator finds a feasible solution, which is non-improving, the scores of operators are increased by π_3 . After 100 iterations, the weights are adjusted using the scores obtained.

4.5. Removal and insertion operators

At each iteration, a set of nodes/users is selected from a current solution x and added to a list L by removal operators. Our five removal operators (R1 to R5) are adopted from Masmoudi et al. (2016), and are applied to destroy the current partial solution x . These operators are: random-users removal (R1), path-removal (R2), time-oriented removal (R3), related removal (R4), and distance-oriented removal (R5).

Also, in our ALNS, four insertion operators (P1 to P4) are implemented to reinsert the removed users to form a new solution, based on Masmoudi et al. (2016). These operators are: basic greedy insertion (P1), best position inter-route insertion (P2), sorting time insertion (P3), and best position intra-route insertion (P4). For a detailed description of these removal and insertion operators, the reader is referred to Masmoudi et al. (2016).

4.6. Evaluation function

We evaluate each solution by the following evaluation function based on Parragh (2011), and Masmoudi et al. (2018b): $f(s) = c(s) + \sum_{r=0}^3 \alpha q_e(s) + \beta d(s) + \gamma w(s) + \tau a(s) + o(s)$. The term $c(s)$ gives the routing cost of solution s . Moreover, the terms $q_r(s), d(s), w(s), a(s)$ and $o(s)$ represent the load, duration, time window, ride time and fuel violations, respectively. The violations are calculated as follows: $q_r(s) = \sum_{i=1}^{2n} (d_i^r - Q^r)^+$, $d(s) = \sum_{k=1}^K (B_e - B_d - T_{max})^+$, $w(s) = \sum_{i=1}^{2n} (B_i - T_i^+)^+$ and $a(s) = \sum_{i=1}^n (l_i - L_{max})^+$. Note that these terms are applied only for all $i \in N$ where $x^+ = \{0, x\}$ and K is the set of the fleet size composed by the CVs and AFVs. The term $o(s)$ is computed as follows: $z_i = z_{i-1} - FR_{ij} * c_{i-1,i}$, if $i \in V \setminus F$ and $z_i = H$, if $i \in F'$. Binary variable $o(s)$ is equal to 0 if $z_i \geq 0, \forall i \in V \setminus F$ and 1 otherwise. The associated penalty parameters α, β, γ and τ are dynamically adjusted during the search (as in Parragh et al., 2010 and Cordeau and Laporte, 2003). We note that a solution s can only become a new best solution if $q_r(s) = d(s) = w(s) = a(s) = o(s) = 0$.

5. Computational experiments

In this section, we present the details of the results obtained by our proposed algorithms. All algorithms are implemented in C language and performed on a configuration Intel Core i7-5555U 3.14 GHz and 8 GB RAM.

5.1. Data and experimental setting

Our generated small-medium dataset instances are based on the benchmark instances generated by Parragh et al. (2012) for the HDARP. These instances are divided into three sets (U, E, I), which have been developed based on the instances of Cordeau and Laporte (2003) for the standard DARP with heterogeneous vehicles and users. Two types of vehicles (T1 and T2) for each of the AFVs and CVs per instance are considered with four distinct resources identified in each vehicle. These include staff seats, patient seats, stretchers and wheelchair places. For each vehicle kind (AFV and CV), type T1 has a capacity of 1 staff seat, 6 patient seats, 0 stretchers and 1 place for a wheelchair; while type T2 has a capacity of 2 staff seats, 1 patient seat, 1 stretcher, and 1 place for a wheelchair. Table 1 presents the structure of the vehicle fleet and provides a general view related to the way of conducting and managing the various kinds of users in each of the three instance sets. The number of requests in these instances ranges from 16 to 96, while the number of vehicles is between 2 and 8 with a single depot. The maximum duration of the working day ranges between 240 and 720 minutes (depending of the instance), and the maximum ride time $L_{max} = 30$ minutes. The time window length is equal to 30 minutes and the fixed service time duration s_i is set to three minutes.

Table 1

Probabilities used to generate instances as in Parragh et al. (2012)

Instance Set	User request probabilities			Probability for a companion (%)	Vehicle fleet
	% Seat	% stretcher	% wheelchair		
<i>U</i>	0.50	0.25	0.25	0.00	Mixed (CVs (T1, T2) ;AFVs(T1,T2))
<i>E</i>	0.25	0.25	0.50	0.10	Mixed (CVs (T1,T2) ; AFVs(T1,T2))
<i>I</i>	0.83	0.11	0.06	0.50	Mixed (CVs (T1,T2) ; AFVs(T1,T2))

We suppose that at the beginning of the working day, each available vehicle type is fully refueled. We decided to set the number of recharging stations equal to the number of vehicles in each instance. The approximation of the number of recharging stations is based on the generated instances of Erdoğan and Miller-Hooks (2012), in which some instances that consist of three vehicles, the number of recharging stations is considered equal to three. Based on how Cordeau (2006) defines the coordinates of pickup and delivery nodes of users, all coordinates of AFSs are randomly generated in a specific square area (i.e., $[-10,10]^2$). To determine the number of CVs and the AFVs used in our problem, we apply the procedure of Goeke and Schneider (2015). First, we start with an overall vehicle number K that is associated with the number of CVs found in the HDARP instances Parragh et al. (2012), and then CVs are progressively replaced with AFVs until the number of the AFVs is equal to the number of CVs divided by two.

For the large size instances, we adopt the generation idea of the benchmark instances of Masmoudi et al. (2018b). These instances are divided into three data sets ($A0$, $A1$ and $A2$), as done in Braekers and Kovacs (2016). In the benchmark instances of Masmoudi et al. (2018b), each data set $A0$, $A1$ and $A2$ contains between 20 and 100 requests. The data set $A0$ is characterized by that the locations of the pickup and delivery of the users are distributed randomly, while the data sets $A1$ and $A2$ have clustered locations. A time window of 15 minutes is specified for the delivery/pickup node, in case of outbound/inbound request. In addition, the minimum ride time is assumed to be 60 minutes, while the maximum ride time is assumed to be double the direct ride time, i.e., $L_i^{max} = \max\{60, 2 \times t_{i,n+i}\}$. In addition, for each user a service time of three minutes is needed to complete the service. The number of refueling stations in each instance is equal to $0.3 \cdot |n|$. The limited route duration ranges between 480 min and 720 min. Accordingly, we generated large size instances containing between 100 and 200 requests. For the number of vehicles of each instance, we use the same available number of vehicles in Braekers and Kovacs (2016), unless it is not enough to serve n users. For more details, we refer to Braekers and Kovacs (2016) and Masmoudi et al. (2018b). In addition, to determine the number of CVs and AFVs used in each instance, we apply the same procedure as defined previously. Different types of users and vehicles are considered as explained previously.

5.2. Parameter setting

This section explains the sensitivity analysis to set the parameters of our algorithm. Mainly, the parameters are chosen based on recommendations from the literature (e.g., Ropke and Pisinger, 2006a,b, Demir et al., 2012; Leung et al., 2013; Masmoudi et al., 2016) and our preliminarily experiments. We initialize $P_d^0 = 0.10$ for the removal operators, 0.125 for the insertion operators, $r_p =$

0.7, $\pi_1=15$, $\pi_2=5$, $\pi_3=10$, as suggested by Masmoudi et al.(2016), the temperature value $T_{max}=25$ as suggested by Leung et al.(2013), since it is enough to accept a deteriorating solution and $\alpha =0.99975$ as suggest by Ropke and Pisinger (2006a,b) and Demir et al.(2012). A summary of all used parameters in our hybrid ALNS is shown in Table 13 in the Appendix.

