

# Vehicle Routing Problem with Mixed Fleet of Conventional and Heterogenous Electric Vehicles and Time Dependent Charging Costs

Ons Sassi, Wahiba Ramdane Cherif-Khettaf, Ammar Oulamara

**Abstract**—In this paper, we consider the vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs, denoted VRP-HFCC, in which a set of geographically scattered customers have to be served by a mixed fleet of vehicles composed of a heterogenous fleet of Electric Vehicles (EVs), having different battery capacities and operating costs, and Conventional Vehicles (CVs). We include the possibility of charging EVs in the available charging stations during the routes in order to serve all customers. Each charging station offers charging service with a known technology of chargers and time dependent charging costs. Charging stations are also subject to operating time windows constraints. EVs are not necessarily compatible with all available charging technologies and a partial charging is allowed. Intermittent charging at the depot is also allowed provided that constraints related to the electricity grid are satisfied. The objective is to minimize the number of employed vehicles and then minimize the total travel and charging costs. In this study, we present a Mixed Integer Programming Model and develop a Charging Routing Heuristic and a Local Search Heuristic based on the Inject-Eject routine with different insertion methods. All heuristics are tested on real data instances.

**Keywords**—charging problem, electric vehicle, heuristics, local search, optimization, routing problem.

## I. INTRODUCTION

**T**HE substantial growth of the transport sector in recent years has made it the prime player in energy consumption and greenhouse gas emissions. Providing better planning of urban transportation services becomes certainly challenging due to the crowded traffic infrastructure, increasing customer expectations and rules set by municipalities. Nowadays, the governments are more and more aware of the urgency to tackle transport problems and conserve the environment. Moreover, there has been a significant body of research on making urban transportation more efficient and sustainable. Investing in more environmentally friendly and safe modes of transportation such as the as the ridesharing service [1] and Electric Vehicles (EVs) use [2] is becoming a necessity today. In fact, the EV represents nowadays a credible alternative to the more conventional engines. Convinced that this green vehicle is one of the responses to the worldwide environmental and energy issues, governments and business organizations pay today particular attention to quickly install EVs in every city and to optimize their use. However, EVs are currently facing several weaknesses related to the limited driving range, the long charging

time, the availability of a charging infrastructure and the high purchasing costs. Thus, the deployment of a large scale of EVs needs, mainly in the business context, a prior reorganization of the vehicles' routes in order to meet the EV limits related to the battery range and to satisfy the customers.

This work is an extension of the real-world problem that was addressed in the framework of the French national R&D project Infini Drive, led by La Poste Group, ERDF (French Public Electricity Distribution Network Manager) and seven other companies and research laboratories. This project has been funded by ADEME (French Environment and Energy Control Agency) as part of the 'Vehicle of the Future' program. It aims at designing, with a progressive approach, a system for managing charging infrastructures that allows for economically viable and ecologically sustainable deployment of EVs fleets of companies and public authorities.

Furthermore, this study follows on from the work presented in [3] where exact and heuristic methods were presented to solve the joint EV scheduling and charging problem. This studied problem consists in assigning EVs and CVs to already constructed routes and optimizing EVs charging. Within this study, we extend this problem to the case where the routes need to be constructed and assigned to the available vehicles with the objective of minimizing the overall routing and charging costs.

In this paper, we consider the vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles, in which a set of geographically scattered customers have to be served by a fleet of CVs and EVs operating with plug-in batteries. EVs need to be charged in charging stations during the trips in order to serve all customers.

More precisely, our problem can be defined as follows: Given a set of customers, a set of charging stations having different types of chargers, proposing different time dependant charging costs and subject to operating time windows constraints, and a number of heterogenous EVs and CVs. We seek to minimize the number of used vehicles while fostering the use of EVs, as well as minimizing both transportation and charging costs for visiting customers, while every customer is visited exactly once and routes start and end at the depot. During the trips, EVs could be charged, either totally or partially, at any of the available charging stations while satisfying temporal, battery and load capacity constraints.

In order to minimize their investment costs of charging infrastructure, companies may accept to share their charging

O. Sassi, W. Ramdane Cherif-Khettaf, A. Oulamara are with the Department of Computer Engineering, Loria, Nancy, France. E-mail:ons.sassi@loria.fr

infrastructures with other EVs users. However, they impose that charging should only be undertaken during limited time intervals and propose time dependant charging costs that may allow for the smoothing of the energy consumption curve and the avoiding of energy consumption peaks. Thus, we consider that the charging stations propose time-dependant charging costs, have predefined opening time windows and allow charging using different charging technologies. Within this study, we consider that the charging stations could propose three different charging technologies: (i) Level 1 charger which is the slowest charging level that provides charging with a power of 3.7 kW; (ii) Level 2 charger offers charging with a power of 22 kW and (iii) Level 3 charger which is the fastest charging level that delivers a power of 53 kW.

Charging at the depot is also allowed and it could be intermittent provided that charging constraints related to the electricity grid are satisfied. In fact, at each time period  $t$ , the total grid power available to charge EVs is limited and the electricity cost may vary. Different Level 1 chargers are available at the depot and could be used during the night to charge EVs. Level 2 chargers are also available and could be used during a limited time interval.

To the best of our knowledge, no previous study was devoted to tackle this problem in the literature.

To solve the VRP-HFCC, we develop a Constructive Heuristic and a Local Search Heuristic based on the Inject-Eject routine with three different insertion strategies. All heuristics are tested on real data instances.

The remainder of this paper is organized as follows. In Section II, a review of related literature is presented. In Section III, we introduce the notation in detail. In Section IV, we provide a mixed-integer linear programming formulation of VRP-HFCC. Section V describes the solving approaches. Experimental results on real data instances are presented in Section VI. Section VII gives a short summary and conclusion of the paper.

## II. RELATED WORK

In this section, we review the literature related to the electric vehicle routing problems and the controlled EV charging problems.

EVs charging problems as well as EVs routing problems have attracted close attention from researchers and business organizations in recent years. Thus, the number of publications focusing on the EV has significantly increased in the last few years. A recent overview of many issues related to the use of EVs for goods distribution can be found in [4].

The controlled charging problem consists in a better management of EVs charging in order to minimize the charging cost. In [5], the authors design a simulation environment, which produces charging schedules using a multi-objective evolutionary optimization algorithm. [6] exposes an energy consumption scheduler able to reduce peak power load in smart places based on genetic algorithms. A concept of real-time scheduling techniques for EV charging to minimize the impact on the power grid and to guarantee the satisfaction of consumers charging requirements is suggested in [7]. In [3], the problem of jointly EVs scheduling and

charging is addressed. This problem consists in simultaneously assigning EVs and CVs to already constructed routes and EVs charging optimizing. The NP-Hardness of this problem is proven and exact and heuristic methods are proposed and tested on real data instances.

In the case where EVs routes are not already constructed, we refer to the Electric Vehicle Routing Problem which is an extension of the more general Vehicle Routing Problem (see for example [8] and [9]).

The problem of energy-optimal routing is addressed in [10]. In [11], the authors formulate the Green Vehicle Routing Problem (GVRP) as a Mixed Integer Linear Program (MIP). Two constructive heuristics are developed to solve this problem. An overview of the GVRP is given in [12].

Schneider et al. [13] combine a Vehicle Routing Problem with the possibility of refueling a vehicle at a station along the route. They introduce the Electric Vehicle Routing Problem with Time Windows and Recharging Stations (E-VRPTW), which incorporates the possibility of recharging at any of the available stations using an appropriate recharging scheme. E-VRPTW aims at minimizing the number of employed vehicles and total traveled distance.

We are also aware of more recent studies that were conducted simultaneously with our work. In [14], the Electric Vehicle Routing Problem with Time Windows and Mixed Fleet (E-VRPTWMF) to optimize the routing of a mixed fleet of EVs and CVs is addressed. On each visit to a recharging station, EVs are recharged to their maximum battery capacity with a constant recharging rate. To solve this problem, an Adaptive Large Neighborhood Search algorithm that is enhanced by a local search for intensification is proposed.

