Enhanced multi-directional local search for the bi-objective heterogeneous vehicle routing problem with multiple driving ranges

Majid Eskandarpour, Djamila Ouelhadj, Sara Hatami, Angel Juan, Banafsheh Khosravi

Version of Record: https://www.sciencedirect.com/science/article/pii/S0377221719302127
Manuscript_9ae8170e541f9e68f4a3e057be81b7a6

Enhanced Multi-Directional Local Search for the Bi-Objective Heterogeneous Vehicle Routing Problem with Multiple Driving Ranges

Majid Eskandarpour[a,*], Djamila Ouelhadj[b], Sara Hatami[c], Angel A. Juan[c], Banafsheh Khosravi[b]

[a] *IÉSEG School of Management (LEM-CNRS 9221)*
[b] *Centre for Operational Research and Logistics, School of Mathematics and Physics, University of Portsmouth, Portsmouth, UK*
[c] *IN3 – Department of Computer Science, Universitat Oberta de Catalunya, Barcelona, Spain*

Abstract

The transportation sector accounts for a significant amount of greenhouse gas emissions. To mitigate this problem, electric vehicles have been widely recommended as green vehicles with lower emissions. However, the driving range of electric vehicles is limited due to their battery capacity. In this paper, a bi-objective mixed-integer linear programming model is proposed to minimise total costs (fixed plus variable) as well as CO_2 emissions caused by the vehicles used in the fleet for a Heterogeneous Vehicle Routing Problem with Multiple Loading Capacities and Driving Ranges (HeVRPMD). To solve the proposed model, an enhanced variant of Multi-Directional Local Search (EMDLS) is developed to approximate the Pareto frontier. The proposed method employs a Large Neighbourhood Search (LNS) framework to find efficient solutions and update the approximated Pareto frontier at each iteration. The LNS algorithm makes use of three routing-oriented destroy operators and a construction heuristic based on a multi-round approach. The performance of EMDLS is compared to MDLS, an Improved MDLS (IMDLS), non-dominated sorting genetic algorithm II (NSGAII), non-dominated sorting genetic algorithm III (NSGAIII), and the weighting and epsilon-constraint methods. Extensive experiments have been conducted using a set of instances generated from the Capacitated Vehicle Routing Problem benchmark tests in the literature. In addition, real data is utilised to estimate fixed and variable costs, CO_2 emissions, capacity, and the driving range of each type of vehicle. The results show the effectiveness of the proposed method to find high-quality non-dominated solutions.

Keywords: Routing, Multi-objective, Multi-directional Local Search, Electric Vehicles, Multiple Driving Ranges

*Corresponding author
Email address: m.eskandarpour@ieseg.fr (Majid Eskandarpour)

1. Introduction

A number of factors stimulate the use of eco-friendly means of transport including (Juan et al., 2016): *(i)* government's incentives for reducing the greenhouse gas emissions; *(ii)* high risk associated with oil-based products in terms of their availability and cost; *(iii)* possibility of the utilisation of these vehicles with lower purchase cost because of government subsidies; and *(iv)* advances in new generations of alternative energy technologies, which make them more competitive compared to Internal Combustion Engine Vehicles (ICEVs). For these reasons, most recently some companies have become interested to use more fuel-efficient vehicles such as electric and hybrid vehicles in countries such as Germany, France and the United Kingdom (Montoya et al., 2017; Browne et al., 2011). Therefore, many researchers and practitioners have focused on the transportation sector, which produces most of the greenhouse gas emissions compared to other sectors in the supply chain such as production and inventory (Dekker et al., 2012). As a consequence, Green Logistics and Green Vehicle Routing Problems (GVRP) have been increasingly receiving attention (Lin et al., 2014).

In the context of GVRP, Electric Vehicles (EVs) are likely to be used in mix fleets with Plug-in Hybrid Electric Vehicles (PHEVs) and ICEVs. For example, UPS announces its plan to expand the number of electric delivery trucks it operates in London from 50 to 70. According to UPS's director of sustainability, the company plans to test a new smart grid technology in London as part of a two-year testing project (Edie.net). However, there is a general agreement that driving range limitations impose a major challenge to the popularisation of electric vehicles.Thus, a report published by the Canadian Pollution Probe organisation states that: "As evident from the driver surveys, concerns about driving range limitations of electric vehicles combined with drivers' tendency to overestimate the distances that they actually drive limits how often drivers choose an electric vehicle for their day-to-day needs" (Council, 2015). Similarly, an article published in The Financial Times states that: "Without a technology breakthrough, battery electric vehicles are not expected to gain significant market share in the foreseeable future..." due to "... high purchase prices, driving range limitations and poor battery performance..." (Clark and Campbell, 2016). Hence, optimisation of routing plans considering these driving range limitations constitutes a necessary step in order to reduce the cost associated with the use of electric vehicles and promote their use in modern freight fleets.

Unlike EVs, the driving ranges of ICEVs and PHEVs are unlimited due to the fact that they can be refuelled at any fuel station along the route. Additionally, the loading capacity of ICEVs and PHEVs is more than EVs. Therefore, from an economic point of view, ICEVs and PHEVs are

more efficient. On the contrary, as stated before, EVs are more eco-friendly compared to ICEVs and PHEVs, and can result in a greener fleet. Hence, given two fleet configurations that could perform the requested delivery, the one using less combustion-engine vehicles and more EVs is considered to be greener. Using various vehicles in the fleet can be a challenging issue if the total costs (fixed plus variable), as well as environmental impacts, are considered simultaneously. Hence, it is highly desirable to provide trade-off solutions considering both environmental impact and monetary costs. Juan et al. (2014b) analyse how distance-based costs increase when a fleet consists of EVs, PHEVs and ICEVs. However, they have not studied the HeVRPMD model as a bi-objective problem, and therefore they are not able to provide the Pareto approximation set explicitly. The research work proposed in this paper addresses this gap.

In this paper, we propose a bi-objective model to integrate the impact of using a greener fleet on the monetary costs. The three primary contributions of this paper are as follows: *(i)* A bi-objective optimisation model is proposed to provide trade-off solutions between monetary costs and environmental impacts; *(ii)* EMDLS algorithm is developed, which is an enhanced variant of the Improved Multi-Directional Local Search (IMDLS) introduced by Lian et al. (2016) for the multi-objective consistent VRP; and *(iii)* Large Neighbourhood Search (LNS) is proposed to solve each of the two objectives. The operators used within LNS differ from the ones in the literature as they are based on each solution routes instead of its nodes. As regards IMDLS, it is an improved variant of the Multi-Directional Local Search (MDLS) proposed by (Tricoire, 2012) for general multi-objective optimisation problems. The effectiveness of EMDLS is demonstrated by comparing its performance with MDLS, IMDLS, non-dominated sorting genetic algorithm II (NSGAII), non-dominated sorting genetic algorithm III (NSGAIII) and two traditional methods, namely: ϵ-Constraints and Weighted Sum.

The rest of the paper is organised as follows. Section 2 provides a review of the literature on GVRP. Section 3 presents the mathematical optimisation model. The developed solution methods are described in Section 4. Section 5 presents the computational experiments and results. Lastly, several conclusions and some possible future research directions are given in Section 6.

2. Related literature review on GVRP

The transportation sector contributes significantly to the generation of GHG emissions. Therefore, it has become a tangible player for reducing these emissions (Tight et al., 2005). In recent years, GVRP has received an increasing amount of attention in the literature with the aim to reduce GHG emissions. In the literature of VRP and GHG emission, two relatively new streams can

be found. In the first category, studies focus on reducing conventional fuel consumption in order to minimise the CO_2 emissions. Whereas, in the second category, alternative fuel consumption is considered to fulfil the same objective. In some practical situations, when alternative fuel is used, a heterogeneous fleet of vehicles is required to serve the customers. This fact is also reflected in the literature where studies on alternative fuel consumption consider Heterogeneous Vehicle Routing Problem (HVRP). HVRPs are an extension of the classical VRP in which one needs to additionally decide about the composition of the heterogeneous fleet. Koç et al. (2016) provide an overview of significant research studies on HVRP in recent years.

The first category of studies seeks to minimise the amount of fuel consumed by a vehicle since CO_2 emissions are directly proportional to this value. The fuel consumption of a given vehicle can be influenced by three dominant factors including travel distance, truckload and vehicle speed (Elhedhli and Merrick, 2012). Consequently, researchers have investigated the environmental VRP from different perspectives (Lin et al., 2014). Many researchers considered the fuel consumption as a function of distance and vehicle load (Zhang et al., 2014; Xiao et al., 2012; Kara et al., 2007). Some researchers have looked into integrating traffic-related parameters, such as vehicle speed, load and acceleration into VRP. This has resulted in a so-called Pollution-Routing Problem (PRP), which consists of generating a set of routes to serve the customers and determining the speed of each vehicle on each route segment (Bektaş and Laporte, 2011). The primary goal of the PRP is to minimise the fuel consumption along with other criteria such as minimising travel distance and travel time (Demir et al., 2014; Eshtehadi et al., 2017; Franceschetti et al., 2017).

With the increasing prevalence of Alternative Fuel Vehicles (AFVs) and the extensive growth of EVs in recent years, the second category of studies on alternative fuel consumption has emerged. In addition, the role of refuelling and recharging has become more significant in GVRP problems. Erdoğan and Miller-Hooks (2012), to the best of our knowledge, provide the first GVRP model which considers recharging stations. New mathematical programming models for GVRPs to formulate refuelling AFVs can be found in Bruglieri et al. (2016); Koç and Karaoglan (2016); Madankumar and Rajendran (2018). Bruglieri et al. (2016) introduce a new mixed-integer linear programming model to minimise the total travel distance. They formulate the visits to the Alternative Fuel Stations (AFSs) with less numbers of variables through pre-computing a set of efficient AFSs which may be actually used in the optimal solution. Koç and Karaoglan (2016) also propose a new formulation of GVRP with less number of variables and constraints to minimise the total travel distance of AFVs with limited refuelling infrastructure. They show the merit of solving the problem to optimality by an exact solution approach based on Simulated Annealing (SA). Another work which investigates minimising total routing and refuelling costs of AFVs is a study on GVRP

with Pickups and Deliveries in a Semiconductor Supply Chain (G-VRPPD-SSC) by Madankumar and Rajendran (2018). Unlike most of the existing models for EVs, Goeke and Schneider (2015) incorporate a non-linear energy consumption model which considers speed, gradient and cargo load distribution.