To study the performance of different removal and reinsertion operators of the ALNS, we use a similar tuning methodology as Demir et al.(2012). Table 2 shows the percentage of time each operator is used by our algorithm within 5 minutes of runtime. The numbers in brackets refer to the total time spent to run each operator. The results are obtained considering five instances from each data set (U, E, I), with different levels of heterogeneity. Each instance is computed ten times. We report the overall average of average results values (Avg) for each data set (U, E, I) and for all instances in the last line of Table 2.

Table 2

Percentage of use within 5 minutes of runtime and the computational time needed by each operator

Inst. (Data)	Removal operators					Insertion operators			
	R1	R2	R3	R4	R5	P1	P2	P3	P4
a4-16(U)	22.27(0.0)	15.15(0.0)	22.67(0.0)	18.62(0.0)	21.29(0.0)	26.80(0.0)	17.74(0.0)	29.22(0.0)	26.26(0.0)
a5-40(U)	21.18(0.0)	26.72(0.0)	24.79(0.0)	8.85(0.0)	18.47(0.0)	37.50(0.0)	13.75(0.0)	21.57(0.0)	27.19(0.1)
a5-60(U)	25.17(0.1)	21.07(0.0)	13.72(0.0)	17.34(0.0)	22.70(0.0)	28.85(0.0)	15.63(0.0)	24.54(0.0)	30.99(0.2)
a6-60(U)	22.28(0.0)	15.25(0.0)	22.71(0.0)	18.40(0.0)	21.36(0.1)	25.64(0.2)	18.45(0.1)	25.02(0.0)	30.91(0.4)
a7-56(U)	18.51(0.0)	23.85(0.0)	21.09(0.0)	21.52(0.0)	15.02(0.0)	40.34(0.1)	18.32(0.1)	15.86(0.1)	25.49(0.5)
Avg (U)	21.88(0.0)	20.41(0.0)	21.00(0.0)	16.95(0.0)	19.77(0.0)	31.83(0.1)	16.78(0.0)	23.24(0.0)	28.17(0.2)
a6-48(E)	18.21(0.0)	29.03(0.0)	19.99(0.0)	21.13(0.0)	11.65(0.0)	31.48(0.3)	19.85(0.0)	18.79(0.1)	29.88(0.4)
a6-72(E)	24.35(0.0)	29.17(0.0)	16.90(0.0)	10.70(0.0)	18.88(0.0)	43.24(0.1)	18.77(0.1)	20.26(0.1)	17.73(0.2)
a7-70(E)	20.26(0.0)	17.17(0.0)	25.32(0.0)	14.03(0.0)	23.22(0.0)	36.17(0.5)	20.49(0.1)	17.64(0.1)	25.51(0.4)
a8-64(E)	20.15(0.0)	24.70(0.0)	11.60(0.1)	19.81(0.1)	23.74(0.0)	26.90(0.0)	22.08(0.0)	20.05(0.1)	30.96(0.6)
a8-96(E)	22.95(0.0)	19.77(0.1)	16.13(0.2)	24.75(0.0)	16.41(0.1)	31.04(0.1)	19.82(0.1)	20.34(0.3)	28.82(0.2)
Avg (E)	21.18(0.0)	23.97(0.0)	17.99(0.1)	18.08(0.0)	18.78(0.0)	33.77(0.2)	20.20(0.1)	19.42(0.1)	26.58(0.4)
a6-48(I)	22.89(0.1)	25.32(0.1)	22.29(0.1)	10.92(0.1)	18.58(0.0)	33.39(0.6)	23.37(0.1)	22.20(0.0)	21.03(0.4)
a6-60(I)	23.33(0.1)	16.57(0.1)	14.40(0.1)	19.85(0.2)	25.83(0.1)	25.35(0.9)	25.49(0.1)	18.57(0.0)	30.59(0.8)
a7-56(I)	25.34(0.1)	21.45(0.2)	15.97(0.1)	18.86(0.1)	18.38(0.1)	26.59(0.7)	25.18(0.2)	18.63(0.2)	29.59(0.6)
a8-64(I)	27.43(0.1)	20.61(0.2)	14.01(0.1)	20.19(0.1)	17.76(0.1)	30.17(0.9)	26.00(0.4)	20.47(0.2)	23.38(0.7)
a8-80(I)	27.53(0.1)	22.02(0.3)	2.99(0.2)	27.91(0.3)	19.55(0.1)	32.77(1.4)	24.81(0.6)	17.21(0.2)	25.20(1.1)
Avg (I)	25.30(0.1)	21.19(0.2)	13.93(0.1)	19.55(0.2)	20.02(0.1)	29.65(0.9)	24.97(0.3)	19.42(0.1)	25.96(0.7)
Avg (UEI)	22.79(0.0)	21.86(0.1)	17.64(0.1)	18.19(0.1)	19.52(0.0)	31.75(0.4)	20.65(0.1)	20.69(0.1)	26.90(0.4)

The results in Table 2 show that removal operators have similar frequency of use in many cases. This is due to applying two operators in the same iteration for most of the cases. Moreover, P1 and P4 operators (as indicated in bold) are applied to some extent more than the other two operators. Therefore, by comparison with the rest of operators in terms of CPU time, P1 and P4 are significantly used more than the other operators.

Table 3 indicates the number of times an operator has found the best and a better solution than the current one, respectively; i.e., the values in brackets refer to the number of times the current solution has been improved, but has not become a best solution. We report the overall average of average results values (Avg) for each data set (U, E, I) and for all instances in the last line of Table 3.

Table 3

Number of global best and number of improved solutions attained by each operator within 5 minutes of runtime