Almost the same problem is addressed in [15]. The only difference here is the fact of considering heterogeneous vehicles that differ in their transport capacity, battery size and acquisition cost. An Adaptive Large Neighbourhood Search with an embedded local search and labelling procedure for intensification is also used to solve the problem. In [16], the authors present a variation of the electric vehicle routing problem in which different charging technologies are considered and partial EV charging is allowed. This problem is the closest to our problem in the sense that we consider different charging technologies and partial EV charging. However, several major differences have to be outlined. Firstly, we consider a mixed fleet composed of heterogeneous EVs and CVs. Secondly, the costs of charging at the depot and at the charging stations are assumed to be time dependent. Moreover, the charging stations are subject to operating time windows constraints and charging at the depot is subject to the grid's maximum capacity constraints. Besides, EVs are not necessarily compatible with all charging technologies.

In short, we differ from all the above-mentioned studies in that we consider a heterogeneous fleet composed of EVs and CVs, different types of charging stations and different time-dependent charging costs. Moreover, EV charging at the depot could be intermittent and is subject to real-life constraints such as the maximum grid capacity constraint. We also consider that not all EV are compatible with fast charging technologies and that partial charging is allowed. Our objective

function is also different. In fact, we aim at minimizing total operating and charging costs involved with the use of a mixed fleet. Our overall objective is to provide enhanced optimization methods for EV charging and routing that are relevant to the described constraints.

### III. PROBLEM DESCRIPTION AND NOTATION

We define the VRP-HFCC on a complete, directed graph  $G = (V', A)$ .  $V'$  denotes the set of vertices composed of the set  $V$  of  $n$  customers, the set  $F'$  of dummy vertices generated to permit several visits to each vertex in the set  $F$  of charging stations  $F = \{1, \dots, f\}$  and  $D'$  the set of dummy vertices generated to allow several visits to each charger at the depot  $D = \{1, \dots, |D|\}$ . The set of arcs is denoted by  $A = \{(i, j) \mid i, j \in V', i \neq j\}$ . The depot is denoted by either 0 or  $n+1$  depending if it is the initial or terminal node of a route. We denote by  $V'_0$  the set  $V' \cup \{0\}$ , by  $V'_{n+1}$  the set  $V' \cup \{n+1\}$  and by  $V'_{0,n+1}$  the set  $V' \cup \{0, n+1\}$ . The optimization time horizon  $[0, T]$ , which represents typically a day, is divided into  $T$  equidistant time periods,  $t = 1, \dots, T$ , each of length  $\delta$ , where  $t$  represents the time interval  $[t-1, t]$ . We define the night interval  $[0, T_0] \subset [0, T]$  during which charging at the depot with Level 1 chargers could be performed. Moreover, no customer has to be served during the night period. We define the service interval  $[T_0, T] \subset [0, T]$  during which all customers have to be served and the EVs could be charged in the different charging stations as well as in the depot using the available chargers. A nonnegative demand  $q_i$  is associated with each customer  $i \in V$ , this represents the quantity of goods that will be delivered to this customer. With each customer we also associate a service time  $s_i$ . Each arc  $(i, j) \in A$  is defined by a distance  $d_{i,j}$  and a nonnegative travel time  $t_{i,j}$  required to travel  $d_{i,j}$ . When an arc  $(i, j)$  is traveled by an EV, it consumes an amount of energy  $e_{i,j}$  equal to  $r \times d_{i,j}$ , where  $r$  denotes a constant energy consumption rate.

Each charging station  $f \in F$  can deliver a maximum charging power  $p_f$  (kW) and proposes a time dependent charging cost  $c_{f,t}, \forall t = T_0, \dots, T$ ; which represents the charging cost during the time period  $t$ , expressed in (euros/kWh). The chargers in charging station  $f$  are available during the time window  $[a_f, b_f]$ . Accordingly, the EV must wait if it arrives at charging station  $f$  before time  $a_f$ .

We consider a set  $M_{EV} = \{1, \dots, m_{EV}\}$  of EVs and a set  $M_{CV} = \{m_{EV} + 1, \dots, m_{EV} + m_{CV}\}$  of Combustion Engine Vehicles (CVs), needed to serve all customers. Each EV  $k$  operates with a battery characterized by its nominal capacity of embedded energy  $CE_k$  (kWh) and its State of Charge ( $SoC_k^0$ ) at time  $t = 0$  expressed as a ratio of the available amount of energy and  $CE_k$  ( $0 = \text{empty}$ ;  $1 = \text{full}$ ). At low and high SoC's values, the battery tends to degrade faster ([17] and [18]). In order to improve its lifetime after repeated use and to respect the security issues, at each time  $t$ ,  $SoC_k^t$  should be in the interval  $[SoC_k^{\text{Min}}, SoC_k^{\text{Max}}]$ , where  $SoC_k^{\text{Min}}$  and  $SoC_k^{\text{Max}}$  are the minimal and maximal allowable values of

SoC, respectively.

Each EV (CV) is characterized by a maximum capacity  $Q^{\text{EV}}$  ( $Q^{\text{CV}}$ ) ( $\text{m}^3$ ) which represents the maximum quantity of goods that could be transported by the vehicle. Denote by  $FC^{\text{EV}}$  ( $FC^{\text{CV}}$ ) (euros/ day) the fixed costs related to EVs (EVs). Denote by  $OC_k^{\text{EV}}$  ( $OC^{\text{CV}}$ ) the operating costs (euros/kilometer) related to the maintenance of the EV (CV), accidents, etc. Thus, if an arc  $(i, j)$  is traveled by an EV  $k$  (CV  $k$ ), this has an operating cost denoted by  $cost_{i,j,k}^{\text{EV}}$  ( $cost_{i,j,k}^{\text{CV}}$ ) and is computed as:  $cost_{i,j,k}^{\text{EV}} = d_{i,j} \times OC_k^{\text{EV}}$  ( $cost_{i,j,k}^{\text{CV}} = d_{i,j} \times OC^{\text{CV}}$ ).

At the depot, a given number of slow chargers are available to charge the EVs during the optimization horizon  $[0, T]$  and a predefined number of fast chargers are available to charge the vehicles only during the service time  $[T_0, T]$ .

At each time period  $t$ , each charger at the depot can apply on EV  $k$  a charging power  $p_{kt} \in [p^{\text{Min}}, p^{\text{Max}}]$  where  $p^{\text{Min}}$  and  $p^{\text{Max}}$  are the minimal and maximal powers that can be delivered by the charger, respectively. Thus, an EV charged with a power  $p_{kt}$  during the time period  $t$  retrieves a total amount of energy equal to  $\delta \times p_{kt}$  (kWh). We denote by  $GP_t$  the electricity grid capacity available for EV charging at time  $t$ ; i.e., at each time period  $t$ , the total grid power available to charge all EVs is limited to  $GP_t$ . Let  $c'_t$  be the energy cost during  $t$ .

Each customer  $i \in V$  should be visited, by either an electric or conventional vehicle, exactly once during  $[T_0, T]$ . Each charging station could be visited as many times as required or not at all. When charging is undertaken in a charging station  $f$ , it is assumed that only the required quantity of energy is injected into the EV battery. Thus, EVs could be partially charged.

Since we consider, within this study, many charging technologies (slow and fast charging), we should also consider the fact that not all EVs technologies are compatible with fast charging. Thus, when we plan the charging of an EV, only the charging stations proposing compatible charging technologies should be considered. A feasible solution to our problem is composed of a set of feasible routes assigned to adequate vehicles and a feasible EVs charging planning.