Among the studies on EV routing, various heuristics and metaheuristics are implemented to consider when and where to charge EVs. Felipe et al. (2014) develop several solution methods including various constructive algorithms, local search heuristics and an SA framework in a study which considers cost and energy savings through partial recharges and usage of several recharge technologies. Hiermann et al. (2014) develop a hybrid variable neighbourhood search and tabu search heuristic to solve electric VRP with time windows. Vaz Penna et al. (2016) propose a hybrid iterative local search for the same problem. Bruglieri et al. (2017) introduce a three-phase matheuristic to solve the electric VRP with partial recharges, which combines an exact method with a Variable Neighbourhood Search local Branching (VNSB). The aim is to minimise hierarchically the number of EVs used and their total travelling, charging and waiting times. For a survey on the recent development of GVRP, we refer the readers to Bektaş et al. (2016).

It should be noted that it is not necessary to consider recharging decisions in our study as a vehicle route is completed before recharging is required. It is important to consider the driving range constraints as this operational limit is relevant to the current practice of EVs. Also, the development of recharging facilities throughout the road transportation networks might be only an option in the long run, despite the recent advances on EVs-related technology and infrastructure. Therefore, the travel range still remains as one of the main issues concerning the use of EVs in transportation. Hence, Juan et al. (2014b) introduce a new variant of VRP, called the VRP with Multiple Driving Ranges, in which vehicles are heterogeneous in terms of driving-ranges. Our study extends the previous research work by proposing the following: *(i)* a bi-objective HeVRPMD to minimise total cost and environmental impacts, and *(ii)* an efficient solution approach inspired by MDLS and IMDLS to provide a set of non-dominated solutions.

3. Problem definition and bi-objective optimisation model

In this section, we describe the proposed bi-objective HeVRPMD optimisation model, which considers a heterogeneous fleet of vehicles with loading capacities and driving ranges. This model extends the VRP with multiple driving ranges model proposed by Juan et al. (2014b) by considering a heterogeneous fleet of vehicles with respect to loading capacities.

Unlike the model proposed by Juan et al. (2014b), where the main goal is to minimise distance,

the proposed model seeks to minimise both the fixed and variable costs and environmental impacts of the fleet used to serve the customers. Indeed, both objectives stem from the real data for different types of vehicles. The fixed cost refers to the cost components such as capital investment over a constant period of time, tax, insurance and warranty costs; while the variable cost includes fuel/energy and driver costs. Inspired by the real data, both fixed and variable costs are defined as a function of distance units. Since CO_2 accounts for a predominant source of greenhouse gas emissions, and thanks to the data-accessibility, the environmental impact could be measured as the amount of CO_2 emissions produced per unit of distance for each vehicle type.

The bi-objective HeVRPMD optimisation model is a complete, directed graph $G = (N, A)$ consisting of a set N of $n+1$ nodes, $N = \{0, 1, \ldots, n\}$ and a set $A = \{(i, j) : i, j \in N, i \neq j\}$, which represents the arcs connecting pairs of nodes. Node 0 denotes the depot, where the vehicle fleets are located, and the remaining nodes represent the n customers. Each customer i has a known demand $q_i > 0$. We denote the distance from node i to node j by d_{ij}, where $d_{ij} = d_{ji} \geq 0$. In addition, there is a set K of k different types of vehicles, $K = \{1, 2, \ldots, k\}$. The number of vehicles for each type is unlimited. The total travel cost using vehicle of type l per each distance unit is denoted by f^l. Similarly, e^l represents the CO_2 emissions generated by vehicle type l per distance unit. Each vehicle of type l has a loading capacity Q^l and a maximum driving range T^l. Three different decision variables are used in the model: *(i)* a binary decision variable x_{ij}^l, which takes the value 1 if vehicle $l \in K$ travels from node i to j, and 0 otherwise; and *(ii)* two continuous decision variables u_i^l and v_i^l, which represent the cumulative amount of load carried and distance traveled, respectively, by vehicle $l \in K$ when leaving customer $i \in N \backslash \{0\}$. It should be noted that u_0^l and v_0^l are both set to zero when leaving the start depot node. The hard constraints are as follows: *(i)* each route starts and ends at the depot and is associated with a vehicle type; *(ii)* each customer belongs to exactly one route; and *(iii)* loading capacities and driving ranges of the vehicles are never exceeded. As mentioned earlier, the vehicles are heterogeneous both in loading capacity and driving range.

The first objective function of the optimisation model is defined in Equation 1. This function calculates the total fixed and variable costs of the used vehicles by multiplying the total cost per unit distance of each vehicle and the traveled distance by each vehicle l over all arcs $(i, j) \in A$:

$$minimise\ z_1 = \sum_{l \in K} \sum_{(i,j) \in A} f^l d_{ij} x_{ij}^l \tag{1}$$

Equation 2 is the second objective function minimising the CO_2 emissions generated by all the vehicles used in the fleet by multiplying the CO_2 emission per unit distance of each vehicle to the traveled distance by each vehicle l over all arcs $(i, j) \in A$:

$$minimise\ z_2 = \sum_{l \in K} \sum_{(i,j) \in A} e^l d_{ij} x_{ij}^l \tag{2}$$

The constraints of the HeVRPMD model are defined in Equations 3 to 9. Constraints 3 ensure that every customer is visited exactly once by a single vehicle:

$$\sum_{l \in K} \sum_{j \in N, i \neq j} x_{ij}^l = 1 \qquad \forall i \in N \backslash \{0\} \tag{3}$$

Constraints 4 guarantee the flow conservation for a given customer using a vehicle of type l:

$$\sum_{j \in N, i \neq j} x_{ij}^l - \sum_{j \in N, i \neq j} x_{ji}^l = 0 \qquad \forall i \in N \backslash \{0\}, l \in K \tag{4}$$

Constraints 5 ensure that the load of the vehicle in the next node j depends on the load of the vehicle in the previous node i as well as the demand of node j:

$$u_i^l \leq u_j^l - q_j x_{ij}^l + Q^l (1 - x_{ij}^l) \qquad \forall l \in K, i \in N, j \in N \backslash \{0\}, i \neq j \tag{5}$$

Constraints 6 ensure that load u_i^l is always greater than zero and less than the maximum capacity Q^l for a vehicle of type l:

$$0 \leq u_i^l \leq Q^l \qquad \forall l \in K, i \in N \backslash \{0\} \tag{6}$$

Constraints 7 and 8 ensure that a route length does not exceed the maximum range of vehicle l. Constraints 7 restrict the accumulated distance travelled at customer j (v_j) to be larger than the accumulated distance travelled at previous visited node i (v_i) plus the distance travelled from customer node i to customer j.

$$0 \leq v_i^l \leq v_j^l - d_{ij} x_{ij}^l + T^l (1 - x_{ij}^l) \qquad \forall l \in K, i \in N, j \in N \backslash \{0\}, i \neq j \tag{7}$$

Constraints 8 ensure that the current accumulated distance travelled to be smaller than the maximum driving range of vehicle type $l \in K$ minus the distance traveled from node $i \in N$ to node $j \in N$.

$$0 \leq v_i^l \leq T^l - d_{ij} x_{ij}^l \qquad \forall l \in K, \forall (i,j) \in N, i \neq j \tag{8}$$

Lastly, constraints 9 guarantee the binary conditions of the decision variables:

$$x_{ij}^l \in \{0, 1\} \qquad \forall l \in K, \forall (i,j) \in A \tag{9}$$

As discussed in Juan et al. (2014b), even small-scale instances for the single objective homogeneous (simplified) version of this problem are hard to solve in reasonable computation times using commercial optimisation packages such as CPLEX. Therefore, in the remaining of this paper, we have developed a meta-heuristic method to solve the problem and generate a set of non-dominated solutions.

4. Enhanced multi-directional local search to solve HeVRPDR

Multi-objective optimisation involves the optimisation of two or more conflicting objectives subject to a set of constraints. A common approach to compare two solutions in multi-objective optimisation is to consider a dominance rule, which means a solution dominates another one if it is better in at least one objective and not worse in all other objectives (Tricoire, 2012). A solution is Pareto-optimal if there does not exist any solution that dominates it. Multi-objective optimisation methods aim to find the Pareto-optimal set consisting of several trade-off solutions rather than a single optimal solution (Eberhart et al., 1996).

In the following sections, MDLS and IMDLS are firstly explained. We then introduce the proposed Enhanced Multi-Directional Local Search (EMDLS), as an enhanced version of MDLS and IMDLS to approximate the Pareto frontier. We complete this section with the description of the LNS framework and its components (e.g. destroy and repair operators), which is used as a local search in EMDLS.

4.1. MDLS and IMDLS methods

MDLS introduced by Tricoire (2012) is a meta-heuristic for multi-objective optimisation. The fundamental idea of this method relies on the concept of Pareto dominance. A neighbour solution x' of x is efficient if x' is better than x for at least one objective. Hence, to find efficient neighbour solutions of x, it is sufficient to search one direction at a time using single objective local search methods.

The initialisation of MDLS requires an initial set $\mathcal{F}$ of non-dominated solutions. At each iteration, a solution x from $\mathcal{F}$ is randomly selected and then for each objective, a corresponding local search method is employed to generate a neighbour solution x'. After that, the non-dominated set $\mathcal{F}$ is updated by merging solutions in $\mathcal{F}$ and the new neighbour solutions using the dominance rule, and deleting all dominated solutions. At the end of the algorithm, MDLS returns the set $\mathcal{F}$ of mutually non-dominated solutions.