Inst. (Data)	Removal operators					Insertion operators			
	R1	R2	R3	R4	R5	P1	P2	P3	P4
a4-48 (U)	27(179)	13(193)	14(215)	31(71)	20(189)	40(51)	16(393)	8(318)	36(85)
a5-50 (U)	20(170)	20(94)	17(219)	27(239)	17(125)	31(158)	19(258)	10(285)	40(146)
a6-72 (U)	27(62)	30(200)	17(84)	16(225)	13(156)	32(94)	18(194)	11(251)	39(188)
a7-56 (U)	28(118)	22(183)	18(212)	11(239)	19(211)	31(249)	14(243)	18(259)	37(212)
a8-80 (U)	15(53)	27(91)	16(225)	25(241)	17(153)	23(166)	19(308)	21(245)	37(44)
<i>Avg (U)</i>	<i>23(116)</i>	<i>22(152)</i>	<i>16(191)</i>	<i>22(203)</i>	<i>17(167)</i>	<i>31(144)</i>	<i>17(279)</i>	<i>14(272)</i>	<i>38(135)</i>
a4-16 (E)	26(130)	18(110)	17(133)	22(232)	21(213)	41(99)	21(300)	12(382)	26(37)
a6-48 (E)	26(101)	20(167)	22(114)	10(151)	22(100)	25(143)	28(195)	15(222)	32(73)
a7-56 (E)	22(117)	16(80)	24(58)	16(228)	29(81)	38(44)	6(196)	14(211)	42(113)
a7-84 (E)	21(70)	18(136)	26(40)	29(207)	25(188)	36(161)	13(190)	8(130)	42(160)
a8-96 (E)	18(131)	29(229)	18(122)	17(183)	18(118)	34(193)	8(125)	17(248)	41(217)
<i>Avg (E)</i>	<i>23(110)</i>	<i>20(144)</i>	<i>21(93)</i>	<i>19(200)</i>	<i>23(140)</i>	<i>35(128)</i>	<i>15(201)</i>	<i>13(239)</i>	<i>37(120)</i>
a4-24 (I)	25(212)	17(194)	21(114)	23(160)	14(86)	35(150)	16(208)	25(310)	24(98)
a5-60 (I)	27(127)	24(133)	20(138)	14(219)	15(76)	29(155)	15(286)	28(197)	28(55)
a7-56 (I)	22(203)	7(159)	11(189)	19(157)	25(64)	26(256)	25(131)	22(213)	27(172)
a7-84 (I)	19(209)	27(233)	25(131)	14(87)	15(31)	32(205)	28(234)	17(176)	23(76)
a8-64 (I)	23(130)	15(237)	24(116)	16(140)	22(152)	29(178)	24(261)	11(273)	36(63)
<i>Avg (I)</i>	<i>23(176)</i>	<i>18(191)</i>	<i>20(138)</i>	<i>17(153)</i>	<i>18(82)</i>	<i>30(189)</i>	<i>22(224)</i>	<i>21(234)</i>	<i>28(93)</i>
<i>Avg (UEI)</i>	<i>23(134)</i>	<i>20(163)</i>	<i>19(141)</i>	<i>19(185)</i>	<i>19(130)</i>	<i>32(153)</i>	<i>18(235)</i>	<i>16(248)</i>	<i>34(116)</i>

As illustrated by the obtained results in Table 3, all removal operators seem to take part in generating better solutions. Even though the ratio of obtaining better or best results changes between operators, the use of various neighborhood structures might help the search towards global optima. By looking at the literature on neighborhood structures (see, e.g., Ropke and Pisinger 2006a, b; and Demir et al., 2012), it is well known that some operators might perform well on different instances of the same problem. On the other hand, with respect to the insertion operators, we observe that some insertion operators (i.e., P2 and P3) do not often contribute much. Nevertheless, as indicated in the literature, ALNS might need various operators as they are beneficial for obtaining better solutions in the following iterations. This is evident by the large number of times these two operators could improve the current solution (as shown from the values between brackets). Improving the current solution is obviously beneficial for the overall performance of the ALNS and to avoid being trapped in local optima. Consequently, it is deduced that there is a positive contribution from all operators, which helps in acquiring high quality solutions for the MF-HDARP.

As Pisinger and Ropke (2007) indicated, it is not essential to delete a large number of users u in the removal phase, because the deletion of a specific number of users will have a considerable influence on the results. Accordingly, the number of deleted users u is chosen randomly in the interval $[u_{min}=0.175.n; u_{max}=0.35.n]$. As demonstrated next, Table 4 gives an idea about the results of the parameter tuning of π_1 , π_2 and π_3 . After considering the combination of seven different values, the tuning sequence is manifested by the arrangement of the parameters, and it is displayed in the first line. To find the best set of parameters, we tested five instances from each data set (U, E, I). These instances have a number of requests ranging from small to large and different levels of heterogeneity. On each instance, we calculate the average solution value for ten runs obtained using each combination of parameters (π_1, π_2, π_3) .

Table 4Effect of π_1 , π_2 and π_3 on the solution quality

Inst. (Type)	(10, 5,1)	(15, 10,5)	(1, 1,1)	(1, 5,5)	(1,5, 10)	(1, 10, 5)	(15, 5,10)	(10, 1,5)	(5, 1,5)
a4-48 (U)	758.20	756.86	759.05	756.52	758.76	757.92	757.45	758.13	757.25
a5-50 (U)	725.44	726.68	727.97	727.01	728.46	725.95	725.08	725.71	728.82
a6-72 (U)	924.90	924.05	927.46	924.15	927.53	926.83	929.15	927.78	928.12
a7-56 (U)	717.69	715.92	718.03	715.12	718.72	716.45	713.58	719.02	719.15
a8-80 (U)	1001.88	1009.72	1010.18	1003.09	1011.01	1009.43	1009.78	1002.63	1011.62
a4-16 (E)	312.14	310.18	312.59	309.90	311.14	312.18	309.86	312.49	310.97
a6-48 (E)	610.05	611.70	609.44	610.70	609.87	610.93	608.35	607.75	609.96
a7-56 (E)	709.68	706.13	712.18	707.46	712.68	706.99	707.46	708.87	713.08
a7-84 (E)	1157.61	1163.63	1162.96	1160.47	1162.36	1162.02	1164.50	1161.16	1161.92
a8-96 (E)	1291.24	1291.88	1292.17	1293.20	1292.33	1290.98	1291.24	1290.62	1292.60
a4-24 (I)	389.01	390.35	388.88	389.97	389.07	389.16	387.68	387.92	389.34
a5-60 (I)	784.90	783.86	786.38	786.33	785.71	783.82	786.81	783.81	784.46
a7-56 (I)	723.52	724.28	724.25	722.92	724.01	723.96	727.63	722.95	722.59
a7-84 (I)	1113.58	1112.99	1112.15	1119.30	1114.08	1120.12	1112.84	1120.13	1114.83
a8-64 (I)	739.86	745.94	740.62	738.96	738.99	745.39	736.02	744.27	736.03

Due to the help of diversification techniques, our setting of the parameters π_1 , π_2 and π_3 is consistent with the expected setting $\pi_1 \geq \pi_3 \geq \pi_2$ for rewarding a good performance of an operator.

5.3. Computational analysis

This section presents and compares the detailed results obtained by our hybrid ALNS tested on the benchmark HDARP instances of Masmoudi et al. (2017) and on our newly generated instances of the MF-HDARP. The detailed results found by our algorithm are available on <http://www.mf-mp-hdarp-88.websself.net>.

5.3.1 Results on the heterogeneous DARP instances of Masmoudi et al. (2017)

To evaluate the performance of our algorithm, we tested it on the large sized benchmark HDARP instances of Masmoudi et al. (2017) having up to 13 vehicles and 144 requests. By replacing our AFVs fleet by CVs, the vehicles do not need to be refueled, so in this case we obtain vehicles of the same type (i.e., all are CVs). Thus, our MF-HDARP is transformed to classical HDARP. Tables 5 and 6 present the detailed results of our algorithm on the large instances of Masmoudi et al. (2017) for the HDARP. Three data set benchmark instances (U, E and I) are used. Each one contains 20 instances. The data set U contains homogenous users and vehicles. Data set E is characterized by heterogeneous users and homogenous vehicles, while data set I contains heterogeneous users and heterogeneous vehicles. We compare our hybrid ALNS with the current state-of-art algorithms in the literature on the HDARP, namely the hybrid Genetic Algorithm (hybrid GA) of Masmoudi et al. (2017) and the Evolutionary Variable Neighborhood Search (EVO-VNS₁) of Masmoudi et al.(2018b). We note that we have chosen only EVO-VNS₁ since it presents the best approach compared to the other EVO-VNS versions (i.e., EVO-VNS₂ and EVO-VNS₃) developed by Masmoudi et al.(2018b). All algorithms (including the hybrid GA and EVO-VNS₁) are applied for five runs as done in Masmoudi et al. (2017). In Tables 5, 6, and 7, column “BKS” refers to the best-known solution. Columns “Best (%)” and “Avg(%)” represent, respectively, the percent gap (deviation) from the best known solution (average solution) in five runs. It should also be noted that to obtain a fair comparison with respect to the computational time, we have run the GA of Masmoudi et al. (2017) on the same machine used in this

work with maximum runtime as a unified stopping criterion, which is equal to 30 minutes for each instance for all algorithms.