A feasible route is a sequence of nodes that satisfies the following set of constraints:

- Each route must start and end at the depot;
- the overall amount of goods delivered along the route, given by the sum of the demands  $q_i$  for each visited customer, must not exceed the vehicle capacity ( $Q^{\text{EV}}$  or  $Q^{\text{CV}}$ );
- the total duration of each route, calculated as the sum of all travel durations required to visit a set of customers, the time required to charge the vehicle during the interval  $[T_0, T]$ , the service time of each customer and, eventually, the waiting time of the EV if it arrives at a charging station before its opening time, could not exceed  $T - T_0$ ;
- no more than  $m_{EV}$  EVs and  $m_{CV}$  CVs are used;
- each customer should be visited between  $T_0$  and  $T$ ;
- the following charging constraints are satisfied:
  - The charging level of the battery of each EV  $k$  must

always be in the interval  $[SoC_k^{\text{Min}}, SoC_k^{\text{Max}}]$ ;

- when charging is undertaken, the EV should be charged with a compatible charging technology;
- when the EVs are charged at the depot, the total power used to charge them does not exceed the grid's maximum capacity and the minimum and the maximum powers of chargers should be respected;
- during  $[0, T_0]$ , EV charging at the depot could only be performed using the available Level 1 chargers;
- at each charging station  $f$ , charging could only be undertaken during its operating time window  $[a_f, b_f]$ ;

We seek to construct a minimum number of routes such that all customers are served, all EVs are optimally charged and the total cost of routing and charging is minimized.

#### IV. PROBLEM FORMULATION

In this section, we propose a Mixed Integer Programming Model (MIP) for the VRP-HFCC. Let  $F'_k$  (respectively,  $D'_k$  and  $D_k$ ) be the set of external charging stations in  $F'$  (respectively, chargers in  $D'$  and chargers in  $D$ ) that are compatible with the vehicle  $k$ . We introduce the following decision variables:

$x_{ijk}^{\text{EV}}$ : is a 0-1 variable equal to 1 if an EV  $k$ ;  $k \in M_{\text{EV}}$ ; travels from  $i \in V'_0$  to  $j \in V'_{n+1}$  and 0 otherwise.

$x_{ijk}^{\text{CV}}$ : is a 0-1 variable equal to 1 if a CV  $k$ ;  $k \in M_{\text{CV}}$ ; travels from  $i \in V \cup \{0\}$  to  $j \in V \cup \{n+1\}$  and 0 otherwise.

$y_{kt}$ : is a 0-1 variable equal to 1 if the vehicle  $k$ ;  $k \in M_{\text{EV}}$ ; is charged during the time interval  $t \in [0, T_0]$  and 0 otherwise.

$y'_{tjk}$ : 0-1 variable specifying if an EV  $k$  is charged during the time period  $t \in [T_0, T]$  in charging station  $f \in F'_k \cup D'_k$ .

$p_{ktg}$ : decision real variable denoting the charging power level applied to EV  $k$ ;  $k = 1, \dots, m_{\text{EV}}$ ; using a charger  $g$  at the depot at time interval  $t \in [0, T_0]$  (kW).

$p'_{tjk}$ : real variable representing the charging rate of EV  $k$  at time period  $t \in [T_0, T]$  in charging station  $f \in F'_k \cup D'_k$  (kW).

$tt_{jk}$ : decision real variable specifying the arrival time of a vehicle  $k$ ;  $k \in M_{\text{EV}} \cup M_{\text{CV}}$ ; at vertex  $j \in V'$ .

$E_{jk}^a$ : real variable specifying the amount of energy available in the battery of vehicle  $k$ ;  $k \in M_{\text{EV}}$ ; when arriving at node  $j$  (kWh).

$E_{jk}^l$ : amount of energy available in the battery of EV  $k$ ;  $k \in M_{\text{EV}}$ ; when leaving node  $j$  (kWh).

$l_{jk}$ : real variable specifying the amount of load left in the vehicle  $k$ ;  $k \in M_{\text{EV}} \cup M_{\text{CV}}$ ; after visiting node  $j$  ( $\text{m}^3$ ).

$W_{jk}$ : the waiting time of EV  $k$ ;  $k \in M_{\text{EV}}$ ; when arriving at charging station  $j \in F'_k \cup D'_k$ .

In the following, we detail the mathematical formulation ( $\mathcal{P}$ ) of VRP-HFCC:

$$\begin{aligned} \text{Min } & \sum_{k \in M_{\text{EV}}} \sum_{i \in V'_0, j \in V'_{n+1}} \text{cost}_{i,j,k}^{\text{EV}} \times x_{ijk}^{\text{EV}} + \\ & \sum_{k \in M_{\text{CV}}} \sum_{i \in V \cup \{0\}, j \in V \cup \{n+1\}} \text{cost}_{i,j,k}^{\text{CV}} \times x_{ijk}^{\text{CV}} + \\ & \sum_{k \in M_{\text{EV}}} \sum_{f \in F'_k \cup D'_k} \sum_{t \in [T_0, T]} c_{f,t} \times p'_{tjk} + \\ & \sum_{k \in M_{\text{EV}}} \sum_{j \in D} \sum_{t \in [0, T_0]} c'_t \times p_{ktj} + \sum_{k \in M_{\text{EV}}} \sum_{j \in V'} FC^{\text{EV}} \times x_{0jk}^{\text{EV}} + \\ & \sum_{k \in M_{\text{CV}}} \sum_{j \in V} FC^{\text{CV}} \times x_{0jk}^{\text{CV}} + \sum_{k \in M_{\text{EV}}} \sum_{f \in F'_k \cup D'_k} \beta \times W_{fk} \quad (1) \end{aligned}$$

$$\sum_{k \in M_{\text{EV}}} \sum_{j \in V'; i \neq j} x_{ijk}^{\text{EV}} + \sum_{k \in M_{\text{CV}}} \sum_{j \in V'; i \neq j} x_{ijk}^{\text{CV}} = 1, \forall i \in V \quad (2)$$

$$\sum_{k \in M_{\text{EV}}} \sum_{j \in V'_{n+1}; i \neq j} x_{ijk}^{\text{EV}} \leq 1, \forall i \in F' \cup D' \quad (3)$$

$$\sum_{k \in M_{\text{EV}}} \sum_{j \in V'_{n+1}; i \neq j} x_{ijk}^{\text{EV}} - x_{jik}^{\text{EV}} = 0, \forall i \in V' \quad (4)$$

$$\sum_{k \in M_{\text{CV}}} \sum_{j \in V \cup \{n+1\}; i \neq j} x_{ijk}^{\text{CV}} - x_{jik}^{\text{CV}} = 0, \forall i \in V \quad (5)$$

$$\sum_{j \in V'; i \neq j} x_{0jk}^{\text{EV}} \leq 1, \forall k \in M_{\text{EV}} \quad (6)$$

$$\sum_{j \in V; i \neq j} x_{0jk}^{\text{CV}} \leq 1, \forall k \in M_{\text{CV}} \quad (7)$$

$$\sum_{j \in V; i \neq j} \sum_{k \in M_{\text{EV}}} x_{0jk}^{\text{EV}} \leq m_{\text{EV}} \quad (8)$$

$$\sum_{j \in V; i \neq j} \sum_{k \in M_{\text{CV}}} x_{0jk}^{\text{CV}} \leq m_{\text{CV}} \quad (9)$$

$$\sum_{k=1}^{m_{\text{EV}}} p_{ktj} \leq GP_t, \forall t \in [0, T_0], \forall j \in D_k \quad (10)$$

$$p^{\text{Min}} \times y_{kt} \leq p_{ktj}, \forall t \in [0, T_0], \forall k \in M_{\text{EV}}, \forall j \in D_k \quad (11)$$

$$p_{ktj} \leq p^{\text{Max}} \times y_{kt}, \forall t \in [0, T_0], \forall k \in M_{\text{EV}}, \forall j \in D_k \quad (12)$$

$$SoC_k^0 + \frac{\sum_{t \leq T_0-1} \delta \times p_{ktj}}{CE_k} \leq SoC_k^{\text{Max}}, \forall t \in [0, T_0], \forall k \in M_{\text{EV}}, \forall j \in D_k \quad (13)$$

