HAL
open science

Electric Vehicle Scheduling and Optimal Charging Problem: Complexity, Exact and Heuristic Approaches

O Sassi, A Oulamara

To cite this version:

O Sassi, A Oulamara. Electric Vehicle Scheduling and Optimal Charging Problem: Complexity, Exact and Heuristic Approaches. 2014. hal-01083888

HAL Id: hal-01083888

https://hal.science/hal-01083888v1

Preprint submitted on 18 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Electric Vehicle Scheduling and Optimal Charging Problem: Complexity, Exact and Heuristic Approaches

O. Sassi[1], A. Oulamara[2]

[1] *LORIA, Campus Scientifique - BP 239 - 54506 Vandoeuvre-lès-Nancy Cedex, France*

[2] *University of Lorraine, Ile du Saulcy, CS 10628, 57045, Metz, France*

Abstract: This paper deals with the Electric Vehicle Scheduling and Optimal Charging Problem. More precisely, given a fleet of Electric Vehicles - EVs and Combustion Engine Vehicles - CVs, a set of tours to be processed by vehicles and a charging infrastructure, the problem aims to optimize the assignment of vehicles to tours and minimize the charging cost of EVs, while considering several operational constraints mainly related to chargers, electricity grid, and EVs driving range. We prove that the Electric Vehicle Scheduling and Charging Problem (EVSCP) is NP-hard in the ordinary sense. We provide a mixed-integer linear programming formulation to model the EVSCP and use CPLEX to solve small and medium instances. To solve large instances, we propose two heuristics: a Sequential Heuristic - SH and a Global Heuristic - GH. The SH considers the EVs sequentially. To each EV, it assigns a set of tours and guarantees the feasibility of a charging schedule using the Maximum Weight Clique Problem. Then, it generates an optimal charging schedule for this EV using a Minimum Cost Flow formulation. However, the GH computes, in the first step, a feasible assignment of tours to all EVs. In the second step, it applies a global Min-Cost-Flow-based charging algorithm to minimize the charging cost of the EVs fleet. To evaluate the efficiency of our solving approaches, computational results on a large set of real and randomly generated test instances are reported and compared. Tested instances include large random instances with up to 200 EVs and 320 tours.

Keywords: Electric Vehicle, Optimal Charging, Complexity, Scheduling, Mixed Integer programming, Greedy Algorithms, Maximum Weight Clique, Minimum Cost Flow, Experiments.

1 Introduction

The transport sector is responsible, to a large extent, for energy consumption and greenhouse gas emissions. To tackle environmental and energy challenges in this sector, Electric Vehicles (EVs) seem to be a potential alternative that offers a reduction of both petroleum consumption and greenhouse gas emissions.

However, electric car industry is still facing many weaknesses related to battery management. The first is the limited EV driving range. The second weakness is related to the long charging time of EVs (fully charging the battery pack can take up to 8 hours with Level 1 chargers), and the availability of a charging infrastructure. The third weakness concerns the electricity grid on which EV charging may put a significant stress. A large scale adoption of EVs may then strongly affect the electricity grid which is already facing numerous challenges. Thus, the new ecosystem of EVs (vehicles - chargers - electricity grid) leads to new optimization challenges aiming to develop efficient models and decision tools to manage EV fleets and take into account all constraints related to the ecosystem.

This study is a part of the French National Project InfiniDrive, led by La Poste Group (Postal services company, France), ERDF (French enterprise that manages the public electricity distribution network) and six other companies and research laboratories. This R&D project has been selected by ADEME (French Environment and Energy Management Agency) as part of the *Vehicle of the Future* program. It aims at designing, with a progressive approach, an intelligent system that manages the charging infrastructures, and allows for an economical and ecological sustainable deployment of EVs fleet in the business context.

Optimization problems related to EVs can be classified into four classes: (i) problems related to EV used in the context of Smart Grid or Vehicle to Grid (V2G), (ii) controlled EV charging problems, (iii) charging infrastructure network design problems and (iv) EV routing problems.

In the V2G context, EVs batteries are used to store energy. Those batteries can serve as energy source by sending electricity back into the grid in order to either prevent or postpone load shedding. Research based on the V2G problems has mainly focused on how to connect the batteries of EVs to the power grid and characterize its feasible services [1], on proving the validity of the V2G concept [2], identifying new markets [3, 4], and controlling V2G systems [5, 6]. The goal of the V2G control is to decide whether the EV should be charged, discharged, or provide frequency regulation at each hour. Hutson et al. [5] consider the problem of maximizing the EVs owners' profit by selling excessive energy to grid. A V2G control algorithm, based on the application of a binary particle swarm optimization

technique, is proposed. Shi and Wong [6] consider the real-time control problem of V2G, and present a greedy algorithm that takes into account price uncertainty. Lopez et al. [7] address the problem of EV charging and V2G capabilities for congestion management. They propose a novel algorithm, based on power distribution factors, for optimal EV congestion management, in which EVs can decrease or increase their battery energy level, stop their charging, or even inject energy in the grid. Other studies related to the V2G include the works [8, 9].

The problem of the network design of the charging infrastructure is considered in a number of studies including the works [10, 11, 12, 13]. The problem of the design of charging infrastructures is addressed by Wang [10]. The author develops a model using an integer program in order to determine the locations of charging stations. Furthermore, Wang [11] presents an integer programming model to optimize the locations and the number of battery exchange stations. Wang and Lin [12] use the concepts of set covering problems to model the allocation of multiple types of charging stations. A mathematical program with complementarity constraints is developed by He et al. [13] to determine an optimal allocation of a given number of public charging stations.

The controlled EV charging problems consist of a better management of the charging load and include delayed, timed and more advanced types of smart charging, [14, 15, 16]. The objective of those problems is to integrate the charging load with the traditional load in the electricity grid and minimize the charging cost. Sundstrom [17] describe an approach based on a mixed integer programming formulation to optimize EV battery charging behavior with the goal of minimizing charging costs. The design of a simulation environment, which produces charging schedules using a multi-objective evolutionary optimization algorithm is presented by Ramezani et al. [18]. Lee et al. [19] expose an energy consumption scheduler that is able to reduce peak power load in smart places based on genetic algorithms. A concept of real-time scheduling techniques for EV charging that minimizes the impact on the power grid and guarantees the satisfaction of consumer's charging requirements is suggested by Kang et al. [20].

The EV routing problem is less considered in the literature. The energy-optimal routing problem is addressed by Artmeier et al. [21] and it is modeled as the shortest path problem with additional constraints related to the energy consumption. The authors propose a family of search algorithms that satisfy the energy constraints. Erdogan and Miller-Hooks [22] 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 by Lin et al. [23].

In this paper, we address the Electric Vehicle Scheduling and Charging Problem (EVSCP). The EVSCP is defined as follows: Given a set of tours, a fleet of electric and combustion engine vehicles and a charging infrastructure, the objective is to seek an optimal way to assign tours to vehicles and to minimize the EVs charging cost while satisfying several constraints mainly related to the electricity grid, chargers and EVs batteries capacities. To the best of our knowledge, no previous study was devoted to tackle this problem in the literature.

The remainder of the paper is organized as follows: Section 2 presents the problem and describes the notation used in the remaining sections. In Section 3, we prove the NP-Hardness of the problem. Section 4 provides a detailed description of the mathematical formulation of the problem. In Section 5, our exact and heuristic approaches are presented. Section 6 summarizes the computational results. Concluding remarks are given in Section 7.