Table 5
Comparison of our hybrid ALNS with the EVO-VNS₁ of Masmoudi et al.(2018b) on data set E

Inst.	BKS ^a	EVO-VNS ₁				Hybrid ALNS			
		Best	Best%	Avg	Avg%	Best	Best%	Avg	Avg%
R1a	195.97	195.97	0.00	195.97	0.00	195.97	0.00	195.97	0.00
R2a	336.34	336.34	0.00	336.34	0.00	336.34	0.00	336.34	0.00
R3a	586.18	586.18	0.00	587.15	0.17	586.18	0.00	586.18	0.00
R4a	639.03	639.03	0.00	642.33	0.52	639.03	0.00	642.16	0.49
R5a	713.09	713.09	0.00	716.66	0.50	713.09	0.00	716.38	0.46
R6a	882.11	882.11	0.00	883.23	0.13	882.11	0.00	883.86	0.20
R7a	310.96	310.96	0.00	313.13	0.70	311.55	0.19	312.67	0.55
R8a	553.82	554.06	0.04	556.40	0.47	554.50	0.12	556.26	0.44
R9a	744.34	744.64	0.04	748.46	0.55	740.36	-0.53	746.92	0.35
R10a	963.08	963.93	0.09	964.56	0.15	958.05	-0.52	963.22	0.01
R1b	190.39	190.39	0.00	190.39	0.00	190.39	0.00	190.39	0.00
R2b	312.92	312.92	0.00	312.92	0.00	312.92	0.00	312.92	0.00
R3b	551.95	551.95	0.00	553.18	0.22	551.95	0.00	551.95	0.00
R4b	605.29	605.29	0.00	608.52	0.53	605.29	0.00	605.29	0.00
R5b	640.50	640.50	0.00	642.09	0.25	641.00	0.08	642.36	0.29
R6b	832.53	832.79	0.03	835.85	0.40	832.53	0.00	835.27	0.33
R7b	276.17	276.17	0.00	276.60	0.16	276.17	0.00	276.93	0.28
R8b	529.96	529.96	0.00	531.79	0.35	530.39	0.08	531.31	0.25
R9b	698.13	698.13	0.00	700.36	0.32	698.13	0.00	699.23	0.16
R10b	902.17	903.28	0.12	904.17	0.22	898.04	-0.46	903.27	0.12
	573.25	573.38	0.02	575.01	0.28	572.70	-0.05	574.44	0.20

^aBest known solutions provided from Masmoudi et al.(2018b)

The results in Table 5 show that our hybrid ALNS is competitive with the EVO-VNS₁ algorithm of Masmoudi et al. (2018b) and provides good results. Regarding the number of best solutions in five runs (column Avg), it is clear that our hybrid ALNS outperforms the EVO-VNS by providing 13 best average solutions compared to only three best averages for EVO-VNS₁. While for the number of best solutions over five runs (column Best), both methods provide the same number with four solutions each. However, three best new solutions are obtained by our ALNS for the instances R9a, R10a and R10b. For the average deviation of the average results from the best-known solution, the gap is very small, where 0.20% is obtained by our ALNS, compared to 0.28% achieved by the EVO-VNS₁. For the average deviation of the best result over five runs, our hybrid ALNS improves the results with 0.05%.

Table 6
Comparison of our hybrid ALNS with the hybrid GA of Masmoudi et al. (2017) on data set U

Inst.	BKS	Hybrid GA				Hybrid ALNS			
		Best	Best %	Avg	Avg%	Best	Best %	Avg	Avg%
R1a	190.02	190.02	0.00	190.02	0.00	190.02	0.00	190.02	0.00
R2a	301.34	301.34	0.00	301.34	0.00	301.34	0.00	301.34	0.00
R3a	532.00	532.00	0.00	534.08	0.39	532.00	0.00	532.47	0.09
R4a	570.25	570.25	0.00	571.45	0.21	570.25	0.00	571.61	0.24
R5a	626.93	628.48	0.25	631.39	0.71	627.77	0.13	628.33	0.22
R6a	785.26	787.41	0.27	788.52	0.42	785.51	0.03	787.60	0.30
R7a	291.71	291.71	0.00	291.79	0.03	291.96	0.09	292.47	0.26

R8a	487.84	488.89	0.22	491.53	0.76	488.71	0.18	489.47	0.33
R9a	658.31	658.31	0.00	660.24	0.29	659.27	0.15	660.09	0.27
R10a	851.82	853.16	0.16	859.91	0.95	853.47	0.19	854.67	0.33
R1b	164.46	164.46	0.00	164.46	0.00	164.46	0.00	164.46	0.00
R2b	295.66	295.66	0.00	295.66	0.00	295.65	0.00	295.65	0.00
R3b	484.83	484.83	0.00	487.23	0.50	485.19	0.07	485.82	0.21
R4b	529.33	531.86	0.48	532.19	0.54	530.46	0.21	530.79	0.28
R5b	577.29	579.03	0.30	582.06	0.83	577.41	0.02	579.05	0.30
R6b	730.69	737.03	0.87	741.06	1.42	731.93	0.17	733.91	0.44
R7b	248.21	248.21	0.00	248.29	0.03	248.67	0.18	248.99	0.31
R8b	458.73	461.11	0.52	463.32	1.00	459.65	0.20	460.00	0.28
R9b	593.49	593.49	0.00	595.37	0.32	593.80	0.05	594.64	0.19
R10b	785.68	791.01	0.68	793.64	1.01	786.06	0.05	788.96	0.42
<i>U</i>	508.19	509.41	0.19	511.18	0.47	508.68	0.09	509.52	0.22

^aBest known solutions provided from Masmoudi et al. (2017)

Again, observing the detailed results in Table 6, our hybrid ALNS obtains good results compared to the hybrid GA of Masmoudi et al. (2017). The average gap to the best solution achieved by our hybrid ALNS algorithm amounts to 0.09%, compared to 0.19% for the hybrid GA. The average deviation of five runs for the hybrid ALNS represents 0.22%, while 0.47% is obtained by the hybrid GA.