Lian et al. (2016) proposed IMDLS which is different from MDLS in two aspects: (i) IMDLS limits the size of $\mathcal{F}$. The crowding distance introduced by Deb et al. (2002) is used to guide the selection of specific solutions in $\mathcal{F}$ in case the size of $\mathcal{F}$ exceeds the maximum size; (ii) at each iteration, the neighbourhood of all the solutions in $\mathcal{F}$ are explored, while MDLS explores only a single solution. Lian et al. (2016) have proved the effectiveness of IMDLS on the multi-objective consistent VRP in comparison to the state of the art methods such as NSGAII (Deb et al., 2002), NSGAIII (Deb and Jain, 2014) and the multi-objective evolutionary algorithm based on decomposition (Zhang and Li, 2007).

We propose to enhance the method further to approximate the Pareto frontier. EMDLS is different from MDLS and IMDLS in three aspects: *(i)* determination of the number of solutions to be explored at each iteration (α); *(ii)* selection of α solutions to be explored ($\mathcal{F}_\alpha$); and *(iii)* a local search is performed on each solution in $\mathcal{F}_\alpha$ for a new direction, so-called adaptive direction, as well as for each objective/direction. This adaptive direction is computed based on how far a solution is from the ideal point.

4.2. Enhanced Multi-Directional Local Search (EMDLS)

Details of EMDLS are formally described in Algorithm 1. Similar to IMDLS, the algorithm requires a set of non-dominated solutions $\mathcal{F}$ and its size limit $\mathcal{F}_{max}$ as inputs. The initial set of non-dominated solutions are generated simply by applying a multi-round approach with respect to each objective ($t \in T$). This approach is detailed in Section 4.3.2. A dominance rule is applied to the $|T|$ generated solutions to determine the initial set of non-dominated solutions. Contrary to MDLS and IMDLS, at each iteration, α neighbour solutions, so-called $\mathcal{F}_\alpha$, are explored with regard to each of the T objectives. It should be noted that a random solution is selected to be explored at the first iteration. The procedure to determine the number of solutions to explore at each iteration is explained in Section 4.2.1. Moreover, we take advantage of the crowding distance to identify α less crowded solutions $\mathcal{F}_\alpha$, which is explained in more detail in Section 4.2.2. After computing an adaptive direction for each solution $f \in \mathcal{F}_\alpha$ using the approach explained in Section 4.2.3, EMDLS performs a local search on all the solutions $\mathcal{F}_\alpha$ for the adaptive direction.

All the neighbourhood solutions are entered to a set $\mathcal{G}$, which is used to update $\mathcal{F}$. The neighbour solution is obtained using the LNS framework. If the size of $\mathcal{F}$ exceeds $\mathcal{F}_{max}$ then the crowding distance is employed to identify more-crowded solutions to be eliminated. The number of solutions remaining in $\mathcal{F}$ must not be greater than $\mathcal{F}_{max}$ after eliminating the more-crowded solutions. As Lian et al. (2016) mentioned, the resizing helps EMDLS to guide the search through less-crowded areas of the non-dominated set and therefore obtain a more diverse set of non-dominated solutions.

4.2.1. Number of solutions to explore

The number of solutions to explore at each iteration influences the convergence speed. Selecting one solution to explore at each iteration may result in a slow convergence speed as well as a poor diversity of the final approximated Pareto frontier (Lian et al., 2016). On the other hand, exploring all non-dominated solutions at each iteration might not be efficient. Particularly, in the early iterations of the search, there is a high chance that a non-dominated solution is dominated in the next iterations. We believe that the number of solutions to explore should be determined depending

Algorithm 1 EMDLS

Require: a set of non-dominated solutions $\mathcal{F}$ and its size limit $\mathcal{F}_{max}$

while the termination criterion is not satisfied **do**
  $\mathcal{G} \leftarrow \emptyset$
  $\alpha \leftarrow$ number of solutions to be explored
  $\mathcal{F}_\alpha \leftarrow$ select α number of solutions
  for every solution $f \in \mathcal{F}_\alpha$ **do**
    for objectives 1 **to** T **do**
      $\mathcal{G} \leftarrow \mathcal{G} \cup LS_t(f)$
    end for
    for the adaptive direction **do**
      compute the relative weight of each direction
      $\mathcal{G} \leftarrow \mathcal{G} \cup LS_{adaptive}(f)$
    end for
  end for
  update$(\mathcal{F}, \mathcal{G})$
  compute crowding distance for $f \in \mathcal{F}$
  determine the number of solutions to be explored α
  select α less crowded solutions $\mathcal{F}_\alpha$
  if $|\mathcal{F}| > \mathcal{F}_{max}$ **then**
    truncate$(\mathcal{F})$
  end if
end while
return Pareto set approximation $(\mathcal{F})$

on the quality of the current set of non-dominated solutions. In essence, fewer solutions should be selected if there is a high chance of finding new non-dominated solutions. We benefit from the search procedure of the variable neighbourhood search method to select the number of solutions α to explore at each iteration, with α set from 1 to $\mathcal{F}_{max}$. Suppose iteration k begins with $alpha = c$, where c is an integer number ranging from 1 to $\mathcal{F}_{max}$. If an improvement of the approximated Pareto frontier is not possible then $\alpha = c + 1$. Note that if $\alpha = \mathcal{F}_{max}$, then α does not change. If an improvement of the approximated Pareto frontier is found then $\alpha = 1$ at iteration $k + 1$.

4.2.2. Selection of solutions to explore

The selection of solutions from $\mathcal{F}$ to explore may lead to a better diversity of the non-dominated solutions. To this end, we benefit from the crowding distance introduced by Deb et al. (2002) to order the non-dominated solutions from the least crowded one to the most crowded one. Then, we perform local searches on α less crowded solutions to find efficient solutions. The biased roulette wheel selection principle described in Algorithm 2 selects α less crowded solutions. The biased roulette wheel selection method will allow diversifying the search. Hence, the solutions are first sorted in non-increasing order based on their crowding distance values $\mathcal{L}$. Then, a solution is

selected giving a higher probability to the first ones. The Parameter $\iota \geq 1$ determines the level of randomness: a low value of ι corresponds to higher randomness (Eskandarpour et al., 2017). In our experiments, $\iota = 10$. This process is repeated α times and all the α solutions are stored in $\mathcal{F}_\alpha$.

Algorithm 2 Biased roulette wheel selection principle

Require: A list $\mathcal{L}$ of solutions, sorted in non-increasing order.
Require: $\iota \geq 1$: a randomness parameter
Require: $\mathcal{F}_\alpha \leftarrow \emptyset$
while $\alpha > 0$ **do**
  Generate a random number ρ according to a continuous uniform distribution in $[0, 1)$
  Choose the solution $\mathcal{N}$ at position $\lceil \rho^\iota |\mathcal{L}| \rceil$ in $\mathcal{L}$
  $\mathcal{F}_\alpha \leftarrow \mathcal{F}_\alpha \cup \mathcal{N}$
  $\alpha = \alpha - 1$
  delete the solution $\mathcal{N}$ from $\mathcal{L}$
end while
return Set of solutions to be explored ($\mathcal{F}_\alpha$)

4.2.3. Adaptive direction

The important measures frequently used to describe how good is a Pareto set approximation are closeness to the Pareto-optimal set and coverage of a wide range of diverse solutions (Zitzler et al., 2003). As mentioned before, IMDLS performs a local search on all solutions in $\mathcal{F}_\alpha$ to find efficient solutions. However, there is a high chance to find other efficient intermediate solutions between those efficient solutions (Caballero et al., 2007). The idea is similar to the concept of path relinking method, which attempts to make the link between a guiding solution and an initial solution via some intermediate solutions. These intermediate solutions might be Pareto-optimal if both initial and guiding solutions are good enough (Glover et al., 2000). To improve both intensification and diversification, EMDLS computes an adaptive direction for a number of less crowded non-dominated solutions. To do so, for each objective, we use the objective value of the neighbourhood solution obtained by applying a single objective local search to calculate the closeness of the neighbourhood solution to the ideal point, and eventually to compute the adaptive direction.

Overall, we perform local searches for all the objective directions as well as the adaptive direction. Figure 1 illustrates the principle with an example of an bi-objective minimisation problem. $\mathcal{S}(f_1)$, $\mathcal{S}(f_2)$ and $\mathcal{S}(f_1, f_2)$ represent the relevant portions of efficient neighbourhood solution spaces around the given solutions in favour of the cost objective, environmental objective, and a compromise solution space, respectively.