$$SoC_k^0 + \frac{\sum_{t \leq T_0-1} \delta \times p_{ktj}}{CE_k} \geq SoC_k^{\text{Min}}, \forall t \in [0, T_0], \forall k \in M_{\text{EV}}, \forall j \in D_k \quad (14)$$

$$tt_{jk} \geq tt_{ik} + (t_{i,j} + s_i) \times x_{ijk}^{\text{EV}} - T \times (1 - x_{ijk}^{\text{EV}}), \forall i \in V \cup \{0\}, \forall j \in V'_{n+1}; i \neq j, \forall k \in M_{\text{EV}} \quad (15)$$

$$tt_{jk} \geq tt_{ik} + (t_{i,j} + s_i) \times x_{ijk}^{\text{CV}} - T \times (1 - x_{ijk}^{\text{CV}}), \forall i \in V \cup \{0\}, \forall j \in V \cup \{n+1\}; i \neq j, \forall k \in M_{\text{CV}} \quad (16)$$

$$tt_{jk} \geq tt_{ik} + t_{i,j} \times x_{ijk}^{\text{EV}} + \sum_{t=T_0}^{t=T} \delta \times y'_{tik} - (T + r \times CE_k) \times (1 - x_{ijk}^{\text{EV}}), \forall i \in F'_k \cup D'_k, \forall j \in V'_{n+1}, \forall k \in M_{\text{EV}} \quad (17)$$

$$T_0 \leq tt_{jk} \leq T, \forall j \in V'_{0,n+1}, \forall k \in M_{\text{EV}} \cup M_{\text{CV}} \quad (18)$$

$$a_j \leq tt_{jk} + W_{jk} \leq b_j, \forall j \in F'_k \cup D'_k, \forall k \in M_{\text{EV}} \quad (19)$$

$$SoC_k^{\text{Min}} \leq \frac{E_{jk}^A}{CE_k} \leq SoC_k^{\text{Max}}, \forall j \in V'_{n+1}, \forall k \in M_{\text{EV}} \quad (20)$$



$$SoC_k^{\min} \leq \frac{E_{jk}^L}{CE_k} \leq SoC_k^{\max}, \forall j \in V'_{n+1}, \forall k \in M_{EV} \quad (21)$$

$$E_{jk}^A = E_{jk}^L, \forall j \in V, \forall k \in M_{EV} \quad (22)$$

$$E_{jk}^A \leq E_{ik}^L - r \times d_{i,j} \times x_{ijk}^{EV} + CE_k(1 - x_{ijk}^{EV}), \quad (23)$$

$$\forall i \in V'_0, \forall j \in V'_{n+1}, \forall k \in M_{EV}$$

$$E_{jk}^L = E_{jk}^A + \sum_{t=T_0}^{t=T} d \times p'_{tjk}, \forall j \in F'_k \cup D'_k, \forall k \in M_{EV} \quad (24)$$

$$0 \leq p'_{tjk} \leq p_j \times y'_{tjk}, \forall j \in F'_k \cup D'_k, \quad (25)$$

$$\forall t \in [T_0, T], \forall k \in M_{EV}$$

$$\sum_{k \in M_{EV}} \sum_{j \in D'_k} p'_{tjk} \leq GP_t, \forall t \in [T_0, T] \quad (26)$$

$$\sum_{t=b_j}^{t=T} y'_{tjk} + \sum_{t=T_0}^{t=a_j} y'_{tjk} = 0, \forall j \in F'_k \cup D'_k, \forall k \in M_{EV} \quad (27)$$

$$l_{jk} \geq l_{ik} + q_j - Q^{EV} \times (1 - x_{ijk}^{EV}), \quad (28)$$

$$\forall i \in V'_0, j \in V \cup \{n+1\}, i \neq j, \forall k \in M_{EV}$$

$$l_{jk} \geq l_{ik} + q_j - Q^{CV} \times (1 - x_{ijk}^{CV}), \quad (29)$$

$$\forall i \in V \cup \{0\}, j \in V \cup \{n+1\}, i \neq j, \forall k \in M_{CV}$$

$$l_{jk} \geq l_{ik} - Q^{EV} \times (1 - x_{ijk}^{EV}), \quad (30)$$

$$\forall i \in V'_{n+1}, j \in F'_k \cup D'_k, i \neq j, \forall k \in M_{EV}$$

$$l_{jk} \leq Q^{EV}, \forall j \in V'_{n+1}, \forall k \in M_{EV} \quad (31)$$

$$l_{jk} \leq Q^{CV}, \forall j \in V \cup \{n+1\}, \forall k \in M_{CV} \quad (32)$$

$$W_{jk} \geq a_j - tt_{jk}, \forall j \in F'_k \cup D'_k, \forall k \in M_{EV} \quad (33)$$

$$x_{ijk}^{EV}, x_{ijk}^{CV}, y_{kt}, y'_{tjk} \in \{0, 1\}; E_{jk}^L, E_{jk}^A, l_{jk}, p_{tk}; \quad (34)$$

$$p'_{tjk}, tt_{jk}, l_{jk}, W_{jk} \geq 0, \forall i, j, t, k \in V'_{0,n+1}$$

$[T_0, T]$  and constraints (19) make sure that the arrival time to any charging station should be in its operating period. Constraints (20) and (21) enforce that the energy amount available when arriving at node  $j$  or when leaving it never exceeds the maximum allowable SoC or goes below the minimum allowable SoC. Constraints (22) guarantee that the amount of energy available when arriving at a customer  $j$  is equal to the energy amount available when leaving it. Constraints (23) link the amounts of energy available when arriving to nodes  $i$  and  $j$ . Constraints (24) take into account the quantity of energy charged at charging station  $j$ . Constraints (25) ensure that the power used to charge an EV during a given time period does not exceed the charging power that could be delivered by the chargers of this station. Constraints (26) prohibit the exceeding of the available grid's maximum capacity. Constraints (27) prohibit any charging at a charging station outside its opening hours. Constraints (28)-(32) ensure the respect of the vehicles capacity constraints. The waiting time for vehicle  $k$  at charging station  $j$  is defined by the constraints 33. Finally, constraints (34) define the domain of all used variables.

## V. SOLVING APPROACHES

### A. Charging Routing Heuristic

In this section, we introduce a Charging Routing Heuristic (CRH) to solve the VRP-HFCC. This heuristic is used to generate initial solutions within a short computational time and it consists of two steps. In the first step, a feasible charging scheme for EVs at the depot during  $[0, T_0]$  is generated. In the second step, a joint routing and charging planning for the service period  $[T_0, T]$  is determined.

#### Step 1: EVs Charging at the depot during $[0, T_0]$ :

This step aims at designing EVs charging strategies at the depot during the time interval  $[0, T_0]$  while satisfying the electricity grid and the chargers constraints. Those charging constraints could limit EV charging in the sense that the available EVs will not necessarily be fully charged at  $t = T_0$ . Our objective is to minimize charging costs while at the same time giving the priority of charging to the EVs (i) with low operating costs, (ii) whose state of charge is still very low, (iii) which are not compatible with all available charging technologies and (iv) whose charging is more costly.

At  $t = 0$ , it is assumed that all batteries of EVs are empty. The heuristic starts by sorting the time periods according to the ascending order of electricity costs. Let  $T_{sorted}$  be the sorted table of all time periods in  $[0, T_0]$ . With each electric vehicle  $k$ , we associate a priority  $priority_t^k$  that translates the fact that EV  $k$  has or not higher priority to charging than the other available EVs during the time period  $t$ . This priority is computed as:  $priority_t^k = \frac{SoC_k^t}{SoC_{max}^t} + \frac{Comp\_Cost_k}{Comp\_Cost_{total}} + \frac{1}{km \times OC_k^{EV}}$ , where:

- $km$  is an estimation of the average number of kilometers traveled by each electric vehicle,
- $Comp\_Cost_k = \sum_{t=T_0}^{t=T} \frac{1}{c_t}$ ; where  $\bar{c}_t$  is the average charging cost at all charging stations in which vehicle

The objective function, measured in monetary units, consists in minimizing five costs: (i) the routing cost, (ii) the charging cost engendered by charging EVs in the charging stations during  $[T_0, T]$ , (iii) the cost of charging EVs at the depot during  $[0, T_0]$ , (iv) the vehicles total fixed cost and (v) the total cost engendered by the waiting time; where  $\beta$  is a coefficient estimating the cost lost because of waiting during one period of time.