Table 7

Comparison of our hybrid ALNS with the hybrid GA of Masmoudi et al. (2017) on data set I

Inst.	BKS	Hybrid GA				Hybrid ALNS			
		Best	Best %	Avg	Avg%	Best	Best %	Avg	Avg%
R1a	190.02	193.27	0.00	193.27	0.00	193.27	0.00	193.27	0.00
R2a	301.34	319.43	0.00	319.87	0.14	319.43	0.00	319.43	0.00
R3a	532.00	584.84	0.00	586.11	0.22	584.05	-0.13	585.05	0.04
R4a	570.25	591.24	0.00	593.56	0.39	590.57	-0.11	591.39	0.03
R5a	626.93	677.50	0.00	679.11	0.24	677.72	0.03	678.26	0.11
R6a	785.26	838.26	0.00	843.27	0.60	836.70	-0.19	838.76	0.06
R7a	291.71	328.10	0.00	329.12	0.31	327.68	-0.13	328.97	0.26
R8a	487.84	552.35	0.00	556.46	0.74	551.01	-0.24	553.47	0.20
R9a	658.31	713.55	0.00	718.55	0.70	713.74	0.03	714.65	0.15
R10a	851.82	932.83	0.00	937.23	0.47	927.75	-0.54	933.22	0.04
R1b	164.46	177.57	0.00	177.57	0.00	177.57	0.00	177.57	0.00
R2b	295.66	304.02	0.00	304.02	0.00	304.02	0.00	304.02	0.00
R3b	484.83	551.13	0.00	555.19	0.74	548.96	-0.39	550.85	-0.05
R4b	529.33	557.99	0.00	559.12	0.20	555.76	-0.40	557.48	-0.09
R5b	577.29	628.62	0.00	630.59	0.31	626.03	-0.41	628.02	-0.09
R6b	730.69	794.03	0.00	797.57	0.45	790.93	-0.39	794.82	0.10
R7b	248.21	297.41	0.00	297.51	0.03	295.03	-0.80	297.10	-0.10
R8b	458.73	517.26	0.00	520.01	0.53	514.00	-0.63	517.20	-0.01
R9b	593.49	662.75	0.00	666.44	0.56	659.86	-0.44	662.39	-0.05
R10b	785.68	865.07	0.00	873.18	0.94	860.23	-0.56	864.81	-0.03
<i>I</i>	508.19	554.36	0.00	556.89	0.38	552.72	-0.27	554.54	0.03

^aBest known solutions provided by Masmoudi et al.(2017)

However, as indicated by the results in Table 7, our hybrid ALNS is more efficient than the hybrid GA in the case where heterogeneous users and vehicles are used. The average deviation of the average results derived from the best-known solutions in data sets I is 0.03% for our hybrid ALNS, compared to 0.38% for the Hybrid GA. In addition, our hybrid ALNS improves the results of Masmoudi et al. (2017) in the average deviation from the best result over five runs by 0.27%. Also, our hybrid ALNS provides 14 new best solutions compared to only two best solutions for the hybrid GA (column Best).

In addition, our hybrid ALNS is able to provide 17 best average solutions, compared to the hybrid GA (column Avg).

To sum up, it seems from the detailed results of Tables 5 to 7 that our hybrid ALNS is more effective on data set E (with heterogeneous users and homogeneous vehicles) and I (with heterogeneous users and heterogeneous vehicles). This is apparently due to the additional diversification and intensification mechanisms used in our algorithm, which seem to contribute positively to improving the quality of solutions.

5.3.2 Results on the new MF-HDARP instances

Since we used benchmark instances from the literature to test our method, we implemented the following approach. For the small-medium instances, our hybrid ALNS was run for a maximum of 5 minutes on each instance. Then, the average of five runs as well as the best solution in five runs obtained after 2 mins, 2.5 mins, 3 mins, 3.5 mins and 4 mins are recorded and the best solution values are compared to the best solution found after 5 minutes. Similarly, for the large size instances, our hybrid ALNS was run for a maximum of 60 minutes on each instance. Then, the best and average solution values obtained after 20 mins, 25 mins, 30 mins, 35 mins and 40 mins in 5 runs are recorded and compared to the values found after 60 minutes. We present the results of our algorithm on the small-medium and large MF-HDARP instances in Tables 8 and 9, respectively. The columns “Best%” and “Avg%” present the percentage of deviation from the best (“Best”) and average (“Avg”) solution values found by our algorithm after 5 minutes for the small-medium instances, and after 60 minutes for the large instances. Each instance is computed five times using each algorithm.

Table 8
Results for small/medium size instances

Inst.	ALNS (5 min)		ALNS (2 min)		ALNS (2.5 min)		ALNS (3 min)		ALNS (3.5 min)		ALNS (4 min)	
	Best	Avg	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%
U	648.70	648.82	20.51	20.53	9.40	9.42	4.25	4.26	1.55	1.56	0.18	0.19
E	657.01	657.25	26.47	26.52	13.55	13.59	6.91	6.94	3.54	3.57	1.36	1.39
I	657.69	657.85	23.21	23.24	11.26	11.28	5.82	5.84	2.41	2.42	0.51	0.53
Avg	654.47	654.64	23.40	23.43	11.40	11.43	5.66	5.68	2.50	2.52	0.68	0.70

Table 8 shows that after 2 minutes the average deviation from the best solution (obtained after 5 mins) is 23.43 percent. Nevertheless, after 4 minutes, the average deviation reduces to 0.70 percent. Furthermore, we can observe that the gap deviation (column Best%) progressively decreases as the time limit increases. That is, the gap is 12.00% (23.40% - 11.40%) when the time limit increases from 2 mins to 2.5 mins, 5.74% (11.40% - 5.66%) when the time limit increases from 2.5 mins to 3 mins, and 3.16% (5.66% - 2.50%) when the time limit increases from 3 mins to 3.5 mins. However, a slight decrease in the gap with 1.82% (2.50% - 0.68%) is observed when the time limit increases from 3.5 mins to 4 mins.

Table 9
Results for large size instances

Inst.	ALNS (60 min)		ALNS (20 min)		ALNS (25 min)		ALNS (30 min)		ALNS (35 min)		ALNS (40 min)	
	Best	Avg	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%
A0	2928.91	2963.78	12.18	13.43	4.16	5.28	1.06	2.14	0.08	1.14	0.00	1.06
A1	2969.22	3009.30	21.98	23.59	11.29	12.71	5.22	6.54	2.18	3.45	0.61	4.39
A2	3022.49	3057.36	15.83	17.10	6.98	8.11	2.75	3.82	0.97	2.01	0.23	2.50

Similar results are observed in Table 9, where it shows a progressive decrease in the change of gap (column Best%), with 9.19% (16.66% - 7.47%) when the time limit increases from 20 mins to 25 mins, 4.46% (7.47% - 3.01%) when the time limit increases from 25 mins to 30 mins. However, a very slight decrease in the deviation is observed with 1.93% (3.01% - 1.08%) for the case when the time limit increases from 30 mins to 35 mins, and 0.80% (1.08% - 0.28%) when the time limit increases from 35 mins to 40 mins.

Figures 1 and 2 summarize these findings, showing the decrease in the average GAP, while increasing the computational time. We can see that when increasing the computational time, the objective function converges. In addition, by using several instances with different characteristics as described in section 5.1, we can observe that our hybrid ALNS is efficient and robust, since it performs with similar quality on these instances, in different limits of computation time.