2 Problem Description and Notation

We consider a set $M_1 = \{1, \ldots, m_1\}$ of Electric Vehicles - EVs and a set $M_2 = \{1, \ldots, m_2\}$ of Combustion Engine Vehicles - CVs that are needed to process a set of n tours during the time horizon $[0, T]$. Each vehicle can process, at most, one tour at a time. Each electric vehicle j operates with a battery characterized by its nominal capacity of embedded energy B_j (kWh) and its State of Charge (SoC_j) defined as the available capacity, and expressed as a percentage of its nominal capacity B_j (0% = empty; 100% = full). Let SoC_j^0 be the initial state of charge of EV j at time $t = 0$. At low and high SoC's values, the battery tends to degrade faster ([24]). In order to improve its lifetime after repeated use and to respect the security issues, at each time, SoC_j should be in the interval $[SoC_j^{\min}, SoC_j^{\max}]$, where $SoC_j^{\min}$ and $SoC_j^{\max}$ are the minimal and the maximal allowable values of SoC, respectively. We assume that there are m_1 chargers available to charge EVs during the time horizon $[0, T]$. This time horizon is divided into T equidistant time periods, $t = 1, \ldots, T$, each of length d, where t represents the time interval $[t-1, t]$ (in our case, $d = 15$ minutes). At each time period t, each charger can provide, to EV j, a charging power $p_{jt} \in [p^{\min}, p^{\max}]$, where $p^{\min}$ and $p^{\max}$ are respectively the minimal and maximal powers that can be delivered by the charger. Thus, an EV charged with a power p_{jt} during the time period t retrieves a total amount of energy equal to $d \times p_{jt}$(kWh). We denote by g_t the electricity grid capacity available for EV charging at time t; i.e., at each time t, the total electricity grid power available to charge all EVs is limited to g_t. Let c_t be the energy cost during time period t. There is a set of n tours to be processed by the vehicles. Each tour i is characterized by a start

time s_i, a finish time f_i, a weight w_i (length expressed in km), and energy E_i (expressed in kWh) required to perform the tour with an EV. Two tours i and j overlap if their intersection is nonempty, i.e., $[s_i, f_i] \cap [s_j, f_j] \neq \emptyset$, otherwise they are disjoint. Given a tour i, let $V(i)$ be the set of tours such that $\forall\, j \in V(i)$, i and j overlap. Note that all tours are already constructed. Thus the start and finish times of each tour are known in advance and, to satisfy operational constraints, preemption of tours is not allowed and the vehicles cannot charge while in tour. The objective consists in maximizing the use of EVs, i.e., the total weight of tours processed with EVs, and minimizing the overall charging costs.

This problem can be seen as a fixed interval scheduling problem ([25, 26]) with additional constraints of energy. The fixed interval scheduling problem without additional constraints is characterized as the problem of scheduling a number of jobs, each with fixed starting and finishing times, and it is stated as follows: Given are n intervals of the form $[s_j, f_j)$ with $s_j < f_j$, for $j = 1, \ldots, n$ and m machines. Each machine can process at most one job at a time and is always available. The objective is to process a maximum number of jobs on the machines. The fixed interval scheduling problem has been largely considered in the literature, particularly the variant where the objective is to maximize the number of (weighted) jobs that can be feasibly scheduled [27, 28, 29, 30]. This variant can be solved using a min-cost flow formulation (see for e.g. [27] and [28]). When jobs have unit weights, a greedy algorithm that finds the maximum number of jobs to process on machines may be used as proposed by [29, 30]. If each job can only be carried out by an arbitrary given subset of machines, the problem becomes NP-hard [27]. However, the fixed interval scheduling problem with energy constraints has, so far, never been considered in the literature.

3 NP-Hardness Result

In this section, we prove the weak NP-hardness of the EVSCP. We use the reduction to the Partition problem, which is known to be NP-complete in the ordinary sense [31]. This problem can be stated as follows: Given a set $P = \{a_1, a_2 \ldots a_p\}$ of p integers such that $\sum_{i=1}^{p} a_i = 2A$, is there a partition of P into two subsets P_1 and P_2 such that $\sum_{i \in P_1} a_i = \sum_{i \in P_2} a_i = A$?

Theorem. 1 *The problem EVRCP is NP-hard in the weak sense.*

Proof. Given an arbitrary instance of the Partition problem, we build an instance ($\mathcal{I}$) of the EVSCP with a set of $2p + 2$ tours. The start times, the finish times and the energy consumptions of those

tours are given in Table 1. We assume that $a_0 = 0$.

Table 1: Characteristics of the instance ($\mathcal{I}$)

Tour	Start time	Finish time	Energy consumption
$T_i; i = 1, \ldots, 2p$	$A + \sum_{i=0}^{i-1} a_k$	$A + \sum_{k=0}^{i} a_k$	a_i
T_{2p+1}	$3A$	$3A + 1$	A
T_{2p+2}	$3A$	$3A + 1$	A

Without loss of generality, we restrict our analysis to the case of two available EVs with a battery capacity of $2A$ (kWh). We assume that there exist two chargers. Each charger can provide one discrete power of 1 (kW). The grid capacity available for EV charging is equal to 2 (kW) at each time interval. At $t = 0$, we assume that the batteries of both EVs are completely depleted. A feasible assignment of tours to EVs should satisfy the following constraints:

- The EVs should have a sufficient energy before processing tours.
- The maximum battery capacity of each EV should never be exceeded during the charging process.
- At each time, the total power used to charge EVs cannot exceed the electricity grid's maximum capacity.

In what follows, we show that there exist a schedule of tours and a feasible charging planning of EVs if, and only if, the Partition problem admits a solution.

First, assume that the Partition problem has a solution and let P_1 and P_2 be the required subsets of P. The desired schedule of tours is constructed by building two sets L_1 and L_2 of tours $T_1, \ldots, T_{2p}$, where tours of L_1 and L_2 correspond to the sets P_1 and P_2, respectively; i.e., $T_i \in L_j$, if $a_i \in P_j$, $i = 1, \ldots, 2p$, $j = 1, 2$. Assign tours of L_1 to EV 1 and tours of L_2 to EV 2. The tours T_{2p+1} and T_{2p+2} are assigned to EV 1 and EV 2, respectively. In order to execute the assigned tours, the EVs are charged as follows: During the time interval $[0, A]$, each EV is charged with a power of 1 (kW); at $t = A$ each EV recovers then A (kWh). During the time interval $[A, 3A + 1]$, EV 1 processes all tours of L_1, for A units of time, and is charged when it is not processing any tour (i.e., EV 1 is charged during the intervals $[A, 3A] \bigcap_{a_i \in P_2} [A + \sum_{k=0}^{i-1} a_k, A + \sum_{i=0}^{i} a_k]$), and, finally, it processes the tour T_{2p+1}, whereas EV 2 processes all tours of L_2 and gets charged during $[A, 3A] \bigcap_{a_i \in P_1} [A + \sum_{k=0}^{i-1} a_k, A + \sum_{k=0}^{i} a_k]$ before performing the tour T_{2p+2}. It is easy to see that, during the different charging phases, each EV recovers enough amount of energy to process all tours. Therefore, the obtained schedule is feasible.