Constraints (2) ensure that each customer is visited exactly once. Constraints (3) guarantee that each charging station is visited at most one time. Constraints (4) and (5) enforce that the number of incoming arcs is equal to the number of outgoing arcs for each node. Constraints (6) and (7) ensure that each vehicle is at most assigned to one route. Constraints (8) and (9) make sure that at most the available vehicles are used. Constraints (10)-(14) concern charging EVs at the depot. Constraints (10) ensure that, at each time period  $t \in [0, T_0]$ , the total power used to charge the EVs does not exceed the grid's maximum capacity. Constraints (11) and (13) guarantee the respect of the minimum and the maximum powers of chargers when charging the EVs. Constraints (13) and (14) ensure that the SoC of each EV is in the interval  $[SoC^{\min}, SoC^{\max}]$ . Constraints (15)-(17) link arrival times at nodes  $i$  and  $j$  and permit to eliminate the sub-tours. Constraints (18) ensure that the arrival time to each node should be in the interval

$k$  could be charged during the time period  $t$ ; i.e.,  $\bar{c}_t = \sum_{f \in F(k)} (\frac{1}{|F(k)|}) \times c_{ft}$ ; where  $f \in F(k)$  if and only if EV  $k$  is compatible with the charging technology proposed by  $f$ ,

$$\bullet \text{ } Comp\_Cost_{total} = \sum_{t=T}^{t=T_0} \frac{1}{\sum_{f \in F(k)} (\frac{1}{|F(k)|}) \times c_{ft}}.$$

The heuristic selects the first available time period in  $T_{sorted}$  as well as the EV with the lowest priority and charges it with the minimal possible charging power between: (i) the maximal power of chargers, (ii) the grid's capacity that is still available, and (iii) the maximum power that will completely full the vehicle's battery. The grid's capacity is then updated and if the new grid's capacity is still positive, the CRH selects a new different EV with the lowest priority. This procedure is repeated until no possible charging could be undertaken. At the end of the first step, a charging scheme is available for all EVs.

### Step 2: Joint charging and routing during $[T_0, T]$ :

Initially, a list of  $m_{EV} + m_{CV}$  empty routes is created. While at least an EV is still available, the heuristic continues with selecting an EV  $k$  with a maximum priority ( $priority_k^k$ ). Then, it inserts iteratively the customers into an active route at the position causing minimal increase in tour cost until a violation of capacity or battery capacity of the selected EV occurs. The heuristic anticipates, when possible, any violation due to the battery capacity constraint by inserting charging stations during the tour construction. The best charging station is selected among the compatible and available charging stations belonging to the neighborhood  $V(i)$  of the current node  $i$ , where  $V(i)$  is the set of all nodes within the circle defined by the center  $i$  and the radius  $\alpha$ ; where  $\alpha$  is the maximum distance that could be traveled by the EV using its current state of charge (see Fig. 1)). If a violation of one of the constraints occurs or the total route time exceeds  $T - T_0$ , the current route is assigned to the selected vehicle, another EV with a maximum priority is selected and a new route is activated.

When a customer could not be reached using any of the available EVs, it is assigned to the CV engendering the minimal cost increase in the solution cost while satisfying the capacity and the total route duration constraints, until at most the predefined number of routes ( $m_{EV} + m_{CV}$ ) is constructed. Algorithm 1 provides more details about the CRH heuristic.

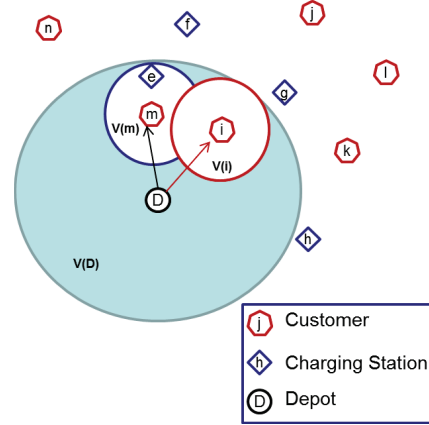


Fig. 1. Illustrative example of the CRH: Assume that an available EV at the depot could reach all nodes in  $V(D)$  and the cost of the arc  $(D, i)$  is minimal. Before adding  $i$  to the active route, the heuristic checks the neighborhood of  $i$  (list of all reachable nodes). In our case,  $V(i) = \emptyset$ . Thus, if the EV visits the customer  $i$ , it will be blocked there since its state of charge will not be sufficient to go back to the depot. The heuristic continues by selecting the node  $m$  and checking its neighborhood. In our case,  $V(m) = e$ . The customer  $m$  will then be added to the active route since the EV will be able to get charged after visiting  $m$ .

### Algorithm 1 Charging Routing Algorithm

- 1: **Input:** A graph  $G = (V', A)$  and a set of  $m_{EV} + m_{CV}$  empty routes
- 2: **Output:** A set of routes assigned to at most  $m_{EV} + m_{CV}$  vehicles
- 3: **Step 1**
- 4: Let  $C = (c'_1, \dots, c'_{T_0})$ ,  $G = (g_1, \dots, g_{T_0})$  and  $E = (e^k_1, \dots, e^k_{T_0})$  be three vectors of  $T_0$  elements, where  $c'_t$ ,  $t = 1, \dots, T_0$ , is the electricity cost during the time interval  $[t-1, t]$ ,  $g_t$ ,  $t = 1, \dots, T_0$ , is the residual capacity of the electricity grid during the time interval  $[t-1, t]$  and  $e^k_t$ ,  $t = 1, \dots, T_0$ ,  $k = 1, \dots, m_{EV}$ , is the quantity of energy injected in the battery of EV  $k$  during the time interval  $[t-1, t]$ . Parameter  $e^k_t$  is initialized to zero.
- 5: Sort the vector  $C$  in the nondecreasing order of  $c'_t$  and let  $C = (c'_{\pi(1)}, \dots, c'_{\pi(T_j)})$  be the sorted vector
- 6: **for** each time interval  $[\pi(t) - 1, \pi(t)]$  such that  $g_{\pi(t)} > 0$  **do**
- 7:   **while** Charging could be undertaken; i.e., there exists at least one available charger and one vehicle  $k$  such that  $e^k_{\pi(t)} = 0$  and  $v_k > 0$ ; where  $v_k$  is maximum quantity of energy that could be injected in the battery of vehicle  $k$  without exceeding the maximum allowable State of Charge **do**
- 8:     Compute the priority associated with each EV and select the EV with the lowest priority among all vehicles having  $e^k_{\pi(t)} = 0$  and  $v_k > 0$
- 9:     Calculate  $Energy\_to\_eject^{\pi(t)} = \min\{p^{Max} \times \delta, g_{\pi(t)} \times \delta, v_k\}$
- 10:     Update  $e^k_{\pi(t)} = Energy\_to\_eject^{\pi(t)}$  and  $g_{\pi(t)} = g_{\pi(t)} - \frac{Energy\_to\_eject^{\pi(t)}}{\delta}$
- 11:   **end while**
- 12: **end for**
- 13: The charging schedule at the depot is given, for each vehicle  $k$ , by the power vector that should be applied to EV  $k$   $P = (\frac{e^k_1}{\delta}, \dots, \frac{e^k_{T_0}}{\delta})$ .