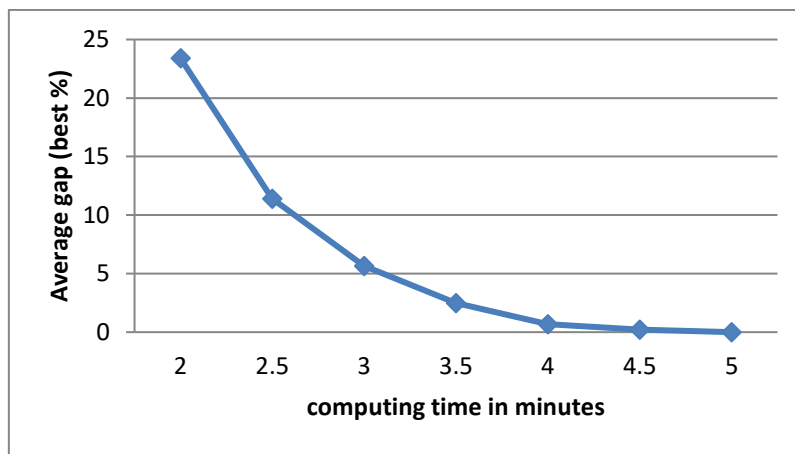


Fig.1. Average gap with respect to the best solution found after five minutes for small/medium instances plotted against computing time

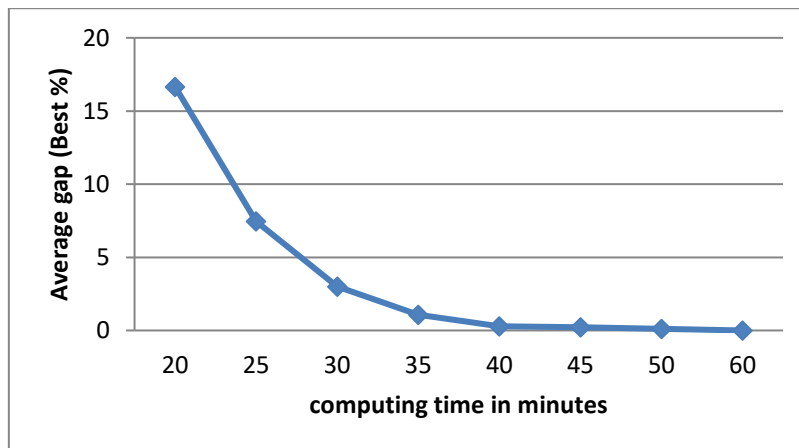


Fig.2. Average gap with respect to the best solution found after 60 minutes for large instances plotted against computing time

5.4. Study of the different algorithmic components of the hybrid ALNS

In this section, we study the impact of our different components (i.e., different crossover operators and the local search procedure with its modified acceptance function) on exploring the search space

and enhancing the solution quality. For this purpose, some combinations of operators are compared, with respect to the standalone (improved) ALNS of Masmoudi et al. (2016), by incorporating different component(s) each time. The detailed results of this comparison are shown in Table 11, where the large benchmark HDARP instances of Masmoudi et al. (2017) (Data set E) is used. First, in the combination “C1”, we apply the standalone (improved) ALNS of Masmoudi et al. (2016). The combination “C2” represents the combination of the (improved) ALNS with the local search procedure together with its acceptance function (Lines 15-19 of Algorithm 1). The combination “C3” represents the standalone (improved) ALNS using only one crossover operator (O1) (without the local search), while the combination “C4” applies three different crossover operators, instead of only one as in “C3”. The combination “C5” represents the combination “C2” by adding only one crossover operator. The combination “C6” represents the combination “C2” by adding two crossover operators (O1 and O2). The same for the combination “C7” but with using the two crossover operators (O1 and O3). While the last combination “C8” represents of the combination “C2” by adding three crossover operators, which reflects our hybrid ALNS described in Algorithm 1. We note that from “C2” to C8”, we apply the modified procedure of decreasing the temperature (steps 23-26). Table 10 summarizes these different combinations.

Table 10
Different combination of the algorithmic components

Comb.	Description
C1	Standard (improved)ALNS of Masmoudi et al. (2016)
C2	Standard (improved)ALNS of Masmoudi et al. (2016)+ local search procedure + modified acceptance function
C3	Standard (improved)ALNS of Masmoudi et al. (2016)+one crossover operator (O1)+ modified acceptance function
C4	Standard (improved)ALNS of Masmoudi et al. (2016)+ three crossover operators+ modified acceptance function
C5	Standard (improved)ALNS of Masmoudi et al. (2016)+ one crossover operator + local search procedure + modified acceptance function
C6	Standard (improved)ALNS of Masmoudi et al. (2016)+ two crossover operators (O1 and O2) + local search procedure + modified acceptance function
C7	Standard (improved)ALNS of Masmoudi et al. (2016)+ two crossover operators (O1 and O3) + local search procedure + modified acceptance function
C8	Standard (improved)ALNS of Masmoudi et al. (2016)+ three crossover operators + local search procedure + modified acceptance function

In Table 11, the columns “Best%” and “Avg%” represent the deviation gap from the best (“Best”) and average (“Avg”) results obtained by the EVO-VNS₁ of Masmoudi et al. (2018b). The run time on each instance is limited to 30 minutes. The best and average result values of this table are shown in our website.

Table 11
Impact of different components on the solution quality

Inst.	EVO-VNS1		C1		C2		C3		C4		C5		C6		C7		C8		
	Best	Avg	Best %	Avg%	Best %	Avg%	Best %	Avg%	Best %	Avg%	Best %	Avg%	Best %	Avg%	Best %	Avg%	Best %	Avg%	
R1a	195,97	195,97	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
R2a	336,34	336,34	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
R3a	586,18	587,15	0,24	0,56	0,24	0,50	0,23	0,52	0,21	-0,01	0,09	0,14	0,02	-0,09	0,08	-0,13	0,00	-0,17	
R4a	639,03	642,33	0,53	0,59	0,41	0,49	0,28	0,51	0,25	0,38	0,23	0,05	0,15	0,04	0,15	0,00	0,00	-0,03	
R5a	713,09	716,66	0,53	0,28	0,35	0,27	0,23	0,21	0,17	0,28	0,02	0,19	0,00	0,07	0,00	0,03	0,00	-0,04	
R6a	882,11	883,23	0,49	0,16	0,33	0,15	0,23	0,15	0,24	0,14	0,12	0,11	0,12	0,08	0,12	0,07	0,07	0,07	
R7a	310,96	313,13	0,27	0,17	0,27	0,17	0,27	0,17	0,27	0,03	0,26	0,11	0,25	0,03	0,25	-0,05	0,25	-0,15	
R8a	554,06	556,40	0,18	0,61	0,18	0,61	0,18	0,60	0,18	0,51	0,17	0,37	0,16	0,09	0,16	-0,01	0,16	-0,03	

R9a	744,64	748,46	0,14	0,32	0,14	0,32	0,12	0,28	0,03	-0,03	-0,35	0,01	-0,36	-0,01	-0,39	-0,12	-0,57	-0,21
R10a	963,93	964,56	0,14	0,71	0,14	0,71	-0,04	0,15	0,12	0,43	-0,42	0,70	-0,30	0,26	-0,35	0,19	-0,61	-0,14
R1b	190,39	190,39	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
R2b	312,92	312,92	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
R3b	551,95	553,18	0,16	0,11	0,15	0,10	0,15	-0,10	0,12	0,08	0,05	-0,02	0,05	-0,04	0,01	-0,18	0,00	-0,22
R4b	605,29	608,52	0,45	0,15	0,31	0,11	0,20	0,02	0,20	-0,04	0,19	-0,28	0,01	-0,45	0,07	-0,42	0,00	-0,53
R5b	640,50	642,09	0,50	0,72	0,41	0,71	0,21	0,66	0,20	0,40	0,17	0,22	0,10	0,18	0,11	0,16	0,08	0,13
R6b	832,79	835,85	0,50	0,61	0,41	0,60	0,19	0,33	0,18	0,56	0,14	0,18	0,08	0,11	0,08	0,11	-0,03	0,09
R7b	276,17	276,60	0,42	0,74	0,19	0,72	0,18	0,60	0,07	0,73	0,07	0,26	0,02	0,15	0,02	0,14	0,00	0,12
R8b	529,96	531,79	0,46	0,74	0,30	0,74	0,16	0,66	0,16	0,41	0,11	0,64	0,08	0,54	0,09	0,32	0,08	-0,09
R9b	698,13	700,36	0,45	0,81	0,35	0,72	0,20	0,25	0,07	0,65	0,02	0,52	0,02	0,07	0,02	0,18	0,00	-0,16
R10b	903,28	904,17	0,38	0,96	0,16	0,96	-0,01	0,90	-0,07	0,95	-0,12	0,50	-0,18	0,28	-0,06	0,21	-0,58	0,12
Avg	573,38	575,01	0,29	0,41	0,22	0,39	0,14	0,30	0,12	0,27	0,04	0,19	0,01	0,07	0,02	0,03	-0,06	-0,06