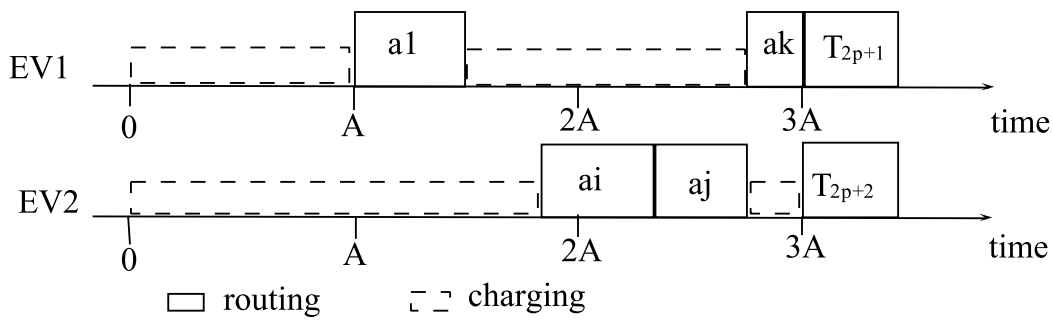

Figure 1: Solution to the scheduling problem on two EVs EV 1 and EV 2.

Conversely, assume now that there exist a feasible schedule of tasks and a charging planning of EVs. Since T_{2p+1} and T_{2p+2} have the same start and finish times, we assume, without loss of generality, that T_{2p+1} and T_{2p+2} are executed by EV 1 and EV 2, respectively. Let R_1 and R_2 be the two sets of tours assigned to EV 1 and EV 2, respectively. Let $\sum_{a_i \in R_1} a_i = A_1$ and $\sum_{a_i \in R_2} a_i = A_2$ be the amounts of energy required by the EVs to execute the sets of tasks A_1 and A_2. Then, EV 1 and EV 2 need a total amount of energy equal to $A + A_1$ and $A + A_2$, respectively. Assume that $A_1 > A$. Then, EV 1 needs an amount of energy greater than $2A$. EV 1 can be charged only during $[0, 3A] \bigcap_{a_i \in R_2} [A + \sum_{k=0}^{i-1} a_k, A + \sum_{k=0}^{i} a_k]$ and it may recover at most a total amount of energy E equal to $3A - A_1 < 2A$ (kWh). Since $E < 2A$, there is not enough energy to process the set of tasks R_1 and T_{2p+1}. Then $A_1 \leq A$. Following the same reasoning, we have $A_2 \leq A$. Since $A_1 + A_2 = 2A$, we conclude that $A_1 = A$ and $A_2 = A$. Therefore, R_1 and R_2 represent a solution to the Partition problem. $\square$

4 Problem Formulation

In this section, we propose a MIP for the EVSCP. We introduce the following decision variables:

x_{ij} : 0-1 variable equals 1 if the vehicle j, $j = 1, \ldots, m_1 + m_2$, is allocated to the tour i, $i = 1, \ldots, n$, and 0 otherwise.

y_{jt} : 0-1 variable equals 1 if the EV j, $j = 1, \ldots, m_1$, is charged during the time interval $t = 1, \ldots, T$ and 0 otherwise.

p_{jt} : real variable denotes the charging power level applied to EV j, $j = 1, \ldots, m_1$ at time interval $t = 1, \ldots, T$.

In order to reduce the complexity of the proposed MIP and, without loss of generality, we assume that the energy E_i required to perform the tour i is consumed during the last period f_i of tour i; i.e, during the time interval $[f_i - 1, f_i]$ ($E_{i,f_i} = E_i$ and $E_{i,l} = 0$, $l = s_i, s_i + 1, \ldots, f_i - 1$). The EVSCP is formulated as a MIP. Its mathematical formulation ($\mathcal{P}$) is the following:

$$\text{Lex} \qquad (\max \sum_{i=1}^{n} \sum_{j=1}^{m_1} w_i \times x_{ij} \ ; \ \min \quad \sum_{j=1}^{m_1} \sum_{t=1}^{T} c_t \times p_{jt}) \tag{1}$$

$$\sum_{j=1}^{m_1+m_2} x_{ij} = 1, \quad \forall i \tag{2}$$

$$x_{ij} + \sum_{i' \in V(i)} x_{i'j} \leq 1, \quad \forall i, \forall j \tag{3}$$

$$\sum_{t=s_i}^{f_i} y_{jt} + (f_i - s_i + 1) \times x_{ij} \leq (f_i - s_i + 1), \quad \forall i, \forall j \in M_1 \tag{4}$$

$$\sum_{j=1}^{m_1} p_{jt} \leq g_t, \quad \forall t \tag{5}$$

$$p^{\min} \times y_{jt} \leq p_{jt}, \quad \forall t, \forall j \in M_1 \tag{6}$$

$$p_{jt} \leq p^{\max} \times y_{jt}, \quad \forall t, \forall j \in M_1 \tag{7}$$

$$\frac{SoC_j^0}{100} + \frac{\sum_{t \leq s_i - 1} d \times p_{jt} - \sum_{l=1/f_l \leq s_i - 1}^{n} E_{l,f_l} \times x_{lj}}{B_j} \leq \frac{SoC_j^{\max}}{100}, \quad \forall i, \forall j \in M_1 \tag{8}$$

$$\frac{SoC_j^0}{100} + \frac{\sum_{t \leq f_i} d \times p_{jt} - \sum_{l=1/f_l \leq f_i} E_{l,f_l} \times x_{lj}}{B_j} \geq \frac{SoC_j^{\min}}{100}, \quad \forall i, \forall j \in M_1 \tag{9}$$

Constraints (2) ensure that each tour is assigned to exactly one vehicle. Constraints (3) guarantee that no vehicle can be assigned to overlapping tours. Constraints (4) prohibit charging the EV when it is in tour. Constraints (5) ensure that, at each time period t, the total power used to charge the EVs does not exceed the electricity grid's maximum capacity. Constraints (6) and (7) guarantee the respect of the minimum and the maximum powers of chargers when charging the EVs. Constraints (8) and (9) ensure that the SoC of each EV is in the interval $[SoC^{\min}, SoC^{\max}]$ during the whole time horizon $[0, T]$.

Our goal is to optimize two lexicographical objective functions ([32], [33]); i.e., the objective functions are arranged in order of their importance and the optimization problems are solved one at a

time. More precisely, if z^* is the optimal value of the primary objective function then the optimization of the secondary objective function is subject to that the primary objective function is bounded by z^*.

In our problem the first objective consists in maximizing the EVs kilometers travelled ($\sum_{i=1}^{n}\sum_{j=1}^{m_1} w_i \times x_{ij}$). The second objective is to minimize the EVs charging costs ($\sum_{j=1}^{m_1}\sum_{t=1}^{T} c_t \times p_{jt}$). The lexicographical optimization of our two objective functions is motivated by the real application in which the most important gain is obtained thanks to the EVs scheduling optimization. Indeed, this gain is about one thousand Euros per day for a fleet of 10 EVs, however, the gain generated thanks to the EVs charging optimization is about thirty euros compared to a non-optimized solution.