**14: Step 2**


---

```

15: while the maximum number of routes is not yet reached AND
    there exists at least one customer that is not yet served do
16:   Select the EV with the highest priority  $priority_{T_0}^k$  at  $t = T_0$ 
    among all available EVs not yet assigned
17:   while the total route duration is less than  $T - T_0$  and the total
    amount of goods delivered along the route is less than  $Q^{EV}$  do
18:     Sort the list of nodes either randomly or in increasing order
    of the angle between the depot and a randomly chosen point and
    select the first customer  $i$  in the list
19:     Let  $V(i)$  be the set of all neighbors of node  $i$  not yet visited
    and that could be visited using the remaining battery energy of the
    current vehicle.
20:     If  $V(i)$  contains at least one customer and either the depot
    or a charging station  $f$ ;  $f \in V(j) \cap F(j)$ , select a node  $j$  from
     $V(i)$  such that the cost  $cost_{i,j,k}^{EV}$  of arc  $(i, j)$  is minimal.
21:     if  $V(j)$  is empty or it only contains customers then
22:       the vehicle should get charged before visiting  $j$ , in that
       case insert the compatible charging station with the lowest cost
       while ensuring that this charging station will be available when
       the EV arrives at this station
23:     end if
24:     if ( $V(j)$  is empty or it contains only customers or incom-
    patible charging stations) AND (charging is not possible) then
25:       Assign  $i$  to the CV having a sufficient capacity and
       engendering a minimum insertion cost
26:     end if
27:   end while
28: end while

```

---

**B. Inject-Eject Routine-Based Local Search**

In this section, we propose a new Inject-Eject-based Local Search (IELS) which starts from a given feasible solution and improves it using the inject-eject routine. The heuristic CRH is used to generate an initial solution to the problem.

The following parameters are useful in IELS method:

- *Iter*: parameter that controls the size of the main loop of the algorithm.
- *IterIE*: parameter that specifies the number of times the inject-eject routine should be repeated.
- *Num*: parameter that controls the size of the neighborhood list that will be used in the inject-eject procedure.

For each feasible solution, IELS heuristic performs *Iter* iterations of the following neighborhood ejection and injection strategy.

A node  $j$  and a set of  $Num - 1$  additional nodes located the nearest possible to  $j$  (in terms of costs), are randomly selected (the selected neighbors may be in different routes and are not necessary in  $V(j)$ ).

This neighborhood of  $Num$  nodes is then ejected from the solution. It is possible to eject a charger or to decrease the charging time at a given charging station to satisfy the charging station operating time windows constraints, in that case, the solution may become unfeasible, thus, a penalty is then added to the total solution cost. The ejected nodes are then re-inserted back into the partial solution using one of the three different insertion methods: (i) random insertion, (ii) insertion method with regret search and (iii) score-based insertion method.

If the solution becomes infeasible, we insert a new charger,

having the lowest cost, in the route while ensuring that the constraints related to the compatibility of the charging stations with the EV as well as the station's operating time windows constraints are satisfied.

If it is not possible to insert the ejected node in an already constructed route, a new route that contains this node and the depot may be created. In that case, the vehicle ownership cost is added to the total route cost.

When all customers have been re-inserted back into the solution using one of the three insertion methods, the new solution is compared with the original solution. If the resulting solution is better than the original solution, then the next iteration continues with the new solution. Otherwise, the next iteration continues with the original solution. After *iter* runs, the best solution found during the search is reported.

In the following, we detail the insertion methods as well as the inject-eject algorithm.

*1) Random Insertion Method:* This method consists in randomly selecting a node among the list of ejected nodes and inserting it in the position that generates the minimal cost increase in the total solution cost. If the insertion of a customer in a given route position leads to a violation of the vehicle capacity or total time constraints, this route position will not be accepted. However, if the insertion of a customer in a given route position still satisfies the vehicle capacity and total time constraints but leads to a violation of the energy constraints (in the case where the EV needs more energy to serve this customer or the time planned for charging decreases since it depends on the opening time windows of the charging stations), this method tries to repair the solution by inserting chargers in the route while ensuring the compatibility between the EV and the chargers and satisfying the charging stations' operating time windows constraints. At each update of the routing and charging solution, the total solution cost is updated. Algorithm 2 provides the detail of the random insertion method.

**Algorithm 2** Random Insertion Method

---

```

1: Input: A partial solution to the VRP-HFCC and a list of ejected
   nodes Eject
2: Output: A solution to the VRP-HFCC
3: Let trial be the number of times the Random Insertion Method
   should be repeated for each list of ejected nodes
4: Let best_increase_cost be the cost of the best solution found.
   Initially, best_increase_cost =  $\infty$ , tr = 0
5: best_total_solution_cost = cost of the best solution found
6: while tr < trial do
7:   Generate a list of ejected nodes Eject and sort it randomly
8:   while Eject  $\neq \emptyset$  do
9:     Select a node  $j$  from the list Eject
10:    for each route position do
11:      try to inject  $j$  in this route position
12:      if the insertion is possible and increase_cost <
         best_increase_cost then
13:        best_increase_cost = increase_cost
14:      else
15:        if the insertion satisfies the total load and total time
         constraints and violates the energy constraints then
16:          try to inject a charger using the Charger Insertion
         Method

```

---

---

```

17:         if increase_cost < best_increase_cost then
18:             best_increase_cost = increase_cost
19:         end if
20:     end if
21:
22:     do not accept this insertion
23: end if
24: end for
25: Evaluate the cost of inserting j in a new route
26: if increase_cost < best_increase_cost then
27:     best_increase_cost = increase_cost
28: end if
29: end while
30: if total_solution_cost < best_total_solution_cost then
31:     best_total_solution_cost = total_solution_cost
32:     save the best order of injected nodes
33: end if
34: end while
35: inject all nodes of Eject and eventually some chargers in the
    best route positions defined

```

---

2) *Insertion Method With Regret Search*: The insertion method with regret search uses the same cheapest insertion method as the random insertion method, but allows previous insertions to be undone if this removal allows for a cheaper insertion of the current customer under consideration. This is similar to the notion of regret described in [19].

At each step, the cheapest next insertion and the maximum cost reduction caused by deleting a node (which is not one of the partial solution vertices participating in the insertion) from the current partial solution are compared. The inject-eject moves remain temporary and become final only when all ejected nodes are re-injected. Algorithm 3 describes the insertion method with regret search.

---

#### Algorithm 3 Insertion Method with Regret Search

---

```

1: Input: A partial solution and a list Eject of ejected nodes
2: Output: A set of routes
3: Let trial be the number of times the Insertion Method with regret
   should be repeated for each list of ejected nodes
4: Initially, node_to_eject = -1, max_eject_cost =  $-\infty$ 
5: for (tr = 0; tr < trial) do
6:     Create a random permutation of the list Eject
7:     Let RandEject be the new list of ejected nodes engendered
       by the random permutation
8:     for (j = 0; j < Num) do
9:         Find the cheapest way to insert the current node (including
           creating a new route) and eventually the best charging station  $f^*$ 
           and the best route position  $p^*$  to insert it
10:        for (k = 0; k < j) do
11:            if the node RandEject[k] isn't involved in the cheapest
               insertion then
12:                if the ejection cost of RandEject[k] is greater than
                   the maximum ejection cost (max_eject_cost) then
13:                    node_to_eject = RandEject[k]
14:                end if
15:            end if
16:        end for
17:        if there is no node to eject OR the cost of insertion of
           RandEject[j] is greater than max_eject_cost then
18:            Insert RandEject[j] in the route position engendering
               the minimum cost increase in the solution cost

```

---



---

```

19:         insert eventually  $f^*$  in  $p^*$ 
20:     else
21:         Eject the node node_to_eject and insert the node
           RandEject[j] in the cheapest insertion position as well as even-
           tually  $f^*$  in  $p^*$ 
22:     end if
23: end for
24: end for

```

---

3) *Score-Based Insertion Method*: The Score-Based Insertion Method is based on the idea of associating a score with each node to inject. This idea is inspired from the Parallel Regret Algorithm introduced in [20].