^aBest known solutions provided by Masmoudi et al.(2018b)

From the detailed results of Table 11, we can see that using the standalone (improved) ALNS (C1) based on Masmoudi et al. (2016) cannot provide good results, with an average gap to the best (average) results of Masmoudi et al. (2018b) that is equal to 0.29% (0.41%). However, a big improvement is obtained when applying the crossover operators in the ALNS (C3 and C4), where a negative deviation gap is obtained in some instances, indicating a better result than the best and average results of the EVO-VNS₁. Thus, combinations C3 and C4 show that using our crossovers is beneficial to enhance the quality of solutions and helps the algorithm to outperform the standalone (improved)ALNS. In addition, we can see that the solution quality is comparable for the two combinations C3 and C4, with a very small difference that is equal to 0.02%(0.03%), which indicates the effectiveness of using our diversification mechanism based on the crossovers.

Moreover, after applying the local search procedure with its acceptance function, we can see that the quality of solutions has improved, compared to the standalone (improved) ALNS, with an average gap equal to 0.07%(0.02%) (between C2 and C1). Also, for the combinations where crossover is applied, the average gap between C5 and C4 is equal to 0.08%(0.08%), and 0.10%(0.11%) between C5 and C3. Observing the combination C5, having the local search procedure and using only one crossover operator, we see that it still has positive average gap value, compared to the results of Masmoudi et al.(2018b) and to the combination C8 (that combines all components). Thus, we can see that applying this combination alone is still not capable of escaping local optima. Moreover, by applying two different crossover operators in C6 and C7 instead of only one (C5), we can see that the average gap is increased with a similar average gap for both these two combinations, where 0.01%(0.07%) is obtained by the C6 and 0.02%(0.03%) for the C7. However, applying all three crossovers and the local search operators with its acceptance function (i.e., C8) provides good solutions and outperforms the EVO-VNS₁ of Masmoudi et al.(2018b), although with a slight improvement of 0.06%(0.06%) over the best (average) results. In conclusion, applying all components (local search with the acceptance function as well as the three different crossovers) enhances both diversification and intensification during the search, compared to the standard(improved)ALNS. Thus, this combination is the most effective combination, compared to the other applied combinations.

6. Conclusions

This study tackles the MF-HDARP, where we have considered a mixed fleet of vehicles composed of both heterogeneous CVs and AFVs within the context of the DARP. We considered different capacity resources of the vehicles as well as the need for refueling. We have proposed an effective hybrid ALNS for solving the MF-HDARP. The algorithm is supported by an efficient constructive heuristic and sophisticated local search and diversification techniques to improve the solution quality. We tested our hybrid ALNS algorithm on newly generated instances and on the benchmark instances of Masmoudi et al.(2017), and compared its performance with state-of-the-art algorithms in the literature (the hybrid GA of Masmoudi et al., 2017 and the EVO-VNS₁ of Masmoudi et al., 2018b). The results indicate that our algorithm obtains high quality solutions and is competitive with the compared algorithms. Moreover, the results also indicate that our different components, which were added to the standard ALNS, improve its performance. Nevertheless, our hybrid ALNS algorithm is just slightly better than the EVO-VNS₁ of Masmoudi et al., 2018b. We believe that our hybrid ALNS can be further improved by utilizing additional removal and insertion operators, which can help in achieving more diversification of the search and make its behavior more robust.

In addition, from a methodological perspective, considering other type of AFVs with recharging such as electric vehicles and hybrid plug-in electric vehicles by developing realistic energy function for these type of vehicles are interesting research directions.

Acknowledgements

Thanks are due to two anonymous reviewers for their useful comments and for raising interesting points for discussion.

References

- Adler, J. D., & Mirchandani, P. B. The vehicle scheduling problem for fleets with alternative-fuel vehicles. *Transportation Science*, 51(2), 441-456 (2016)
- Alinaghian, M., & Shokouhi, N. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, 76, 85-99 (2018)
- Amirgholy, M., Gonzales, E. J., 2016. Demand responsive transit systems with time-dependent demand: user equilibrium, system optimum, and management strategy. *Transportation Research Part B: Methodological*, 92, 234-252.
- Andelmin, J., & Bartolini, E. An Exact algorithm for the green vehicle routing problem. *Transportation Science*, 51(4), 1288-1303 (2017)
- Androutsopoulos, K. N., Zografos, K. G.: An integrated modelling approach for the bicriterion vehicle routing and scheduling problem with environmental considerations. *Transportation Research Part C: Emerging Technologies*, 82, 180-209 (2017)
- Barth, M., Boriboonsomsin, K.: Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment*, 14(6), 400-410 (2009)
- Braekers, K., Caris, A., Janssens, G.K.: Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67, 166–186 (2014)
- Braekers, K., Kovacs, A.: A multi-period dial-a-ride problem with driver consistency. A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological*, 94, 355-377 (2016)
- Cordeau, J.-F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B*, 37(6), 579–594 (2003)
- Cordeau, J.-F.: A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3), 573-586 (2006)