5 Solving Approaches

5.1 Exact Method

The proposed MIP is solved in two steps using CPLEX. In the first step, the objective is to maximize the EVs kilometers travelled ($\sum_{i=1}^{n}\sum_{j=1}^{m_1} w_i \times x_{ij}$). The optimal solution generated in the first step serves as the starting solution of the second step. A new constraint which ensures that the number of EVs kilometers travelled is greater than the objective function found in the first step is added to the MIP in the second step. Then, the new MIP is solved with the second objective function ($\sum_{j=1}^{m_1}\sum_{t=1}^{T} c_t \times p_{jt}$). These two steps are described in Algorithm 1.

Algorithm 1 Exact Method

Input: Data of EVSCP
Output: A solution to the EVSCP
Step 1: Solve the MIP with the objective function $K = \sum_{i=1}^{n}\sum_{j=1}^{m_1} w_i \times x_{ij}$
Let K^* be the value of the objective function of the optimal solution S^*
Step 2: Add the constraint $\sum_{i=1}^{n}\sum_{j=1}^{m_1} w_i \times x_{ij} \geq K^*$ to the MIP formulation
Solve the new MIP with the objective function $\sum_{j=1}^{m_1}\sum_{t=1}^{T} c_t \times p_{jt}$ starting from the solution S^*

5.2 Sequential Heuristic Method

In this section, we describe a Sequential Heuristic - SH to solve the EVSCP. This heuristic is mainly based on the following idea: For each EV, a set of tours is selected and assigned to that EV. Then, a charging schedule for this EV is proposed while satisfying all the constraints described in Section 4. This heuristic interleaves two steps: the tours selection step and the charging schedule step. The overall Sequential Heuristic is given in Algorithm 2.

Algorithm 2 Overall Heuristic's Algorithm

Input: Data of the EVSCP
Output: A solution to the EVSCP
for Each EV **do**
    Apply the *Tours Selection Algorithm* to get a set of tours achievable by that EV
    Given the set of selected tours, apply the *Charging Schedule Algorithm* to optimize the charging cost of that EV
end for
for Each CV **do**
    Apply the *Tours Selection Algorithm*
end for

5.2.1 Tours Selection Algorithm

To assign a set of tours to an EV and maximize the kilometers travelled (i.e., maximize the weight of tours assigned to that EV), the Maximum Weight Clique Problem (MWCP) ([35, 36]) is deployed. Recall that the MWCP is defined as follows: Let $\mathcal{G} = (\mathcal{V}; \mathcal{E})$ be an arbitrary undirected and weighted graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. To each node i is associated a weight w_i. Two distinct nodes are said to be adjacent if they are connected by an edge. Given a subset $\mathcal{S}$ of nodes, the weight of $\mathcal{S}$ will be denoted by $W(\mathcal{S}) = \sum_{i \in S} w_i$. A clique of graph $\mathcal{G}$ is a subset of $\mathcal{V}$ in which all nodes are pairwise adjacent. A clique S is called maximal if no strict superset of $\mathcal{S}$ is a clique. A maximal weight clique $\mathcal{S}$ is a clique that is not contained in any other clique having a weight larger than $W(\mathcal{S})$.

The problem of tours selection is represented through an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V}$ represents a tour which is characterized by a weight (the length of tour in kilometers). We assume that the nodes are indexed in a nondecreasing order of their starting times s_i. Two nodes i and j are connected by an edge if the tours i and j are disjoint. The tours selection problem is then equivalent to the MWCP in graph $\mathcal{G}$ with additional constraints. The MWCP is a NP-hard problem [31]. Balas and Yu [38] provide some polynomial solvable cases of MWCP in specific graphs. The interval graph is one of the specific graphs for which the MWCP is a polynomial problem. However, even if graph $\mathcal{G}$ is an interval graph, the MWCP in graph $\mathcal{G}$ should ensure that a feasible charging schedule for EVs exists.

To solve the tours selection problem, we propose a heuristic that finds the maximum weight clique in graph $\mathcal{G}$ and makes sure that the charging schedule exists.

Overview of the algorithm. Our algorithm consists of two steps. In the first step, a maximal weight clique $\mathcal{C}$ is constructed using a greedy heuristic and in the second step, the weight of $\mathcal{C}$ is improved by removing and/or adding one or several nodes. The two steps are repeated till stop conditions are fulfilled. The best weighted clique is updated at each iteration. The details of both steps are as follows:

- **Initial step.** Initially, the clique $\mathcal{C}$ is empty. The algorithm randomly selects a vertex i and adds it to $\mathcal{C}$. To expand this clique, a list of candidate nodes $L = \{v \in \mathcal{V} | v \text{ is connected to all nodes of } \mathcal{C} \text{ and } v \notin \mathcal{C}\}$ is created. The algorithm, then, selects a node v^* from L and adds it to $\mathcal{C}$. v^* is the node with the largest total sum of weights of its adjacent vertices ($W(v^*) = \max_{v \in L} \sum_{k \in V(v)} w_k$, where $V(v)$ is the set of nodes adjacent to v) that respects the *admission condition* (described later) related to the feasibility of a charging schedule. This process is repeated until L is empty or all the nodes of L do not satisfy the *admission condition*.

- **Improvement Step.** When there are no more nodes that could increase the weight of clique $\mathcal{C}$, for each $i \in \mathcal{C}$, the algorithm computes the list of nodes $N_i = \{v \notin \mathcal{C} |\ v \textit{ is connected to all nodes of } \mathcal{C} \setminus \{i\}\}$, and removes from $\mathcal{C}$ the node i with $W(N_i) \geq W(N_j)$, $\forall j \in \mathcal{C}$, where $W(N_i) = \sum_{k \in N_i} w_k$. A new list L of candidate nodes is then selected, and the best node of L is added to $\mathcal{C}$ as described in *Initial step*. The process of adding and/or removing nodes is repeated until a specific timeout is reached. When the improvement step is stopped, the best clique is updated, the current clique is cleared and the algorithm restarts with the initial step until the stop condition is reached. Finally, the algorithm generates the best clique found among all constructed cliques and removes the best clique's nodes from $\mathcal{G}$.

- **Stop Condition.** The algorithm continues until either a fixed limited processing time is reached or the best clique weight reaches the target clique weight; i.e., when a clique that has the same weight as the total weight of the graph, is found.

Admission Condition. The admission condition ensures that the tours corresponding to the current clique's nodes can be processed by an EV; i.e., there is a feasible charging schedule that allows

the EV to perform all tours of the current clique. The existence of a feasible schedule is guaranteed by the metric R. R is the minimal amount of energy that the EV should recover before starting the first tour of the clique, i.e., $R = \sum_{k=1}^{|\mathcal{C}|}(E_k - Y_k)$, where Y_k is the amount of energy that may be charged between successive tours $k-1$ and k. In other words, $Y_k = d \times \sum_{t=f_{k-1}+1}^{s_k-1} \min\{p^{\max}, p'_t\}$, where p'_t is the residual electricity grid power during time period t. We assume that $Y_1 = 0$. A node v is added to the current clique C, if, following the chronological order, for each node $j \in \mathcal{C}$, $R_j = \sum_{k=1}^{j}(E_k - Y_k)$ is lower than the EV's battery capacity and $R_j \leq d \times \sum_{t=0}^{s_1-1} \min\{p^{\text{Max}}, p'_t\}$.