In the proposed method, the adaptive direction for each solution (for instance, solution a in Figure 1) is determined as follows: $Z = \sum_{i=1}^{I} w_i f_i^{normal}$, where f_i^{normal} represents the normalised

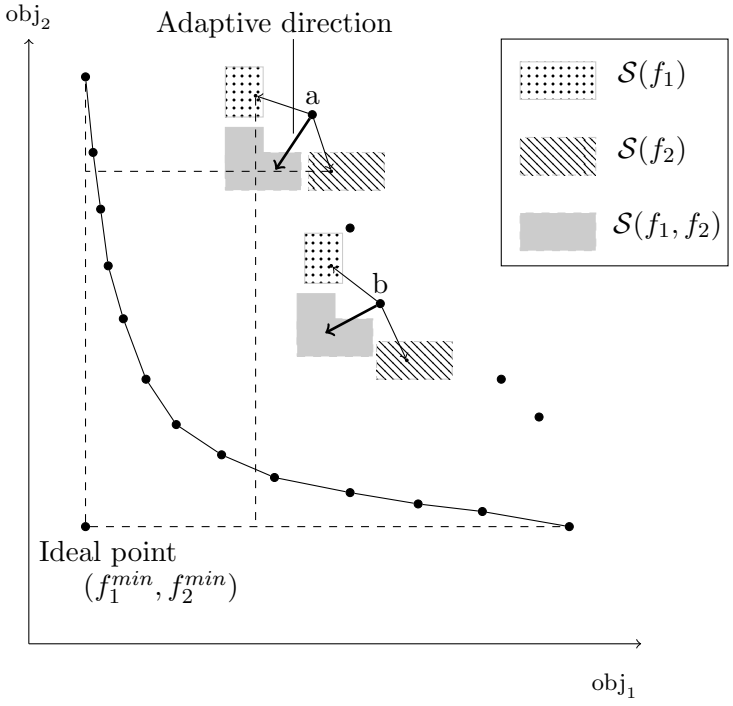

Figure 1: Relevant portions of efficient solution space

fitness of objective function i and is calculated as follows: $f_i^{normal} = \frac{f_i}{f_i^{min}}$. It should be noted that f_i is the objective value of the neighbourhood solution obtained by applying local search on the respective solution (e.g., solution a) for objective/direction i and f_i^{min} is the minimum value of objective function i at the current iteration. The weight of each objective is calculated in equation 10:

$$w_i = \frac{d_i}{\sum_{s=1}^{S} d_s} \qquad i = 1, \ldots, n \tag{10}$$

where d_i is the normalised distance between the i^{th} objective and the ideal point, computed in Equation 11:

$$d_i = \frac{f_i - f_i^{min}}{f_i^{min}} \qquad i = 1, \ldots, n \tag{11}$$

The weights will ensure that more importance is given to the objective with further distance to the ideal point in order to avoid fast convergence for one specific objective, which could result in poor diversity.

4.3. Single objective local search method: Large Neighbourhood Search (LNS)

EMDLS uses a single objective local search method to find a set of non-dominated solutions. We propose to use LNS as a single objective local search method due to its effectiveness in solving several combinatorial optimisation problems such as VRP (Pisinger and Ropke, 2007) and facility location (Eskandarpour et al., 2017). Its primary principle is to destroy and repair the current solution to find a neighbouring solution. In the destroy phase, a number of customers are targeted to be removed from the current solution based on some specific criteria. In the repair phase the removed customers are placed back into the solution with regard to some other criteria.

Algorithm 3 details the proposed LNS method for an iteration, with three destroy operators and a repair operator. The algorithm starts with a given solution to be minimised. Then, a destroy operator is selected randomly to choose a ratio $\mathcal{P}$ of the routes to be removed from $\mathcal{S}$, which results in $\mathcal{S}'$. The remaining $1-\mathcal{P}$ are saved as partial solutions in $\mathcal{S}''$. After realising the associated nodes of $\mathcal{S}'$ and storing them in $n_{s'}$, the repair operator is applied to $n_{s'}$ to obtain a set of routes denoted by $\mathcal{S}^*$. If the objective of the new generated routes is better than the removed ones, we merge the new routes $\mathcal{S}^*$ and $\mathcal{S}''$ to obtain a complete solution.

Algorithm 3 LNS

Require: solution $\mathcal{S}$ and its desired objective function
1: select a destroy operator randomly τ
2: $\mathcal{S}' \leftarrow$ apply τ on $\mathcal{S}$ to get a ratio $\mathcal{P}$ of the routes
3: $\mathcal{S}'' \leftarrow$ remaining $1-\mathcal{P}$ routes
4: $n_{s'} \leftarrow$ release the associated nodes of $\mathcal{S}'$
5: $\mathcal{S}^* \leftarrow$ apply repair operator on $n_{s'}$ to get some partial solutions
6: Let $Z_{\mathcal{S}^*}$ and $Z_{\mathcal{S}'}$ denote the objective values of solutions $\mathcal{S}^*$ and $\mathcal{S}'$, respectively
7: **if** $Z_{\mathcal{S}^*} < Z_{\mathcal{S}'}$ **then**
8: $\quad \mathcal{S} \leftarrow$ Merge($\mathcal{S}''$, $\mathcal{S}^*$)
9: **end if**
10: **return** solution $\mathcal{S}$

4.3.1. **Destroy operators**

Given a complete solution, the aim of the destroy operators is to first select a ratio $\mathcal{P}$ of the routes in the current solution to be discarded, and then release the associated customers (nodes) of the selected routes. Therefore, unlike the most destroy operators applied in the literature, we propose to destroy the solution based on the route specification rather than the customer node. Three destroy operators proposed are as follows:

- Random subset of routes: This operator randomly discards $\mathcal{P}$ routes of the current solution in order to diversify the search.

- Expensive subset of routes: This operator discards the routes with the highest cost/environmental impact. Thus, the routes are ranked in non-increasing order of the respective objective function value, then the first $\mathcal{P}$ routes are selected.The aim is to replace the expensive routes with the better ones.

- Clustered subset of routes: This operator discards the routes which are geographically located in the same regions. We first obtain the geographical center of route r consisting of N_r nodes as follows: $(C_r^x, C_r^y) = (\frac{\sum_{n \in N_r} x_r^n}{|N_r|}, \frac{\sum_{n \in N_r} y_r^n}{|N_r|})$. We then calculate the Euclidean distance between the geographical centres of each pair of routes, which represents the distance between each pair of routes. We rank the routes in a non-decreasing order of the Euclidean distance values and discard the first $\mathcal{P}$ routes.

4.3.2. Repair operator

The outcome of destroy operators are two sets: partially destroyed solution and released nodes $n_{s'}$. The aim of the repair operator is to rebuild a complete solution by generating new routes using the set $n_{s'}$ and then adding them to the partial solution. To this end, we use a multi-round approach inspired by the successive approximation method proposed by Juan et al. (2014a).

Given a set of nodes and ν types of vehicles, the multi-round approach splits the HeVRPMD into ν homogeneous VRPMD. Therefore, each round is subject to a homogeneous VRPMD consisting of a subset of nodes and an unlimited number of vehicles of the same type. Each vehicle type has its own driving range and loading capacity. Then, a routing algorithm is called at each round to generate routes for each homogeneous CVRP with a specific route length restriction. The driving range of the vehicle used at each round is set as a route length restriction. We employ the SR-GCWS-CS algorithm which is extensively tested on a set of CVRP standard benchmarks in Juan et al. (2011) as a routing algorithm. SR-GCWS-CS makes use of biased-randomization (Juan et al., 2013; Grasas et al., 2017) to enhance the behaviour of the classical savings heuristic (Clarke and Wright, 1964) by including randomness. The solution of each round is a set of routes which cover all the nodes. Eventually, a superior solution regarding the desired objective is added to the partial solutions.

5. Computational experiments

In this section, we describe the benchmark instances, the computational experiments, and the indicators used to assess the performance of the proposed bi-objective optimisation model and solution method against the state of the art ones.

5.1. Benchmark instances and experimental design

To evaluate the performance of the proposed model and solution method, we have used the 20 mid-size CVRP instances between 22 to 135 nodes proposed by Juan et al. (2014b) ; and 5 other instances between 151 to 420 nodes. We have selected instances, which provide detailed information on routes for the optimal or pseudo-optimal solution. In addition, the distance between the furthest node to the depot and the depot must be less than the maximum driving range of the vehicles. This will allow all nodes to be visited by at least one type of vehicle. Moreover, we have considered a variety in specifications such as the vehicle capacity, the location of the depot with regard to the customer nodes (in a corner or in the centre) and the topology of the customer nodes. Table 1 provides an overview of the test instances used in the experiments. It should be noted that the last column shows the distance between the furthest node and the depot for each instance.

Table 1: Test instances used in our experiments

Instance name	# nodes	Capacity	Minimum # of vehicles needed	Depot's location	Nodes' topology	Maximum distance
A-n32-k5	32	100	5	Corner	Randomly	101
A-n38-k5	38	100	5	Not Center	Randomly	75
A-n65-k9	65	100	9	Not Center	Randomly	74
A-n80-k10	80	100	10	Corner	Randomly	126
B-n50-k7	50	100	7	Center	Cluster	65
B-n52-k7	52	100	7	Not Center	Cluster	71
B-n57-k9	57	100	9	Corner	Cluster	111
B-n78-k10	78	100	10	Not Center	Cluster	90
E-n22-k4	22	6000	4	Center	Randomly	49
E-n30-k3	30	4500	3	Center	No clear pattern	69
E-n51-k5	51	160	5	Center	Randomly	87
E-n76-k10	76	140	10	Center	Randomly	43
E-n76-k14	76	100	14	Center	Randomly	43
F-n135-k7	135	2210	7	Not center	Real case data	147
Golden-17	240	200	22	Center	Concentric pointed star	20
Golden-19	360	200	33	Center	Concentric pointed star	31
Golden-20	420	200	38	Center	Concentric pointed star	38
M-n101-k10	101	200	10	Center	Cluster	59
M-n121-k7	121	200	7	Corner	Cluster	99
M-n151-k12	151	200	12	Center	Randomly	50
M-n200-k17	200	200	17	Center	Randomly	50
P-n50-k10	50	100	10	Center	Randomly	37
P-n55-k15	55	70	15	Center	Randomly	37
P-n70-k10	70	135	10	Center	Randomly	43
P-n76-k5	76	280	5	Center	Randomly	43

Three types of vehicles are considered in the experiments: *(i) 2018 Ford Transit Connect Wagon LWB FWD*, ICEV vehicle L and have an autonomy of 348 mile distance units, capacity of 243.936 in^3, total (fixed and variable) cost per mile of 0.64$ and variable emission of 404 grams of CO_2 per mile; *(ii) 2017 Toyota Chrysler Pacifica* , PHEV vehicle M and have an autonomy of 570 mile distance units, capacity of 200.922 in^3, total (fixed and variable) cost per mile of 0.90$ and variable

emission of 220 grams of CO_2 per mile; and *(iii) 2012 Azure Dynamics Transit Connect Electric Van*, an electric vehicle S and have an autonomy of 96 mile distance units, capacity of 185.973 in^3, total (fixed and variable) cost per mile of 1.14$ and variable emission of 190 grams of CO_2 per mile.