For this method, the *Eject* list is only composed of customers. However, it is possible to inject and eject chargers to repair a solution. A score is associated with each node of the *Eject* list. It translates the difficulty of injecting the node in the current solution and it is used to select the next customer *j* to inject in the current solution. For each node  $j \in Eject$ , the score  $score(j)$  of node *j* includes (i) the penalty  $penalty(j, r)$  occurred when the customer *j* is not assigned to its preferred route; (ii) the distance of *j* to the closest available charging station ( $d_{j, f_c}$ ); (iii) the number of available charging stations that could be reached by the EV after visiting customer *j* ( $\eta_f$ ).

Thus,  $score(j) = penalty(j, r) + d_{j, f_c} - \eta_f$ ,

where  $penalty(j, r)$  is the difference between the cost engendered by inserting the customer *j* in the second best route position and the cost of inserting it in the best route position. When a customer could only be inserted in one possible route position, the penalty  $penalty(j, r)$  takes a large value imposing that the customer should be inserted in the best route position.

Algorithm 4 details the Score-Based Insertion Method.

---

#### Algorithm 4 Score-Based Insertion Method Algorithm

---

```

1: Input: A partial solution and a list Eject of ejected customers
2: Output: A solution to the VRP-HFCC
3: for each customer j in Eject do
4:     Compute the costs engendered by injecting j in each route
       position including the possibility of injecting a charger when
       needed to repair the solution
5:     compute the score  $score(j)$ 
6: end for
7: while Eject  $\neq \emptyset$  do
8:     Select the customer  $j^*$  with the maximum value of score
9:     if the cost of injection of  $j^*$  in a new route is cheaper then
       the cost of inserting  $j^*$  in any of the already constructed routes
       then
10:        initialize a new route  $\tilde{r}$ 
11:        inject  $j^*$  in this route
12:     else
13:        insert  $j^*$  in the best route position
14:        delete  $j^*$  from Eject
15:     end if
16:     for each customer j in Eject do
17:         update the score  $score(j)$ 
18:     end for
19: end while

```

---

In the following, we detail the *Charger Insertion Method*



which is used by the different insertion methods when the energy constraints are not satisfied.

When an ejected node has to be injected in the solution, different route positions are evaluated and a violation of the energy constraints may occur. Rather than excluding this route position, we try to repair the solution by injecting a charger in the unfeasible route. This Charger Insertion Method searches the best charging node and the best route position that engender the lowest charging cost and guarantee the feasibility of the route. The detail of the Charger insertion method is given in Algorithm 5.

---

**Algorithm 5** Charger Insertion Method

---

```

1: Input: A partial solution with an infeasible route  $\hat{r}$  assigned to EV  $\hat{k}$ 
2: Output: A feasible solution
3: Initially,  $feasible\_route\_cost = \infty$ 
4: for each charging station  $\hat{f}$  compatible with EV  $\hat{k}$  do
5:   for each position  $\hat{p}$  in route  $\hat{r}$  do
6:     if route  $\hat{r}$  becomes feasible when  $\hat{f}$  is inserted in  $\hat{p}$  AND the time operating constraints of  $\hat{f}$  are satisfied then
7:       Adjust the amount of energy to inject using the Charging Adjustment Procedure
8:       if  $feasible\_route\_cost < best\_feasible\_route\_cost$  then
9:          $p^* = \hat{p}$ 
10:         $f^* = \hat{f}$ 
11:       end if
12:     end if
13:   end for
14: end for

```

---

Algorithm 6 provides details on the *Charging Adjustment Procedure* used to estimate the minimum required amount of energy to inject in the EV when charging should be undertaken.

---

**Algorithm 6** Charging Adjustment Procedure

---

```

1: Input: Initial feasible solution  $r$ 
2: Output: Improved feasible route  $r'$ 
3: Initialize  $r'$  to  $r$ 
4: Let  $(d_0, p_1, \dots, f_x, \dots, p_r, d_{n+1})$  be the sequence of customers and charging stations in route  $r'$ .
5: for each subsequence of customers between two charging stations do
6:   Calculate the minimum required amount of energy that should be injected in the EV at the next charging station.
7: end for
8: Update the total cost and duration of  $r'$ .

```

---

Now, we have all sub-routines to describe the Inject-Eject method. In the following, Algorithm 7 describes the Inject-Eject method in detail.

---

**Algorithm 7** Inject-Eject Method

---

```

1: Input: A graph  $G = (V', A)$  and a set of  $m_{EV} + m_{CV}$  vehicles
2: Output: A set of routes assigned to at most  $m_{EV} + m_{CV}$  vehicles
3: Let  $S$  be the best solution obtained by the Charging Routing Heuristic. Initially,  $best\_obj = solution\_cost(S)$  and  $d = 0$ 
4: for  $d = 0; d < Iter$  do
5:   Let  $S$  be the current solution and initialize  $d$  to 0
6:   for  $a = 0; a < IterIE$  do
7:     Select, randomly, a node  $j$  (different from the depot) and  $Num - 1$  nodes from the list of neighbors of node  $j$  to eject. Let  $Eject$  be the list of ejected nodes.
8:     Inject all nodes of the list  $Eject$  again using one of the three insertion methods
9:     If the insertion method leads to an unfeasible route, repair the solution by including a charger using the Charger Insertion Method
10:    If a new route is created, add the vehicle possession cost to the total solution cost
11:    if  $total\_route\_cost < best\_obj$  then
12:       $best\_obj = total\_route\_cost$ 
13:      Update  $S$ 
14:    end if
15:  end for
16:  if  $total\_route\_cost < best\_obj$  then
17:     $best\_obj = total\_route\_cost$ 
18:    Update  $S$ 
19:  end if
20: end for

```

---

## VI. COMPUTATIONAL RESULTS

We conducted numerical experiments on real data instances provided by two French companies that manage large heterogenous vehicle fleets. Our heuristics are implemented using C++. All experiments were carried out on an Intel Xeon E5620 2.4GHz processor, with 8GB RAM memory. The half of EVs considered have 22 kWh battery packs and the rest have 16 kWh battery packs.

The optimization procedure is based on a 24 hour period. EVs charging at the depot could be performed during the time interval  $[8pm, 8am]$ . The customers should be served not earlier than 8 am and not later than 8 pm.

Concerning charging at the depot, prices for electricity are based on those provided by EDF (French Electricity Distribution company). At most  $m_{EV}$  Level 1 chargers, with a range of 1.5-3.7 kW could be used to charge EVs at the depot during all the optimization horizon. One Level 2 charger, with a range of 1.5-22.0 kW, could be used to charge EVs during the day. The minimal allowable SoC is fixed at 0.2 and the maximal allowable SoC is fixed at 0.95. Initially, the EVs batteries are empty ( $SoC^{t=8pm} = 0$ ).

The operating cost of each EV expressed in (*euros/kilometer*) is calculated as the sum of different costs engendered by the maintenance, accidents, etc. This cost does not include the electricity costs which are computed separately and include EVs charging costs at the depot and other charging stations. Concerning the CVs, their operating costs include the costs engendered by the maintenance, accidents, etc., as well as the gasoline cost which is calculated by multiplying the gasoline consumption per kilometer by the cost per unit of gasoline. Concerning charging at the different external charging stations, we consider only stations proposing slow charging (with Level 1 chargers) or medium charging (using

TABLE I  
PERFORMANCE RESULTS OF THE DIFFERENT HEURISTIC METHODS ON 9 REAL DATA INSTANCES

Instance	Number of nodes	CRH-Objective	CRH-CPU(s)	IELS-Rand-Gap	IELS-Rand-CPU(s)	IELS-Regret-Gap	IELS-Regret (s)-CPU(s)	IELS-Score-Gap	IELS-Score-CPU(s)
1	550	2840.01	1.19	-38.00%	88.61	-39.31%	601.41	-38.64%	208.11
2	550	2486.91	1.12	-13.09%	93.85	-14.91%	623.63	-13.38%	230.62
3	500	2398.22	1.01	-30.6%	81.68	-30.6%	472.86	-19.03%	142.18
4	500	2266.54	0.99	-14.26%	83.47	-16.19%	305.32	-14.26%	149.81
5	400	2249.35	0.89	-15.28%	60.71	-15.28%	200.75	-15.28%	100.64
6	400	2939.24	0.89	-37.60%	54.49	-37.60%	209.69	-37.60%	101.29
7	400	2966.26	0.82	-40.24%	43.65	-44.30%	358.56	-43.29%	102.29
8	300	1946.58	0.80	-64.40%	29.19	-64.07%	194.25	-64.07%	81.79
9	300	1829.70	0.79	-37.49%	31.42	-64.00%	126.69	-62.20%	91.59
Average	433	2435.86	0.96	-32.32 %	63.00	-36.25%	343.68	-34.19%	134.25

Level 2 chargers). Charging stations proposing fast charging using Level 3 chargers are not considered within those experiments since our real data instances include only EVs that are not compatible with fast charging.