5.2.2 Charging Schedule Algorithm

Given the best clique $\mathcal{C}$ found by the *Tours Selection Algorithm*, the objective here is to provide an optimal charging schedule that minimizes the total charging cost and satisfies the constraints described in Section 4. Below, two optimal charging schedule algorithms are proposed. The first is based on a Minimum Cost Flow Formulation and runs in $O(((T+|\mathcal{C}|)log(T+|\mathcal{C}|))^2)$ time and the second runs in $O((T+|\mathcal{C}|)^2)$ time.

Min-Cost-Flow-Based Charging Algorithm. In this section, we show that the charging problem can be reduced to the Minimum Cost Flow Problem (MCFP) ([34]) in a given network. For a given set V of vertices of the clique $\mathcal{C}$, a network $\mathcal{G} = (\mathcal{W}, \mathcal{A})$ is defined as follows:

- The set of nodes $\mathcal{W}$ consists of (i) a source s, (ii) the nodes t_l representing the time periods $[l-1, l]$, except the time periods when the EV is not available for charging, i.e., $l \in \{1, \ldots, T\} \setminus \cup_{v \in \{1,\ldots,|V|\}}\{s_v + 1, \ldots, f_v\}$, (iii) the nodes v_k, $k = 1, \ldots, |V|$, representing the tours indexed in increasing order of their start times s_k, (iv) a sink p.

- The set $\mathcal{A}$ of directed arcs with restricted capacities consists of (i) arcs (s, t_l) with a maximum capacity $a_l = \min\{p^{max}, p'_l\} \times d$, where p'_l is the residual electricity grid power during time period l, and a cost c_l corresponding to the energy cost during the time period l, (ii) arcs (t_l, v_k) if $l < s_k + 1$ and $l > f_{k-1}$ with a capacity $+\infty$ and a cost equal to zero, (iii) arcs (v_{k-1}, v_k), $k = 2, \ldots |V|$ with a capacity equal to $B_j - E_{k-1}$ and a cost equal to zero, (iv) arcs (v_k, p), $k = 1, \ldots, v_{|V|}$ with a lower bound and an upper bound of the capacity equal to E_k, and a cost equal to zero.

Let $f(i,j)$ be the flow on the arc $(i,j) \in \mathcal{A}$ in an optimal solution to the MCFP with a total cost $\sum_{l \in \{1,\ldots,T\} \setminus \cup_{v \in V}\{s_v+1,\ldots,f_v\}} c_l \times f(s, t_l)$. We define a feasible schedule to the corresponding charging

problem as follows: at each time period l, where $l \in \{1, \ldots, T\} \setminus \cup_{v \in V}\{s_v+1, \ldots, f_v\}$, apply on EV j a charging power $p_{j,l} = \frac{f(s,t_l)}{d}$. Using this procedure, any feasible solution to the MCFP can be converted to a feasible solution S to the charging problem with a cost $C(S) = \sum_{l \in \{1,\ldots,T\} \setminus \cup_{v \in \mathcal{V}}\{s_v+1,\ldots,f_v\}} \mathrm{c}_l \times f(s, t_l)$. Thus, when the flow cost is minimized, the total charging cost is also minimized.

Lemma. 1 *An optimal solution to the charging schedule problem can be found in* $O(((T+|V|)log(T+|V|))^2)$ *polynomial time.*

Proof. The MCFP can be solved in $O(((T+|V|)log(T+|V|))^2)$ time (see for e.g. [37]). Obviously, the optimal solution to the resulting minimum cost flow problem converts into an optimal solution to the charging schedule problem in $O(T)$ time. □

In the following paragraph, a greedy charging schedule algorithm, running in $O(T \times |V|)$ time, is proposed.

Greedy Charging Schedule Algorithm. Let $C = (c_1, \ldots, c_T)$ and $G = (g_1, \ldots, g_T)$ be two vectors of T elements, where c_t and g_t are the electricity cost and the electricity grid capacity during time period $[t-1, t]$, respectively. Let $E = (e_1, \ldots, e_T)$ be the energy vector, where e_t, $t = 1, \ldots, T$, represents either the quantity of energy charged during the time interval $[t-1, t]$ or the energy consumed if the EV is in tour during the time interval $[t-1, t]$. e_t is initialized to $\frac{-E_k}{f_k - s_k + 1}$ if $t \in [s_k, f_k]$, $\forall k \in V$, and to zero otherwise.

The goal of the algorithm is to provide each EV with the maximum possible amount of energy during the cheapest time intervals. At each time interval $[t-1, t]$, e_t is equal to the maximum amount of energy that could be charged during the time interval $[t-1, t]$ if the EV is not in tour. To compute the value of e_t, the residual capacity of EV's battery and the residual available capacity of the electricity grid during the time interval $[t-1, t]$ are taken under consideration. More precisely, the algorithm starts by sorting the vector C in a nondecreasing order of c_t. For each value c_t such that $e_t = 0$; i.e., the time interval $[t-1, t]$ is not already scanned and the EV$_j$ is available for charging, let V_t be the set of tours such that $V_t \subseteq V$ and $v_k \in V_t$ if, and only if, $s_k > t$. For each $v_k \in V_t$ sorted in order of their indices, (i) compute the energy $E_k^{t,B}$ that could be charged without exceeding the capacity of the battery, i.e., $E_k^{t,B} = \frac{SoC^{max} \times B_j}{100} - \sum_{l=1}^{s_k} e_l$, (ii) calculate the energy $E_k^{t,V}$ that should be charged in order to process the tour v_k, i.e., $E_k^{t,V} = \frac{Soc_j^{min} \times B_j}{100} - \sum_{l=1}^{f_k} e_l$. Let $E_k^t = \min\{E_k^{t,B}, E_k^{t,V}\}$. After scanning all tours of V_t, the energy E^t that will be charged during the time interval $[t-1, t]$

is $E^t = \min\{p^{\max} \times d, g_t \times d, \min_{k \in V_t}\{E_k^t\}\}$. Note that we assume that $p^{\min} = 0$. Finally, vectors E and G are updated, i.e., $e_t := E^t$ and $g_t := g_t - \frac{E^t}{d}$. The overall greedy approach is as described by Algorithm 3.

Theorem. 2 *Algorithm 3 provides an optimal solution to the charging schedule problem and runs in $O(T \times |V|)$ polynomial time.*

Proof. Let S be the charging solution generated by the *Greedy Charging Schedule Algorithm*, and let S^* be the optimal solution provided by the *Min-Cost-Flow-Based Charging Algorithm*. Let $C(S)$ and $C(S^*)$ be the costs of solutions S and S^*, respectively. Let t_1 be the first time period such that $p_{j,t_1}(S) \neq p_{j,t_1}(S^*)$ where c_{t_1} is the smallest electricity cost among all electricity costs. By construction, the *Greedy Charging Schedule Algorithm* applies, at each time period, the maximal available power to charge EV j. We have then $p_{j,t_1}(S) > p_{j,t_1}(S^*)$. Let t_2 be another time period such that $p_{j,t_2}(S) < p_{j,t_2}(S^*)$. Let S' be a new solution derived from S^* where $p_{j,t_i}(S^*) = p_{j,t_i}(S')\ \forall i \neq 1, 2$, $p_{j,t_1}(S') = p_{j,t_1}(S') + \epsilon$ and $p_{j,t_2}(S') = p_{j,t_2}(S') - \epsilon$, where $0 < \epsilon \leq p_{j,t_1}(S) - p_{j,t_1}(S^*)$. It is easy to see that $S^{'}$ is a feasible solution to the charging problem, furthermore, $C(S') = C(S^*) - c_{t_2} + c_{t_1} < C(S^*)$ since $c_{t_2} > c_{t_1}$. This contradicts the fact that S^* is an optimal solution. It then follows that, $\forall i$, $p_{j,t_i}(S) = p_{j,t_i}(S^*)$. Therefore, S is an optimal solution to the charging problem. The complexity of the greedy charging algorithm is dominated by the two loops 5 and 7 of Algorithm 3. The greedy algorithm produces an optimal solution in $O(T \times |V|)$-time. $\square$