The total range that each vehicle could drive with full gasoline tank or battery is obtained from (Fueleconomy.gov). The loading capacity of vehicles are obtained through each manufacturer website. The total cost (fixed and variable) and CO_2 emissions of each vehicle are calculated from the websites provided by (Financial mentor, 2018) and (Union of Concerned scientists, 2018), respectively. Accordingly, to set up the loading capacity coefficient of the vehicles, we suppose that the fixed capacity in VRP instances, Q_0, corresponds to a vehicle of type M. We also assume that a vehicle of type S has a lower loading capacity than a vehicle of type M. Accordingly, a vehicle of type M has a lower loading capacity than a vehicle of type L. In order to calculate the capacity coefficient of all vehicle types, we assume that the capacity coefficient of vehicle type M is equal to 1 and the other coefficients are calculated with respect to Q_0. Therefore, the capacity coefficient associated with vehicles of types S and L is set to 0.925 and 1.214, respectively.

The performance of EMDLS is compared to MDLS, IMDLS, two state-of-the-art population-based approaches NSGAII and NASGAIII, and two classical multi-objective optimisation methods ϵ-Constraint and Weighted Sum. MDLS and IMDLS employ the LNS already explained in Subsection 4.3. For these methods, the $\mathcal{P}$ parameter which indicates the portion of routes to be removed from the current solution is set to 0.4 based on numerous computational experiments. NSGAII and NASGAIII use the classical Ordered Crossover proposed by Koç et al. (2015) and the generalised mutation operator proposed by Matei et al. (2015) for the heterogeneous VRP. Both crossover and mutation operators are applied to the parent solutions with probability 1. The population size in NSGAII is set to 100, and the population size for NSGAIII is determined by the size of the reference point set defined by the user (Deb and Jain, 2014). The size of the reference point set is highly related to the desired number of non-dominated solutions. Hence, both reference point set and population size are set to 31.

Regarding the ϵ-Constraint and Weighted Sum methods, we use the Multi-Round heuristic introduced by Juan et al. (2014a) to find a set of non-dominated solutions. The $\mathcal{P}$ parameter is set to 0.6 for these methods. For ϵ-Constraint method, we convert the second objective into a constraint by imposing an upper bound ϵ. For the Weighted Sum method, the objective is to minimise the sum of a normalised weighted bi-objective function. $\mathcal{F}_{max}$ is set to 30 in NSGAII, NSGAIII, IMDLS and EMDLS in order to allow the algorithms to have a fair balance between exploring the search space and the computational time. To conduct a fair comparative analysis, the ϵ-Constraint and Weighting Sum methods are run 30 times with different values for ϵ and objective

weights, respectively. The same approach used by Demir et al. (2014) is employed to find the value of ϵ and objective weights for each run. The time limit is set to 1000 seconds as a stopping criteria for NSGAII, NSGAIII, MDLS, IMDLS and EMDLS. In practice, 1000 should be a reasonable time to solve an operational problem such as VRP. For the ϵ-Constraint and Weighting Sum methods, the time limit is set to 40 for each ϵ and a set of weights. For all methods, we conduct ten runs for each instance using different seeds for the random number generation. All methods are implemented using the Java programming language and run on an Intel® Core™i5-4430 CPU 3.00GHz with 8GB RAM.

5.2. Solution quality assessment indicators

In the computational experiments, we make use of four classical performance quality indicators. Each quality assessment indicator is explained below;

- *Hypervolume:* The hypervolume $I_{h\mathfrak{v}}$ is the area of the union of all hypercubes formed by each non-dominated solution $i \in A$ and a reference point r, which can be formulated as follows (Lwin et al., 2014): $I_{h\mathfrak{v}} = volume(\bigcup_i^{|A|} c_i)$, where c_i represents a hypercube from solution i and the reference point r. It is necessary to normalise the objective values if their scales are not the same (Lian et al., 2016). The larger the value of the indicator, the better is the set of non-dominated solutions.

- *Epsilon:* The unary epsilon indicator introduced by Zitzler et al. (2003) provides a value indicating how far are two sets of non-dominated solutions from each other. For a minimisation problem with k objectives, a non-dominated solution with the objective vector $z^1 = (z_1^1, z_2^1, \ldots, z_k^1) \in Z$ is said to $\epsilon-$dominate another non-dominated solution with objective vector $z^2 = (z_1^2, z_2^2, \ldots, z_k^2) \in Z$, if and only if, there exists an $\epsilon > 0$ such that $z_i^1 \leq \epsilon \times z_i^2$, $\forall 1 \leq i \leq k$. With this indicator, the smallest value is 1, and smaller values are better than higher ones.

- *Ratio:* The Ratio of a set of non-dominated solutions, A, refers to the solutions from set A not dominated by any solution in B (Zitzler et al., 2000). This ratio $I_{\mathcal{R}}(A, B)$ is computed as follows: $I_{\mathcal{R}}(A, B) = \frac{|A - \{X \in A | \exists Y \in B : Y \succeq X\}|}{|A|}$, where $Y \succeq X$ means solution X is dominated by solution Y. The largest value is 1 and larger values imply better performance. It should be noted that $I_{\mathcal{R}}(A, B)$ is not necessarily equal to $1 - I_{\mathcal{R}}(B, A)$.

- *Inverted Generational Distance (IGD):* Let $\mathcal{R}$ be a reference set and let A be a set of non-dominated solutions obtained by an algorithm. The average distance from $\mathcal{R}$ to A is calculated as follows: $I_{IGD}(A, \mathcal{R}) = \frac{\sum_{v \in \mathcal{R}} d(v,A)}{|\mathcal{R}|}$, where $d(v, A)$ represents the minimum Euclidean distance between each solution in $\mathcal{R}$ and the nearest solution in A. A lower value of IGD suggests the good convergence of solutions to the Pareto front as well as a good diversity over the Pareto front (Rakshit and Konar, 2015). For a fair comparison over all instances, we use normalised values of the objectives for this indicator.

5.3. Computational results

This section presents a comparative analysis of the methods developed with the ones from the literature using the assessment indicators mentioned in the previous section. Concerning each assessment indicator, the result of each method is compared with a reference set $\mathcal{R}$ for each instance, which is the union of replications of the methods and removing all dominated solutions.Each method is replicated 10 times and the average overall replications is reported for each assessment indicator. Table 2 shows the number of Pareto solutions. The first column represents the instance number. The next columns indicate the average number of Pareto solutions over 10 runs for each method, respectively. There are three instances (B-n57-k9, E-n30-k3 and M-n121-k7) for which all the methods have failed to generate a considerable amount of non-dominated solutions. MDLS produces the maximum average number of Pareto solutions since the $\mathcal{F}_{max}$ is infinite. However, the weighting method produces the minimum number of Pareto solutions with an average of 2.9. This is due to the fact that the weighting method is able to produce only supported solutions, which can be found by solving the associated single-objective projected problem using a weight vector (Tricoire, 2012). The ϵ-Constraint produces more Pareto solutions than the Weighting method but most of them are dominated by those provided by the other methods.

Table 3 shows the comparison between the ratios provided by each method. EMDLS yields the best performance compared to the other solution methods with an average of 0.49. NSGAII, NSGAIII, IMDLS and MDLS provide relatively similar results with an average of 0.35. The Weighting method yields the worst performance due to the fact that it is not able to provide as many good non-extreme Pareto solutions.

Table 4-5 present the results of hypervolume and epsilon indicators for all the instances, respectively. The first column indicates the instance name and the next columns show the average hypervolume and epsilon indicators over 10 runs for all the methods. Table 4 illustrates that most of the methods have achieved relatively the same results, except the epsilon constraint and weighting methods. MDLS yields slightly better performance in terms of the hypervolume indicator with an

Table 2: Comparison of the number of non-dominated solutions of the different methods

Instance	ϵ-Constraint	Weighting Sum	NSGAII	NSGAIII	MDLS	IMDLS	EMDLS
A-n32-k5	2	2.4	3.8	4.2	3	3.4	3
A-n38-k5	2.9	3	17.3	20.8	12.8	12.5	11.4
A-n65-k9	2.8	2.2	30	30	25.5	26.1	22.3
A-n80-k10	2	2	13.2	16.4	8.4	8.3	8.9
B-n50-k7	4.8	2.8	11.1	7.6	5.3	5.5	5
B-n52-k7	4.6	3.1	30	30	38.9	30	30
B-n57-k9	2	2	2	2	2	2	2.1
B-n78-k10	2.7	2	30	30	44.9	30	30
E-n22-k4	6	2.2	18.6	21.8	10.8	13.5	10
E-n30-k3	2	2	2	2	2	2	2
E-n51-k5	7.8	4.1	30	30	54.7	30	30
E-n76-k10	7.9	3.6	30	30	164.8	30	30
E-n76-k14	8	3.3	30	30	205.5	30	30
F-n135-k7	4	3.2	30	30	62.6	30	30
Golden-17	7.4	4	30	30	245.1	30	30
Golden-19	6.7	4.2	30	30	321.9	30	30
Golden-20	7.4	4.1	30	30	296.7	30	30
M-n101-k10	6.6	2	30	30	69.5	30	30
M-n121-k7	2	2	3.5	3.3	3.2	3.3	3.2
M-n151-k12	4.8	2.3	30	30	112.6	30	30
M-n200-k17	3.4	2	30	30	145.8	30	30
P-n50-k10	8	3.5	30	30	172.5	30	30
P-n55-k15	7.3	4.4	30	30	350.2	30	30
P-n70-k10	8.5	2.7	30	30	159.9	30	30
P-n76-k5	8.5	3.3	30	30	56.8	30	30
Average	5.2	2.9	23.2	23.5	103.0	22.3	21.9