Experiments were conducted on 9 real data instances. The number of nodes for the considered instances ranges between 300 and 550. We consider that 18 EVs and 8 CVs are available to serve the customers. At the depot, 18 Level 1 chargers and 1 Level 2 charger are available. The number of external charging stations is adapted to the total number of customers  $n$  of the instances. More precisely, one to two charging stations are randomly located for each set of 20 customers. The characteristics of each charging station as well as the time-dependent proposed costs are defined randomly. The load capacity of each EV ranges between 3 and 5  $m^3$ . Each CV has a capacity of 5  $m^3$ .

The computational results obtained with the different heuristic methods are summarized in Table I. The entries of Table I show, for the CRH, the value of the objective function of the generated solution as well as the average run time in seconds (s) and for the three other methods (IELS-Rand, IELS-Regret, IELS-Score), the Gap of the generated solution (s) in relation to the solution generated by the CRH ( $s_{CRH}$ ) computed as:  $Gap = \frac{s - s_{CRH}}{s}$  as well as the average run time in seconds (s). The computational results show that the three different IELS heuristics have generated the same solution for 4 instances. For the remaining instances, we can notice that the IELS with regret insertion method generates better solutions than the other two methods. In fact, the IELS with regret insertion method has an average improvement gap in relation to the CRH of about 36%, compared to 34% for the score-based IELS and 32% for the IELS with regret insertion strategy. Concerning the computational time, among all the IELS methods, the IELS with random insertion method seems to be the fastest with an average CPU of around one minute compared to around 2 minutes for the score-based IELS and around 5 minutes for the IELS with regret insertion method.

## VII. CONCLUSION

In this paper, we considered a new vehicle routing problem with mixed fleet of conventional and heterogeneous electric vehicles and time dependent charging costs. This problem consists in optimizing the routing of a set of vehicles with the objective of minimizing the overall routing

and charging costs. Contrary to existing studies that focus on the Electric Vehicle Routing Problem, we consider that the charging stations, which are subject to operating time windows constraints, propose charging using different charging technologies and time dependent charging costs. Moreover, charging at the depot is subject to the electricity grid and the chargers constraints. We also consider the compatibility constraints between the EVs and the different charging technologies. To solve this problem, we developed a Charging Routing Heuristic to generate initial solutions as well as an Inject-Eject-Based Local Search with three different insertion strategies. All heuristic methods were tested on real data instances.

As further work, we will test our methods on newly designed data instances as well as on benchmark instances of some related problems. Moreover, we will consider some classical meta-heuristics to solve our problem.

## REFERENCES

- [1] K. Aissat and A. Oulamara, "A posteriori approach of real-time ridesharing problem with intermediate locations," in *proceedings of ICORES 2015*, 2015.
- [2] O. Sassi and A. Oulamara, "Simultaneous electric vehicles scheduling and optimal charging in the business context: Case study," IET, 2014.
- [3] O. Sassi and Oulamara, "Joint scheduling and optimal charging of electric vehicles problem," in *Computational Science and Its Applications-ICCSA 2014*. Springer, 2014, pp. 76–91.
- [4] S. Pelletier, O. Jabali, and G. Laporte, "Goods distribution with electric vehicles: Review and research perspectives," 2014.
- [5] M. Ramezani, M. Graf, and H. Vogt, "A simulation environment for smart charging of electric vehicles using a multi-objective evolutionary algorithm," in *Information and Communication on Technology for the Fight against Global Warming*, vol. ICT-GLOW 2011, LNCS 6868, 2011, p. pp. 5663.
- [6] J. Lee, G. Park, H. Kwak, and H. Jeon, "Design of an energy consumption scheduler based on genetic algorithms in the smart grid," in *Computational Collective Intelligence. Technologies and Applications*, vol. ICCCI 2011, Part I, LNCS 6922, 2011, p. 438447.
- [7] J. Kang, S. J. Duncan, and D. N. Mavris, "Real-time scheduling techniques for electric vehicle charging in support of frequency regulation," *Procedia Computer Science*, vol. 16, pp. 767–775, 2013.
- [8] P. Toth and D. Vigo, *The vehicle routing problem*. Siam, 2001.
- [9] C. B. e. P. W. Ramdane cherif, M. Haj Rachid, "New notation and classification scheme for vehicle routing problems," *RAIRO*, vol. DOI 10.1051/ro/2014030, p. To appear, 2014.
- [10] A. Artmeier, J. Haselmayr, M. Leucker, and M. Sachenbacher, "The optimal routing problem in the context of battery-powered electric vehicles," in *Workshop CROCS at CPAIOR-10, 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*, 2010.
- [11] S. Erdogan and E. Miller-Hooks, "A green vehicle routing problem," *Transport. Res.*, vol. Part E 48, pp. 100–114, 2012.

- [12] C. Lin, K. Choy, G. Ho, S. Chung, and H. Lam, "Survey of green vehicle routing problem: Past and future trends," *Expert. Syst. Appl.*, vol. 41, pp. 1118–1138, 2014.
- [13] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle routing problem with time windows and recharging stations," University of Kaiserslautern, Germany, Tech. Rep., 2012.
- [14] D. Goeke, M. Schneider, and D. S. E. A. Professorship, "Routing a mixed fleet of electric and conventional vehicles," Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), Tech. Rep., 2014.
- [15] G. Hiermann, J. Puchinger, and R. F. Hartl, "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," Working Paper. Accessed July 17, 2014. URL: [http://prolog.univie.ac.at/research/publications/downloads/Hie\\_2014\\_638.pdf](http://prolog.univie.ac.at/research/publications/downloads/Hie_2014_638.pdf), Tech. Rep., 2014.
- [16] Á. Felipe, M. T. Ortuno, G. Righini, and G. Tirado, "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges," *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, pp. 111–128, 2014.
- [17] S. Bashash, S. J. Moura, J. C. Forman, and H. K. Fathy, "Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity," *Journal of Power Sources*, vol. 196, pp. 541–549, 2010.
- [18] A. Millner, "Modeling lithium ion battery degradation in electric vehicles," in *Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010 IEEE Conference on*. IEEE, 2010, pp. 349–356.
- [19] R. Hassin and A. Keinan, "Greedy heuristics with regret, with application to the cheapest insertion algorithm for the tsp," *Operations Research Letters*, vol. 36, no. 2, pp. 243–246, 2008.
- [20] M. Dell'Amico, M. Monaci, C. Pagani, and D. Vigo, "Heuristic approaches for the fleet size and mix vehicle routing problem with time windows," *Transportation Science*, vol. 41, no. 4, pp. 516–526, 2007.

**Ons Sassi** is a PhD student in Computer Science at Loria Laboratory in France. Her main interests are Operations Research and combinatorial optimization. Her current work concentrates on developing optimization models for the use, the routing and the charging of electric vehicles in a business context.

**Wahiba Ramdan Cherif** is an associate professor at the Ecole des Mines de Nancy (University of Lorraine), and LORIA laboratory in France. Her research topics are mainly operations research applied to optimization problems in vehicle routing and transportation.

**Ammar Oulamara** is a professor in Computer Science at the University of Lorraine, France. His research lies mainly in scheduling and combinatorial optimization with applications in the context of electric vehicle routing and electro-mobility.