Algorithm 3 Greedy Charging Schedule Algorithm

Input: A set of tours assigned to EV j

Output: A charging schedule of EV j

Let $C = (c_1, \ldots, c_T)$, $G = (g_1, \ldots, g_T)$ and $E = (e_1, \ldots, e_T)$ be three vectors of T elements, where c_t, $t = 1, \ldots, T$, is the electricity cost during the time interval $[t-1, t]$, g_t, $t = 1, \ldots, T$, is the residual capacity of the electricity grid during the time interval $[t-1, t]$ and e_t, $t = 1, \ldots, T$, is either the quantity of energy charged during the time interval $[t-1, t]$ or the energy consumed during the time interval $[t-1, t]$ if the EV is in routing. Parameter e_t is initialized to $\frac{-E_k}{f_k - s_k + 1}$ if $t \in [s_k, f_k]$, $\forall k \in V$ and to zero otherwise.

Sort the vector C in the nondecreasing order of c_t and let $C = (c_{\pi(1)}, \ldots, c_{\pi(T)})$ be the sorted vector

for each time interval $[\pi(t) - 1, \pi(t)]$ such that $e_{\pi(t)} = 0$ **do**

  Let $V_{\pi(t)}$ be a set of tours such that $V_{\pi(t)} = \{v_k \in V | s_k > \pi(t)\}$ sorted in the increasing order of the indices of tours.

  for each v_k in $V_{\pi(t)}$ **do**

    calculate $E_k^{\pi(t)} = \min\{\frac{SoC^{max} \times B_j}{100} - \sum_{l=1}^{s_k} e_l \; ; \; \frac{SoC_j^{min} \times B_j}{100} - \sum_{l=1}^{f_k} e_l\}$

  end for

  Calculate $E^{\pi(t)} = \min\{p^{\max} \times d, g_{\pi(t)} \times d \; ; \; \min_{k \in V_{\pi(t)}}\{E_k^{\pi(t)}\}\}$

  Update $e_{\pi(t)} = E^{\pi(t)}$ and $g_{\pi(t)} = g_{\pi(t)} - \frac{E^{\pi(t)}}{d}$

end for

The charging schedule is given by the power vector that should be applied to EV j $P = (\frac{\max\{0, e_1\}}{d}, \ldots, \frac{\max\{0, e_T\}}{d})$.

5.3 Global Heuristic Approach

In this section, we propose a new Global Heuristic - GH to solve the EVSCP problem. The GH algorithm consists of two steps. The first step aims at assigning tours to vehicles and guaranteeing the feasibility of a charging schedule. Then, given a feasible assignment of tours to vehicles, the second step determines the optimal charging schedule for all EVs using the MCFP. Both steps are detailed in the next paragraph.

In the first step, the *Tours Selection Algorithm* presented in Section 5.2.1 is used to assign tours to vehicles, and a new *Estimation Procedure* is introduced. In fact, in *Sequential Algorithm*, for each vehicle, the charging schedule is calculated after the tours selection. This allows to update the electricity grid power available at each time. However, in GH algorithm, the charging schedule of all vehicles is calculated in the second step. In order to ensure the existence, in the second step, of a charging schedule of all vehicles, we need to estimate the remaining capacity of the electricity grid after the assignment of a set of tours to each vehicle. Thus, for each vehicle, after the tours selection algorithm, a new *Estimation Procedure* of the available electricity grid power is introduced. Indeed,

the electricity grid power is updated by considering the fact that the current vehicle is charged at the last possible time. More precisely, the *Greedy Charging Schedule Algorithm* presented in Section 5.2.1 is applied to simulate the charging schedule of the EV, and update the electricity grid power available at each time. To ensure that the vehicle is charged at the last possible time, in Algorithm 3, instead of sorting the cost vector C (step 3 of Algorithm 3) in nondecreasing order of c_i, vector C is sorted in the decreasing order of time indices t.

Given the sets of tours assigned to vehicles, the second step of GH Algorithm determines an optimal charging schedule of all vehicles. More precisely, given the sets $L_1, \ldots, L_{m1}$, of tours assigned to EV_1,..., EV_{m1}, respectively, the optimal charging schedule problem is modeled as a Minimum Cost Flow Problem in a network $\mathcal{G} = (\mathcal{W}, \mathcal{A})$ that is defined as follows:

- The set of nodes $\mathcal{W}$ consists of (i) a source s, (ii) the nodes t_l representing the time periods $[l-1, l_i]$, (iii) the nodes $v_k^j \in L_j$, $j = 1, \ldots, m_1$ representing the tours indexed in the increasing order of their start times, and (iv) a sink p.

- The set $\mathcal{A}$ of directed arcs with restricted capacities, consists of (i) arcs (s, t_l) with bounded capacity $a_l = \min\{p^{max}, g_l\} \times d$ and a cost c_l corresponding to the energy cost during the time period l, (ii) arcs (t_l, v_k^j) if $t_l < s_k^j + 1$ and $t_l > f_{k-1}^j$ with a capacity $+\infty$, and a cost equal to zero, (iii) arcs (v_{k-1}^j, v_k^j), $k = 2, \ldots |L_j|$ with a maximum capacity equal to $B_j - E_{k-1}$ and a cost equal to zero, (iv) the arcs (v_k^j, p), $k = 1, \ldots, |L_j|$, $j = 1, \ldots, m_1$, with a lower bound and an upper bound equal to E_k^l, and a cost equal to zero.

Let $f(i, j)$ be the flow on the arc $(i, j) \in \mathcal{A}$ in an optimal solution to the MCFP with a total cost $\sum_{l=1}^{T} c_l \times f(s, t_l)$. We define an optimal charging schedule as follows: at each time interval l, where $l \in \{1, \ldots, T\}$, apply to EV j a charging power $p_{j,l} = \frac{f(t_l, v_k^j)}{d}$, where $(t_l, v_k^j) \in \mathcal{A}$. It is easy to show that the obtained charging schedule is optimal. The proof is similar to that for Theorem 2. Using this procedure, an optimal solution to the MCFP can be converted into an optimal solution S to the charging problem. Thus, when the flow cost is minimized, the total charging cost is also minimized. An optimal solution to the charging schedule problem can be found in $O(((T+|L|)log(T+|L|))^2)$-time, where $L = \cup_{j=1}^{m_1} L_j$.