Table 3: Comparison of the ratios of the different methods

Instance	ϵ-Constraint	Weighting Sum	NSGAII	NSGAIII	MDLS	IMDLS	EMDLS
A-n32-k5	0.55	0.13	0.31	0.20	0.33	0.38	0.33
A-n38-k5	0.35	0.15	0.26	0.43	0.54	0.63	0.51
A-n65-k9	0.43	0.37	0.24	0.48	0.33	0.35	0.07
A-n80-k10	0.50	0	0.46	0.25	0.09	0.21	0.35
B-n50-k7	0.21	0.37	0.16	0.33	0.81	0.76	0.86
B-n52-k7	0.30	0.00	0.45	0.43	0.69	0.71	0.78
B-n57-k9	0.60	0.50	0.50	0.50	0.50	0.50	0.50
B-n78-k10	0.28	0.36	0.55	0.44	0.61	0.66	0.65
E-n22-k4	0.48	0.27	0.10	0.31	0.91	0.95	0.99
E-n30-k3	0.30	0.45	0.50	0.50	0.50	0.50	0.50
E-n51-k5	0.28	0.08	0.36	0.29	0.60	0.58	0.52
E-n76-k10	0.05	0.06	0.28	0.50	0.45	0.24	0.48
E-n76-k14	0.08	0.03	0.31	0.46	0.26	0.16	0.54
F-n135-k7	0	0.03	0.41	0.25	0.09	0.08	0.16
Golden-17	0.05	0.13	0.20	0.50	0.10	0.09	0.30
Golden-19	0	0	0.30	0.23	0.11	0.00	0.44
Golden-20	0.04	0	0.27	0.18	0.11	0.01	0.15
M-n101-k10	0.20	0.13	0.41	0.38	0.62	0.57	0.87
M-n121-k7	0	0	0.46	0.19	0.18	0.17	0.35
M-n151-k12	0.12	0	0.47	0.19	0.10	0.11	0.33
M-n200-k17	0	0	0.38	0.46	0.10	0.11	0.67
P-n50-k10	0.03	0	0.25	0.51	0.47	0.29	0.59
P-n55-k15	0.04	0	0.50	0.16	0.33	0.11	0.46
P-n70-k10	0.04	0	0.11	0.43	0.43	0.29	0.38
P-n76-k5	0	0	0.30	0.33	0.24	0.32	0.37
average	0.20	0.12	0.34	0.36	0.38	0.35	0.49

average of 0.495 over all instances. EMDLS produces the second best performance with an average of 0.485. MDLS outperforms EMDLS due to the large number of Pareto solutions generated by MDLS. In essence, if the Pareto frontiers of two given methods are close enough, the one with more Pareto solutions yields a better result regarding hypervolume. With respect to the epsilon indicator, the average results provided by the methods are very close to each other. Surprisingly, in contrast to the other indicators, the weighting method performs slightly better than the other methods due to its capability to produce good extreme solutions. However, NSGAII and NSGAIII provide the worst results due to their limitation in producing very good extreme solutions. Lastly, table 6 presents the results for the IGD indicator. EMDLS provides the best results with an average of 0.03 over all instances. This proves that the set of non-dominated solutions provided by EMDLS are spread all over the approximated Pareto frontier. NSGAII, NSGAIII and MDLS obtain relativity the same performances. As expected, epsilon constraint and weighting methods did not provide good results as the set of non-dominated solutions generated by these methods were not be able to truly approximate Pareto front.

Table 4: Comparison of the hypervolumes of the different methods

Instance	ϵ-Constraint	Weighting Sum	NSGAII	NSGAIII	MDLS	IMDLS	EMDLS
A-n32-k5	0.336	0.274	0.479	0.356	0.441	0.362	0.440
A-n38-k5	0.389	0.307	0.529	0.383	0.491	0.411	0.491
A-n65-k9	0.324	0.258	0.391	0.563	0.436	0.356	0.434
A-n80-k10	0.328	0.262	0.474	0.434	0.446	0.366	0.435
B-n50-k7	0.367	0.296	0.347	0.439	0.459	0.379	0.459
B-n52-k7	0.359	0.297	0.436	0.586	0.460	0.380	0.459
B-n57-k9	0.323	0.262	0.532	0.471	0.412	0.332	0.398
B-n78-k10	0.343	0.262	0.424	0.513	0.460	0.380	0.460
E-n22-k4	0.361	0.267	0.375	0.403	0.455	0.380	0.461
E-n30-k3	0.323	0.280	0.562	0.398	0.395	0.315	0.395
E-n51-k5	0.473	0.368	0.431	0.499	0.574	0.494	0.567
E-n76-k10	0.428	0.333	0.572	0.457	0.553	0.471	0.536
E-n76-k14	0.439	0.336	0.557	0.406	0.563	0.480	0.536
F-n135-k7	0.346	0.277	0.582	0.440	0.465	0.384	0.455
Golden-17	0.430	0.373	0.537	0.409	0.576	0.493	0.550
Golden-19	0.422	0.376	0.339	0.489	0.564	0.475	0.543
Golden-20	0.428	0.375	0.577	0.333	0.560	0.472	0.542
M-n101-k10	0.357	0.284	0.398	0.405	0.453	0.372	0.449
M-n121-k7	0.310	0.279	0.421	0.504	0.407	0.327	0.407
M-n151-k12	0.354	0.276	0.354	0.586	0.494	0.414	0.489
M-n200-k17	0.333	0.264	0.450	0.422	0.487	0.404	0.469
P-n50-k10	0.429	0.327	0.447	0.598	0.548	0.464	0.528
P-n55-k15	0.428	0.331	0.525	0.399	0.548	0.462	0.520
P-n70-k10	0.432	0.297	0.578	0.350	0.553	0.470	0.532
P-n76-k5	0.485	0.388	0.579	0.515	0.582	0.503	0.571
average	0.382	0.306	0.476	0.454	0.495	0.414	0.485

Overall, it can be seen from the comparative analysis that EMDLS outperforms the other methods with regard to the ratio and IGD indicators. On the other hand, the weighting method and epsilon constraint yield the worst performance. Regarding IGD indicator, the size of the reference set

Table 5: Comparison of the epsilons of the different methods

Instance	ϵ-Constraint	Weighting Sum	NSGAII	NSGAIII	MDLS	IMDLS	EMDLS
A-n32-k5	1.797	1.797	2.040	2.279	1.818	1.818	1.818
A-n38-k5	1.800	1.800	1.939	2.105	1.801	1.801	1.801
A-n65-k9	1.814	1.814	2.195	2.061	1.814	1.814	1.814
A-n80-k10	1.885	1.885	2.086	2.103	1.890	1.890	1.890
B-n50-k7	1.836	1.836	2.235	1.946	1.837	1.837	1.837
B-n52-k7	1.803	1.803	2.094	1.906	1.815	1.815	1.815
B-n57-k9	1.958	1.958	2.284	2.084	1.959	1.959	1.959
B-n78-k10	1.862	1.862	2.060	2.218	1.862	1.862	1.862
E-n22-k4	1.764	1.778	1.908	1.861	1.764	1.764	1.764
E-n30-k3	1.662	1.662	2.203	2.004	1.743	1.743	1.743
E-n51-k5	2.093	2.096	2.295	2.134	2.093	2.093	2.093
E-n76-k10	2.034	2.040	1.885	2.165	2.033	2.033	2.032
E-n76-k14	2.111	2.109	2.280	2.111	2.108	2.109	2.110
F-n135-k7	1.768	1.763	2.033	2.079	1.789	1.789	1.789
Golden-17	2.197	2.214	2.213	2.061	2.145	2.116	2.113
Golden-19	2.175	2.198	1.929	2.048	2.162	2.157	2.118
Golden-20	2.182	2.206	1.997	1.971	2.172	2.163	2.139
M-n101-k10	1.698	1.680	2.265	2.116	1.689	1.689	1.690
M-n121-k7	1.695	1.695	1.875	2.174	1.727	1.727	1.727
M-n151-k12	1.857	1.856	2.030	2.071	1.863	1.863	1.863
M-n200-k17	1.820	1.820	2.055	2.225	1.830	1.830	1.830
P-n50-k10	2.075	2.067	2.065	1.882	2.065	2.065	2.065
P-n55-k15	2.192	2.195	1.919	2.244	2.191	2.191	2.191
P-n70-k10	2.034	2.037	2.288	2.183	2.023	2.020	2.031
P-n76-k5	2.269	2.151	2.181	1.878	2.256	2.256	2.256
average	1.935	1.933	2.094	2.076	1.938	1.936	1.934