- Demir, E., Bektas, T., Laporte, G.: An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2), 346–359 (2012)
- Demir, E., Bektas, T., Laporte, G.: A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237, 775–793 (2014)
- Delaware Valley Regional Planning Commission (DVRPC) (2011). Overview of Challenges and Opportunities for Alternative Fuel Vehicles in the Delaware Valley. Available on <https://www.dvrpc.org/reports/10055B.pdf>
- Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1), 100-114 (2012)
- Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., Stobbe, M.: A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research*, 259(3), 972-991 (2017)
- Fuller, M. Wireless charging in California: Range, recharge, and vehicle electrification. *Transportation Research Part C: Emerging Technologies*, 67, 343-356 (2016)
- Goeke, D., & Schneider, M. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1), 81-99 (2015)
- Goldberg, D. E., & Holland, J. H. Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99 (1988)
- Hiermann, G., Hartl, R. F., Puchinger, J., & Vidal, T. Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1), 235-248 (2019)
- Ho, S. C., Szeto, W. Y., Kuo, Y. H., Leung, J. M., Petering, M., Tou, T. W.: A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111, 395-421 (2018).
- Koç, Ç., & Karaoglan, I. The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, 39, 154-16 (2016)
- Leung, S. C., Zhang, Z., Zhang, D., Hua, X., & Lim, M. K. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*, 225(2), 199-210 (2013)
- Lim, A., Zhang, Z., Qin, H.: Pickup and delivery service with manpower planning in Hong Kong public hospitals. *Transportation Science*, 51(2), 688-705 (2016)
- Lin, S.: Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44, 2245–2269 (1965)
- Liu, M., Luo, Z., Lim, A.: A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, 81, 267-288 (2015)
- Masmoudi, M. A., Hosny, M., Braekers, K., Dammak, A.: 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 (2016)
- Masmoudi, M. A., Braekers, K., Masmoudi, M., Dammak, A.: A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. *Computers & Operations Research*, 81, 1-13 (2017)
- Masmoudi, M. A., Hosny, M., Demir, E., & Cheikhrouhou, N. A study on the heterogeneous fleet of alternative fuel vehicles: Reducing CO₂ emissions by means of biodiesel fuel. *Transportation Research Part D: Transport and Environment*, 63, 137-155 (2018a)
- Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., & Cheikhrouhou, N. The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research Part E: Logistics and Transportation Review*, 118, 392-420 (2018b)
- Molenbruch, Y., Braekers, K., Caris, A.: Typology and literature review for dial-a-ride problems. *Annals of Operations Research*, 1-31 (2017)
- Muelas, S., LaTorre, A., Peña, J. M.: A variable neighborhood search algorithm for the optimization of a dial-a-ride problem in a large city. *Expert Systems with Applications*, 40(14), 5516-5531 (2013)
- NAEI. Emission factors. Technical report, National Atmospheric Emissions Inventory, United Kingdom. <http://naei.defra.gov.uk/data/emission-factors> (2012) Last accessed on 10/06/2016.
- Parragh, S. N., Doerner, K. F., Hartl, R. F.: Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 37(6), 1129-1138 (2010)
- Parragh, S. N.: Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19(5), 912-930 (2011)
- Parragh, S. N., Cordeau, J. F., Doerner, K. F., Hartl, R. F.: Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum*, 34(3), 593-633 (2012)
- Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8), 2403–2435 (2007)
- Potvin, J. Y., Rousseau, J. M.: An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, 46(12), 1433-1446 (1995)
- Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12), 1985-2002 (2004)

- Réseau de transport de la Capitale (RTC). Available on http://ocpm.qc.ca/sites/ocpm.qc.ca/files/document_consultation/3.1_anglais_ocpm_fossil_fuels_en_1.pdf
- Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* (4), 455–472 (2006a)
- Ropke, S., Pisinger, D. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3),750–775 (2006b)
- Salehi, M., Jalalian, M., Siar, M. M. V.: Green transportation scheduling with speed control: trade-off between total transportation cost and carbon emission. *Computers & Industrial Engineering*, 113, 392-404 (2017)
- Sassi, O., Cherif, W. R., & Oulamara, A. (2014). Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. Technical Report . <https://hal.archives-ouvertes.fr/hal-01083966>
- Savelsbergh, M. W.: The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2), 146-154 (1992)
- Schneider, M., Stenger, A., & Goeke, D. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4), 500-520 (2014)
- Sevaux, M., Dauzère-Pérès, S.: Genetic algorithms to minimize the weighted number of late jobs on a single machine. *European journal of operational research*, 151(2), 296-306 (2003)
- Société de transport de Montréal (STM). Available on: <http://www.stm.info/en/press/press-releases/2012/the-stm-announces-the-purchase-of-regular-biodiesel-electric-buses>
- Thériault, M. (2005). La gestion du transport de personnes à mobilité réduite (Doctoral dissertation, Université Laval).
- Toro, E. M., Franco, J. F., Echeverri, M. G., Guimarães, F. G.: A multi-objective model for the green capacitated location-routing problem considering environmental impact. *Computers & Industrial Engineering*, 110, 114-125 (2017)
- Transport Adapté du Québec Métro Inc. (TAQM). Available on: <https://www.accesstotourisme.gc.ca/1.aspx?lang=fr>
- U.S. EIA.: Alternative Vehicle Fuels. Available on <https://www.epa.gov/greenvehicles/alternative-vehicle-fuels> (2013). Last accessed on 19/01/2017.
- Verma, P., & Sharma, M. P. Review of process parameters for biodiesel production from different feedstocks. *Renewable and Sustainable Energy Reviews*, 62, 1063-1071 (2016)
- Wong, K.I., Bell, M.G.H.: Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research*, 13 (3), 195–208.
- Xiang, Z., Chu, C., Chen, H.: A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, 174(2), 1117–1139 (2006)
- Xiao, Y., & Konak, A. A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem. *Journal of Cleaner Production*, 167, 1450-1463 (2017)
- Yavuz, M.: An iterated beam search algorithm for the green vehicle routing problem. *Networks*, 69(3), 317-328 (2017)
- Zhang, Z., Liu, M., Lim, A.: A memetic algorithm for the patient transportation problem. *Omega*, 54, 60-71 (2015)
- Žulj, I., Kramer, S., & Schneider, M. A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem. *European Journal of Operational Research*, 264(2), 653-664 (2018)
- US DOE, Department of Energy (2018). Alternative Fuels and Advanced Vehicles. Available on <https://www.afdc.energy.gov/fuels/>.

Appendix

All parameters used in the CMEM model are given in Table 12 below.

Table 12
Parameters used in the CMEM of the MF-HDARP model

Notation	Description	Value
g	Gravitational constant (meter/second ²)	9.81
u	Air density (kilogram/meter ³)	1.2041
A	Frontal surface area of the vehicle(meter ²)	3.912
C_r	Coefficient rolling friction	0.01
C_d	Coefficient of aerodynamic drag	0.7
ξ	Fuel-to-air mass ratio	1
κ	Heating value of typical diesel fuel (kilojoules per gram)	44
e	Engine friction factor (kilojoules per revolution per liter)	0.2

N	Engine speed (revolution per second)	33
D	Engine displacement(liters)	5
λ	Factor converting the fuel rate (grams per second to liters per second)	737
η	Efficiency parameter for diesel engines	0.9
η_{tf}	Drive train efficiency	0.4
m	Vehicle mass (kilogram)	6,350
v^l	Lower vehicle speed (km/h)	20
v^w	Upper vehicle speed (km/h)	90

Table 13
Parameters used in our proposed algorithm

Algorithm	Description of parameters	Best value	Score
Hybrid	Number of users removed at each	$[u_{min}= 0.175.n; u_{max}= 0.35.n]$	Based on Pisinger and Ropke (2007)
ALNS	ALNS iteration (u)		Experimental results of Masmoudi et al. (2016)
	Number of iterations to update the	100	Experimental results of Masmoudi et al. (2016)
	Weights of operators (n_{seq})		Experimental results of Masmoudi et al. (2016)
	Roulette wheel parameter (r_p)	0.70	Experimental results of Masmoudi et al. (2016)
	P_d^0 used for removal operators	0.10	Experimental results of Masmoudi et al. (2016)
	P_d^0 used for insertion operators	0.125	Experimental results of Masmoudi et al. (2016)
	Score of a global better solution (π_1)	15	Our experimental result in Table 5
	Score of a better solution (π_2)	10	Our experimental result in Table 5
	Score of a worse solution (π_3)	5	Our experimental result in Table 5
	Initial temperature (T_{max})	25	Based on Leung et al.(2013)
	Cooling rate (α)	0.99975	Suggested by Ropke and Pisinger (2006a)