Algorithm 4 GH Algorithm

Input: Data of the EVSCP
Output: A solution to the EVSCP
for each EV **do**
  Apply the *Tours Selection Algorithm* described in Section 5.2
  Update the electricity power grid capacity by applying the *Estimation Procedure*; i.e., sort the cost vector C in the decreasing order of time indices t and apply Algorithm 3 as described above
end for
Apply the global *Min-Cost-Flow-Based Charging Algorithm* to determine an optimal charging schedule to all EVs
for each CV **do**
  Apply the *Tours Selection Algorithm*
end for

6 Computational Experiments and Discussion

To evaluate the quality of solutions obtained through the exact method and the two heuristics, extensive numerical experiments are conducted on real instances and randomly generated instances. To be able to asses the solution quality, we use real instances provided by the Group La Poste and other randomly generated instances. All the experiments are carried out on an Intel Xeon E5620 2.4GHz processor, with 8GB RAM memory. The tested algorithms are coded in the C++ language. The commercial solver ILOG CPLEX 12.5 is used to solve the MIP. The results produced by the algorithms are compared in terms of quality of the generated solutions and run times. The optimization procedure is based on a $24h$ time horizon. Prices of electricity are based on those proposed by EDF (French electric utility Company). Note that at most m_1 Level 1 chargers, with a range of $0.0(\text{kW}) - 3.7(\text{kW})$, could be used.

The first real data instance is composed of 18 CVs and 8 EVs. All EVs considered have 22(kWh) battery packs. Forty-five tours have to be assigned to those vehicles. The second data instance is composed of 14 CVs, 8 EVs with a battery capacity of 22(kWh) and 4 other EVs with a battery capacity of 23.5(kWh). Forty-six tours have to be assigned to those vehicles. For these two instances, CPLEX and the heuristics succeeded to generate optimal solutions within short time periods (less than three seconds). For confidentiality reasons, we are not allowed to present the details of the computational results related to those instances.

Random instances are generated as follows. The total number of vehicles $nv = m_1 + m_2$ takes its values from the set $\{40, 80, 120, 160, 200\}$. The number of EVs m_1 is a ratio of nv; i.e., $m_1 = ev \times nv$. ev takes its values from the set $\{0.25, 0.5, 0.75, 1.0\}$. A parameter a is used to distinguish the capacity

of EVs batteries, namely, $a = 1$ means that all EVs have a capacity of 22(kWh) and $a = 2$ means that 50% of EVs have a capacity equal to 22(kWh) and the rest of EVs have a capacity of 16(kWh). At most, m_1 chargers, providing a power ranging between 0(kW) and 3.7(kW), could be used to charge EVs. The parameter tt is related to the number of tours. If $tt = 1$, then $n = nv \times c$, where $c \in [1.2, 1.3]$. If $tt = 2$, then $n = nv \times c$, where $c \in [1.5, 1.6]$. All tours start no earlier than $6am$ and finish no later than $8pm$, and their energy need is between 0.15(kWh) and 0.35(kWh) per kilometer. 50% of those tours are long; i.e., they last between 5 and 7 hours, start between $6am$ and $1pm$, and their lengths lie within $[50(km), 80(km)]$. The rest of tours are considered medium; i.e., they last between 2 and 4 hours, and their lengths are in $[20(km), 45(km)]$. A fraction of $\frac{1}{6}$ of medium tours start between $6am$ and $8am$, $\frac{1}{3}$ start between $8am$ and $10am$, $\frac{1}{6}$ finish between $6pm$ and $8pm$, and $\frac{1}{3}$ finish between $4pm$ and $6pm$. Regarding the electricity grid capacity available for EVs charging, we assumed that, if $t \in$[00am, 6am], $g_t = \frac{2}{3} \times ev \times nv \times p^{\text{Max}}$, if $t \in$[6am, 6pm], $g_t = \frac{1}{5} \times ev \times nv \times p^{\text{Max}}$ and if $t \in$[6pm, 00am], $g_t = \frac{1}{2} \times ev \times nv \times p^{\text{Max}}$, where $p^{\text{Max}} = 3.7$(kW).

For each value of nv, sixteen classes of instances are then constructed; and for each class, ten instances are randomly generated. Each class is denoted by $nv_w_ev_x_a_y_tt_z$, where $w = nv$, x takes the values 1, 2, 3, 4 if ev is equal to : 0.25, 0.50, 0.75, 1, respectively, $y = a = \{1, 2\}$ and $z = tt = \{1, 2\}$. The computational results produced by CPLEX, SH and GH Algorithms are summarized in Table 2, and a time limit of 3600 seconds is set for each instance. Note that, here, we only compare the first objective function values, i.e. $\sum_{i=1}^{n} \sum_{j=1}^{m_1} w_i \times x_{ij}$, since the charging costs is correlated with the EVs traveled kilometers.

In Table 2 the following indicators are reported: (i) the *Success*; i.e., the percentage of solved instances among the tested instances within the same class. Solved instances refer to instances for which at least one solution is generated, within the time limit, even if it is not optimal, (ii) the Gap_{UB} of the best solution, denoted by S, produced by either CPLEX or heuristics SH and GH in relation to the Upper Bound (UB) generated by CPLEX, i.e., $\text{Gap}_{\text{UB}}(S) = \frac{UB-S}{S}$, (iii) the Gap_{max} computed as $Gap_{max}(s) = \frac{\max(S_{\text{EM}}, S_{\text{SH}}, S_{\text{GH}}) - S}{S}$, where S_{EM}, S_{SH}, and S_{GH} are the values of the objective function of the solutions generated by CPLEX, SH Algorithm and GH algorithm, respectively, and, (iv) the average run time (s) needed to generate an assignment and a charging solution to the EVSCP.

Table 2: Performance results of CPLEX, SH and GH heuristics on randomly generated instances. Entries '*' mean that the run-time limit was reached without finding any feasible solution.