Table 6: Comparison of IGD values of the different methods

Instance	ϵ-Constraint	Weighting Sum	NSGAII	NSGAIII	MDLS	IMDLS	EMDLS
A-n32-k5	0.21	0.28	0.07	0.13	0.03	0.02	0.02
A-n38-k5	0.17	0.25	0.06	0.07	0.04	0.04	0.04
A-n65-k9	0.11	0.30	0.01	0.03	0.01	0.03	0.02
A-n80-k10	0.27	0.32	0.05	0.13	0.10	0.14	0.06
B-n50-k7	0.03	0.22	0.01	0.01	0.01	0.01	0.01
B-n52-k7	0.14	0.21	0.01	0.02	0.01	0.07	0.01
B-n57-k9	0.09	0.24	0.10	0.36	0.09	0.25	0.09
B-n78-k10	0.11	0.16	0.01	0.01	0.01	0.01	0.01
E-n22-k4	0.09	0.22	0.05	0.02	0.02	0.01	0.07
E-n30-k3	0.23	0.36	0.31	0.23	0.19	0.19	0.19
E-n51-k5	0.15	0.25	0.09	0.08	0.02	0.11	0.01
E-n76-k10	0.11	0.27	0.03	0.03	0.02	0.07	0.01
E-n76-k14	0.10	0.28	0.01	0.04	0.04	0.06	0.01
F-n135-k7	0.32	0.25	0.03	0.08	0.03	0.11	0.03
Golden-17	0.20	0.28	0.03	0.03	0.03	0.07	0.01
Golden-19	0.23	0.35	0.06	0.02	0.03	0.08	0.01
Golden-20	0.19	0.35	0.06	0.03	0.04	0.08	0.01
M-n101-k10	0.13	0.23	0.01	0.03	0.02	0.05	0.01
M-n121-k7	0.37	0.21	0.02	0.04	0.04	0.04	0.03
M-n151-k12	0.18	0.23	0.04	0.02	0.03	0.06	0.02
M-n200-k17	0.14	0.27	0.03	0.02	0.02	0.03	0.01
P-n50-k10	0.10	0.28	0.03	0.03	0.03	0.06	0.01
P-n55-k15	0.17	0.23	0.05	0.04	0.03	0.06	0.01
P-n70-k10	0.08	0.19	0.01	0.01	0.02	0.02	0.01
P-n76-k5	0.11	0.20	0.03	0.02	0.03	0.07	0.02
Average	0.16	0.26	0.05	0.06	0.04	0.07	0.03

may affect the outcome. It is more likely to obtain smaller IGD when the size of the approximated Pareto frontier matches the reference set size (Bezerra et al., 2017). In the experiments, reference sets are composed of all the non-dominated solutions obtained by all the algorithms. Therefore, the size of the reference sets could be large. That could justify why MDLS with larger set sizes obtains the second best performance for IGD. Regarding the hypervolume and epsilon indicators, although all the methods produce nearly similar results, MDLS and weighting methods yield slightly better results, respectively. As mentioned earlier, hypervolume can be affected by the number of non-dominated solutions. If the approximated Pareto front values of two given methods are close enough, the one with more non-dominated solutions obtains a greater volume and therefore yields a better hypervolume. Among all the indicators, the epsilon indicator is more sensitive to the quality of two extreme points of an approximated Pareto front. When the endpoint is closer to the ideal point, the outcome of this indicator is better.

Figure 2 shows the approximated Pareto frontier found by EMDLS for instance E-n51-k5. In this instance, the customer nodes are randomly scattered within a square of side 75 and the depot is located in the centre of the square. The approximated Pareto frontier contains 30 non-dominated solutions ranging from the extreme cost solution to the extreme environmental impact solution. While the extreme cost solution (A) uses 5 vehicles of type L (ICEs) to meet all customer demands, the extreme environmental impact solution (C) utilises 8 vehicles of type S (short-range EVs). The results show that the neighbouring non-dominated solutions use the same fleet configuration. However, there is only one non-dominated solution which uses the vehicles of type M (PHEVs) within its fleet configuration. Although PHEVs produce relatively lower CO_2 emissions than ICEs, EVs and ICEs are preferred in the fleet configuration of almost all the non-dominated solutions.

Table 7 investigates in more detail the usage of greener vehicles within a set of non-dominated solutions provided by EMDLS for each instance. Columns 2-3 provide information on the location of the depot and the way customer nodes are scattered. Columns 4-6 display the percentage utilisation of each type of vehicle through the entire number of non-dominated solutions of each instance. Column 7 shows the percentage difference between the best and worst cost solutions. It should be noted that the worst cost solution provides the greener fleet with the highest number of EVs used. Column 8 represents the same information concerning the environmental objective. The average percentage usage of the vehicle type S is 40% through all sets of non-dominated solutions. The results show that the location of the depot can have an impact on the usage of EVs. The average percentage of vehicle type S used in the instances, where the depot is located in the centre, is 0.46%; while the value is 30% for the other instances. Regarding the data pattern, the results do not show any significant differences among different types of geographical patterns of the customer nodes.

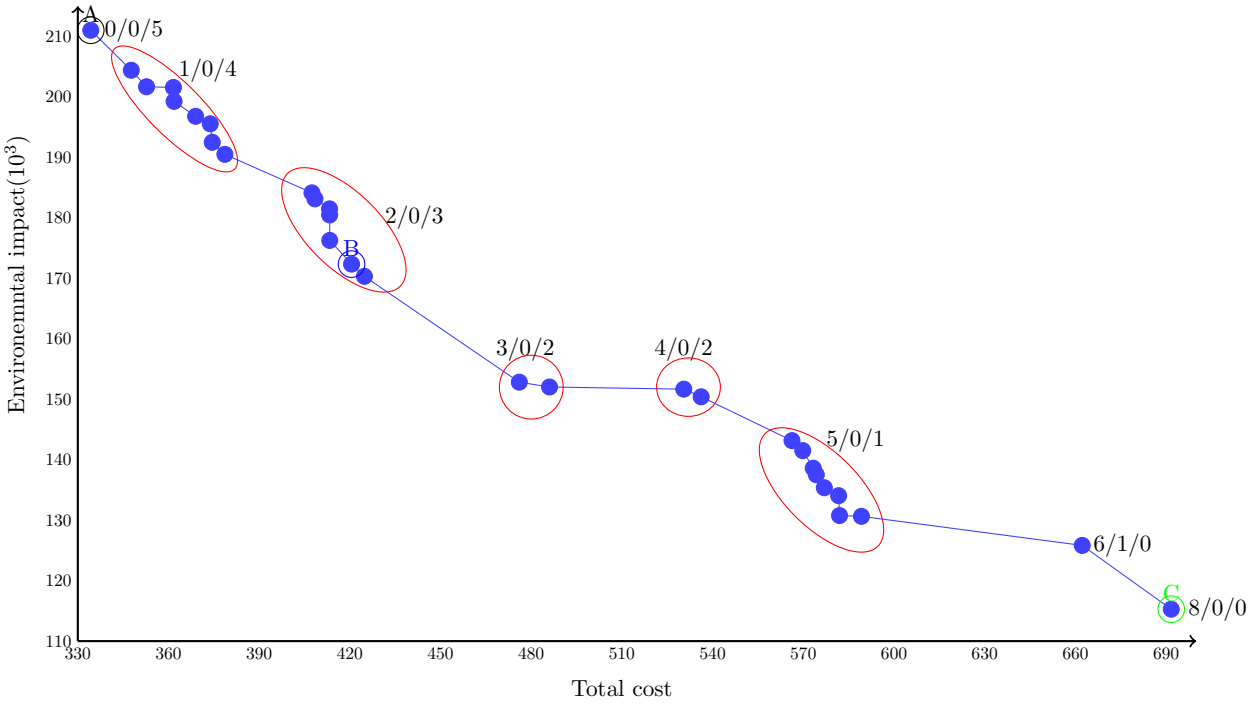

Figure 2: EMDLS approximated pareto front for instance E-n51-k5

For instances where the data is randomly generated, the average percentage usage of vehicle type S is 41% against 38% for the other patterns.

The average difference between the minimum and maximum values of cost and environmental impact over all instances are 92% and 67%, respectively. This means that deploying a fleet configuration of green vehicles can be considerably expensive. Another issue is that vehicle type M has been rarely deployed, although it has some competitive specifications such as the longest driving range. This proves that decreasing cost is still as important as increasing the driving range. Fortunately, there are ongoing advancements in the technology of electric vehicles and more precisely their battery charges. For example, Tesla recently introduced the longest-range consumer electric vehicle in the world with a range of 335 miles on a full charge (Snyder, 2017). Therefore, having electric vehicles with longer driving ranges could lead to a greener and meanwhile less costly fleet configurations.

Table 7: Fleet configuration analysis in terms of the greener vehicles usage and environmental impact

Instance name	Depot's location	Data pattern	% usage of			Gap %	
			S	M	L	Cost	Environmental
A-n32-k5	Corner	Randomly	0.15	0	0.85	76.47	61.88
A-n38-k5	Not Center	Randomly	0.26	0.03	0.71	80.24	53.59
A-n65-k9	Not Center	Randomly	0.31	0.03	0.66	81.41	58.30
A-n80-k10	Corner	Randomly	0.14	0.03	0.84	87.68	52.78
B-n50-k7	Center	Cluster	0.25	0.05	0.70	83.78	56.13
B-n52-k7	Not Center	Cluster	0.28	0.05	0.67	81.47	58.07
B-n57-k9	Corner	Cluster	0.44	0.06	0.50	96.09	46.44
B-n78-k10	Not Center	Cluster	0.23	0.05	0.72	86.26	53.64
E-n22-k4	Center	Randomly	0.35	0.05	0.60	63.03	76.36
E-n30-k3	Center	Not clear	0.40	0	0.60	74.43	64.12
E-n51-k5	Center	Randomly	0.50	0.04	0.45	108.13	82.30
E-n76-k10	Center	Randomly	0.48	0.05	0.46	101.41	87.96
E-n76-k14	Center	Randomly	0.51	0.04	0.45	110.56	79.96
F-n135-k7	Not center	Real case	0.34	0.13	0.53	78.82	60.05
Golden-17	Center	Star shape	0.56	0.05	0.39	109.93	80.55
Golden-19	Center	Star shape	0.49	0.05	0.46	115.72	75.70
Golden-20	Center	Star shape	0.53	0.05	0.42	114.56	76.63
M-n101-k10	Center	Cluster	0.36	0.05	0.58	68.98	68.94
M-n121-k7	Corner	Cluster	0.55	0.05	0.40	72.61	65.81
M-n151-k12	Center	Randomly	0.42	0.09	0.49	85.32	54.32
M-n200-k17	Center	Randomly	0.46	0.06	0.49	81.12	57.95
P-n50-k10	Center	Randomly	0.52	0.04	0.44	103.16	81.73
P-n55-k15	Center	Randomly	0.53	0.07	0.40	119.40	72.84
P-n70-k10	Center	Randomly	0.53	0.09	0.38	102.45	87.46
P-n76-k5	Center	Randomly	0.53	0.15	0.32	125.82	64.08
average			0.40	0.06	0.54	92.35	67.10

6. Conclusion and future work

This paper proposes a bi-objective optimisation model and a new EMDLS method to minimise the total monetary cost (including both fixed and variable costs) along with CO_2 emissions, considering hybrid fleets of traditional and electric vehicles with limited driving range. The primary goal of this study is to provide decision-makers with a set of different fleet alternatives, ranging from those with low cost to the most environment-friendly ones. To the best of our knowledge, this is the first time that the Pareto frontier has been explicitly approximated.

The proposed EMDLS method aims at finding a set of mutually non-dominated solutions, and presents the following novel concepts: *(i)* devising an approach based on the variable neighbourhood search metaheuristic, which allows to determine the number of solutions to be explored; *(ii)* a selection mechanism of less crowded solutions to explore with the aim of obtaining a better approximation of the Pareto frontier; and *(iii)* the use of an adaptive weighted method to find non-supported efficient solutions. A comparative analysis has been conducted to compare the performance of the

EMDLS with MDLS, IMDLS, NSGAII, NSGAIII and the weighting and the epsilon-constraint methods. The computational results show that EMDLS provides competitive results in terms of several indicators. For instance, it outperforms the other methods with regard to the ratio and IGD indicators. Furthermore, the weighting and epsilon-constraint methods yield relatively the worse performance compared to the other methods inspired by MDLS.

The numerical results also show that there is a significant difference between the least and most environment-friendly fleets in terms of distance-based cost and environmental cost. On one hand, the average percentage difference between the minimum and maximum values of cost over all instances is 92%. On the other hand, the average difference between the maximum and minimum values of CO_2 emissions is 67%. However, the results for the hybrid vehicles show the viability of employing electric vehicles within the fleet configurations subject to the data used in the experiments. Interestingly enough, the most environment-friendly fleets utilise only short-range electric vehicles in some instances, which proves that it is viable to use electric vehicles in the fleet configuration. In those instances, the average difference between the minimum and maximum values of distance-based cost is 100%. As a promising fact, the technology of electric vehicles and more precisely their battery charges is noticeably progressing, which may lead to greener and meanwhile less costly fleet configurations.

One of the main future research directions of this work is to study the stochastic variants of the heterogeneous vehicle routing problem with multiple driving-range limitations. Environmental data could be a source of uncertainty because of the lack of historical data and inherited uncertainties of measuring environmental costs. Due to problem complexity and lack of benchmark solutions, an exact method could be used to obtain optimal solutions or lower bounds in small-scale scenarios. Finally, applying a more comprehensive method to measure the environmental impact through the entire life cycle of a vehicle could also be a promising research area.

Acknowledgments

This work has been partially supported by the University of Portsmouth and the Spanish Ministry of Economy and Competitiveness (TRA2015-71883-REDT).

References

Bektaş, T., Demir, E., and Laporte, G. (2016). *Green Transportation Logistics*, chapter Green vehicle routing, pages 243–265. Springer International Publishing.

Bektaş, T. and Laporte, G. (2011). The Pollution-Routing Problem Tolga Bektas. *Transportation Research Part B: Methodological*, 45(8):1232–1250.

Bezerra, L. C. T., López-Ibáñez, M., and Stützle, T. (2017). An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. In Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., and Grimme, C., editors, *Evolutionary Multi-Criterion Optimization*, pages 31–45. Springer International Publishing.

Browne, M., Allen, J., and Leonardi, J. (2011). Evaluating the use of an urban consolidation centre and electric vehicles in central London. *IATSS Research*, 35(1):1–6.

Bruglieri, M., Mancini, S., Pezzella, F., and Pisacane, O. (2016). A new mathematical programming model for the green vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 55:89 – 92.

Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., and Suraci, S. (2017). A three-phase matheuristic for the time-effective electric vehicle routing problem with partial recharges. *Electronic Notes in Discrete Mathematics*, 58:95 – 102.

Caballero, R., González, M., Guerrero, F., Molina, J., and Paralera, C. (2007). Solving a multiobjective location routing problem with a metaheuristic based on tabu search. application to a real case in Andalusia. *European Journal of Operational Research*, 177(3):1751 – 1763.

Clark, P. and Campbell, P. (2016). Motor industry: Pressure on the pump. `https://www.ft.com/content/31d68af8-6e0a-11e6-9ac1-1055824ca907`. Accessed: 2018-10-16.

Clarke, G. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581.

Council, N. R. (2015). Overcoming barriers to deployment of plug-in electric vehicles. Technical report, National Academies Press.

Deb, K. and Jain, H. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Dekker, R., Bloemhof, J., and Mallidis, I. (2012). Operations Research for green logistics - An overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3):671–679.

Demir, E., Bektaş, T., and Laporte, G. (2014). The bi-objective Pollution-Routing Problem. *European Journal of Operational Research*, 232(3):464–478.

Eberhart, R. C., Simpson, P., Dobbins, R., and Dobbins, R. W. (1996). *Computational intelligence PC tools*. Boston: Academic Press.

Edie.net. Ups to trial electric vehicle 'smart grid' in london. `https://www.edie.net/news/8/UPS-to-trial-electric-vehicle--smart-grid--in-London/`. Accessed: 2018-10-16.

Elhedhli, S. and Merrick, R. (2012). Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, 17(5):370–379.

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

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

Eskandarpour, M., Dejax, P., and Péton, O. (2017). A large neighborhood search heuristic for supply chain network design. *Computers and Operations Research*, 80:23–37.

Felipe, A., Ortuño, M. T., Righini, G., and Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111 – 128.

Financial mentor, m. (2018). Car cost calculator. `https://financialmentor.com/calculator/car-cost-calculator`. Accessed: 2018-04-10.

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

Fueleconomy.gov. The official u.s government source for fuel economy information. `https://www.fueleconomy.gov/feg/`. Accessed: 2018-04-10.

Glover, F., Laguna, M., and Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 29(3):652–684.

Goeke, D. and Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99.

Grasas, A., Juan, A. A., Faulin, J., de Armas, J., and Ramalhinho, H. (2017). Biased randomization of heuristics using skewed probability distributions: A survey and some applications. *Computers & Industrial Engineering*, 110:216 – 228.

Hiermann, G., Puchinger, J., and Hartl, R. (2014). The Electric Vehicle Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, 48(4):500–520.

Juan, A., Faulin, J., Jorba, J., Riera, D., Masip, D., and Barrios, B. (2011). On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics. *Journal of the Operational Research Society*, 62(6):1085–1097.

Juan, A. A., Faulin, J., Cruz, J. C., Barrios, B. B., and Martinez, E. (2014a). A successive approximations method for the heterogeneous vehicle routing problem: analysing different fleet configurations. *European Journal of Industrial Engineering*, 8(6):762.

Juan, A. A., Faulin, J., Ferrer, A., Lourenço, H. R., and Barrios, B. (2013). Mirha: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *Top*, 21(1):109–132.

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

Juan, A. A., Mendez, C. A., Faulin, J., De Armas, J., and Grasman, S. E. (2016). Electric vehicles in logistics and transportation: A survey on emerging environmental, strategic, and operational challenges. *Energies*, 9(2):1–21.

Kara, I., Kara, B. Y., and Yetis, M. K. (2007). *Energy Minimizing Vehicle Routing Problem*, pages 62–71. Springer Berlin Heidelberg, Berlin, Heidelberg.

Koç, C., Bektaş, T., Jabali, O., and Laporte, G. (2015). A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers and Operations Research*, 64:11 – 27.

Koç, C., Bektaş, T., Jabali, O., and Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1 – 21.

Koç, C. and Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, 39:154 – 164.

Lian, K., Milburn, A. B., and Rardin, R. L. (2016). An improved multi-directional local search algorithm for the multi-objective consistent vehicle routing problem. *IIE Transactions*, 8830(August):1–18.

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., and Lam, H. Y. (2014). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*, 41(4 PART 1):1118–1138.

Lwin, K., Qu, R., and Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24:757 – 772.

Madankumar, S. and Rajendran, C. (2018). Mathematical models for green vehicle routing problems with pickup and delivery: A case of semiconductor supply chain. *Computers & Operations Research*, 89:183 – 192.

Matei, O., Pop, P. C., Laszlo Sas, J., and Chira, C. (2015). An improved immigration memetic algorithm for solving the heterogeneous fixed fleet vehicle routing problem. *Neurocomputing*, 150:58 – 66.

Montoya, A., Gu, C., Villegas, J. G., and Mendoza, J. E. (2017). The electric vehicle routing problem with partial charging and nonlinear charging function. *Transportation Research Part B*, 0:1–10.

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

Rakshit, P. and Konar, A. (2015). Differential evolution for noisy multiobjective optimization. *Artificial Intelligence*, 227:165 – 189.

Snyder, J. B. (2017). Tesla model s 100d goes 335 miles per charge. `https://www.autoblog.com/2017/01/20/tesla-model-s-100d-range-price/?guccounter=1`. Accessed: 2018-10-16.

Tight, M. R., Bristow, A. L., Pridmore, A., and May, A. D. (2005). What is a sustainable level of CO2 emissions from transport activity in the UK in 2050? *Transport Policy*, 12(3):235–244.

Tricoire, F. (2012). Multi-directional local search. *Computers and Operations Research*, 39(12):3089–3101.

Union of Concerned scientists, U. (2018). How clean is your electric vehicle? `https://www.ucsusa.org/clean-vehicles/electric-vehicles/ev-emissions-tool#.Wu\-CEeNRuaM9`. Accessed: 2018-04-10.

Vaz Penna, P. H., Afsar, H. M., Prins, C., and Prodhon, C. (2016). A Hybrid Iterative Local Search Algorithm for The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations. *IFAC-PapersOnLine*, 49(12):955–960.

Xiao, Y., Zhao, Q., Kaku, I., and Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers and Operations Research*, 39(7):1419–1431.

Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

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

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary computation*, 8(2):173–195.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.