	Exact method				**SH Algorithm**				**GH Algorithm**			
Instance	Success	Gap_{UB}	Gap_{max}	Time(s)	Success	Gap_{UB}	Gap_{max}	Time(s)	Success	Gap_{UB}	Gap_{max}	Time(s)
nv_40_ce_1_a_1_tt_1	90.00%	5.27%	0.00%	3600.00	100.00%	6.72%	1.36%	0.07	100.00%	5.94%	0.62%	0.29
nv_40_ce_1_a_1_tt_2	90.00%	10.62%	1.02%	3602.00	100.00%	10.88%	1.29%	0.09	100.00%	9.46%	0.00%	0.26
nv_40_ce_1_a_2_tt_1	90.00%	10.29%	2.95%	3601.98	100.00%	7.36%	0.29%	0.07	100.00%	7.14%	0.00%	0.23
nv_40_ce_1_a_2_tt_2	100.00%	14.85%	3.68%	3608.23	100.00%	11.38%	0.61%	0.09	100.00%	10.71%	0.00%	0.27
nv_40_ce_2_a_1_tt_1	90.00%	8.54%	0.00%	3602.18	100.00%	13.70%	4.76%	0.09	100.00%	9.84%	1.21%	6.39
nv_40_ce_2_a_1_tt_2	100.00%	11.96%	0.00%	3601.00	100.00%	16.57%	4.17%	0.12	100.00%	12.61%	0.63%	6.57
nv_40_ce_2_a_2_tt_1	90.00%	13.39%	3.32%	3601.00	100.00%	12.75%	2.76%	0.09	100.00%	9.73%	0.00%	5.86
nv_40_ce_2_a_2_tt_2	100.00%	17.97%	4.37%	3601.00	100.00%	16.43%	3.13%	0.14	100.00%	12.89%	0.00%	7.36
nv_40_ce_3_a_1_tt_1	90.00%	0.49%	0.00%	3601.00	100.00%	0.50%	0.00%	0.11	100.00%	0.50%	0.00%	0.59
nv_40_ce_3_a_1_tt_2	100.00%	7.19%	0.00%	3602.00	100.00%	8.46%	1.18%	0.16	100.00%	8.23%	0.96%	0.93
nv_40_ce_3_a_2_tt_1	90.00%	1.40%	0.90%	3606.47	100.00%	0.46%	0.46%	0.12	100.00%	0.00%	0.00%	0.61
nv_40_ce_3_a_2_tt_2	100.00%	12.57%	4.58%	3601.00	100.00%	8.12%	0.49%	17.49%	100.00%	7.59%	0.00%	1.26
nv_40_ce_4_a_1_tt_1	90.00%	0.00%	0.00%	13.47	100.00%	0.46%	0.46%	0.12	100.00%	0.00%	0.00%	0.60
nv_40_ce_4_a_1_tt_2	100.00%	0.00%	0.00%	29.96	100.00%	0.00%	0.00%	0.22	100.00%	0.00%	0.00%	1.19
nv_40_ce_4_a_2_tt_1	90.00%	0.00%	0.00%	19.14	100.00%	0.00%	0.00%	0.13	100.00%	0.00%	0.00%	0.60
nv_40_ce_4_a_2_tt_2	100.00%	0.00%	0.00%	52.64	100.00%	0.00%	0.00%	0.19	100.00%	0.00%	0.00%	1.32
nv_80_ce_1_a_1_tt_1	100.00%	18.65%	1.80%	3601.00	100.00%	19.75%	2.77%	0.33	100.00%	16.53%	0.00%	14.19
nv_80_ce_1_a_1_tt_2	100.00%	26.14%	5.57%	3601.00	100.00%	23.19%	3.15%	0.51	100.00%	19.43%	0.00%	12.37
nv_80_ce_1_a_2_tt_1	100.00%	24.43%	7.00%	3601.00	100.00%	20.05%	3.25%	0.29	100.00%	16.27%	0.00%	10.84
nv_80_ce_1_a_2_tt_2	100.00%	32.55%	10.40%	3601.00	100.00%	22.83%	2.57%	0.41	100.00%	19.75%	0.00%	12.33
nv_80_ce_2_a_1_tt_1	100.00%	20.47%	5.57%	3601.20	100.00%	14.12%	0.31%	0.39	100.00%	13.77%	0.00%	3.44
nv_80_ce_2_a_1_tt_2	100.00%	28.40%	11.23%	3601.48	100.00%	14.58%	0.14%	0.55	100.00%	14.42%	0.00%	4.18
nv_80_ce_2_a_2_tt_1	100.00%	35.45%	18.26%	3601.85	100.00%	14.30%	0.47%	0.35	100.00%	13.76%	0.00%	3.15
nv_80_ce_2_a_2_tt_2	100.00%	52.92%	32.61%	3602.36	100.00%	14.76%	0.32%	0.49	100.00%	14.40%	0.00%	3.92
nv_80_ce_3_a_1_tt_1	100.00%	1.18%	0.00%	818.78	100.00%	1.46%	0.27%	0.39	100.00%	1.35%	0.17%	4.88
nv_80_ce_3_a_1_tt_2	100.00%	15.49%	4.89%	3601.98	100.00%	10.05%	0.49%	0.63	100.00%	9.51%	0.00%	5.98
nv_80_ce_3_a_2_tt_1	100.00%	7.00%	5.47%	3606.53	100.00%	1.47%	0.09%	0.41	100.00%	1.38%	0.00%	4.20
nv_80_ce_3_a_2_tt_2	60.00%	25.87%	15.54%	3602.01	100.00%	8.61%	0.00%	0.60	100.00%	8.72%	0.10%	5.24

	Exact method				SH Algorithm				GH Algorithm			
Instance	Success	Gap_{UB}	Gap_{max}	Time(s)	Success	Gap_{UB}	Gap_{max}	Time(s)	Success	Gap_{UB}	Gap_{max}	Time(s)
nv_80_ce_4_a_1_tt_1	100.00%	0.00%	0.00%	117.31	90.00%	0.12%	0.12%	0.47	100.00%	0.00%	0.00%	4.93
nv_80_ce_4_a_1_tt_2	90.00%	0.00%	0.00%	798.39	90.00%	0.12%	0.12%	0.56	100.00%	0.00%	0.00%	6.34
nv_80_ce_4_a_2_tt_1	100.00%	0.00%	0.00%	217.83	100.00%	0.00%	0.00%	0.42	100.00%	0.00%	0.00%	5.01
nv_80_ce_4_a_2_tt_2	30.00%	0.00%	0.00%	1062.29	90.00%	0.00%	0.00%	0.55	100.00%	0.00%	0.00%	4.29
nv_120_ce_1_a_1_tt_1	100.00%	24.62%	6.93%	3601.74	100.00%	17.05%	0.47%	0.64	100.00%	16.51%	0.00%	3.54
nv_120_ce_1_a_1_tt_2	100.00%	48.84%	22.46%	3601.21	100.00%	20.88%	0.00%	1.02	100.00%	20.88%	0.00%	3.93
nv_120_ce_1_a_2_tt_1	100.00%	37.80%	17.91%	3601.01	100.00%	17.01%	0.39%	0.61	100.00%	16.56%	0.00%	3.14
nv_120_ce_1_a_2_tt_2	100.00%	73.91%	40.38%	3601.76	100.00%	21.77%	0.41%	1.08	100.00%	21.27%	0.00%	4.15
nv_120_ce_2_a_1_tt_1	100.00%	103.17%	63.42%	3602.63	100.00%	15.66%	0.83%	0.76	100.00%	14.72%	0.00%	8.35
nv_120_ce_2_a_1_tt_2	50.00%	111.66%	78.48%	3602.42	100.00%	19.50%	1.46%	1.33	100.00%	17.78%	0.00%	18.65
nv_120_ce_2_a_2_tt_1	100.00%	115.33%	80.87%	3603.01	100.00%	15.41%	0.73%	0.83	100.00%	14.57%	0.00%	8.66
nv_120_ce_2_a_2_tt_2	20.00%	95.76%	66.32%	3602.42	100.00%	18.57%	1.20%	1.23	100.00%	17.16%	0.00%	17.01
nv_120_ce_3_a_1_tt_1	100.00%	31.21%	29.70%	3603.29	100.00%	1.48%	0.71%	0.85	100.00%	0.77%	0.00%	15.15
nv_120_ce_3_a_1_tt_2	0.00%	*	*	*	100.00%	10.45%	1.88%	1.48	100.00%	8.41%	0.00%	15.86
nv_120_ce_3_a_2_tt_1	90.00%	29.44%	28.09%	3603.97	100.00%	1.19%	0.58%	0.89	100.00%	0.61%	0.00%	15.10
nv_120_ce_3_a_2_tt_2	0.00%	*	*	*	100.00%	10.86%	2.26%	1.72	100.00%	8.41%	0.00%	13.23
nv_120_ce_4_a_1_tt_1	90.00%	0.00%	0.00%	504.42	100.00%	0.00%	0.00%	0.79	100.00%	0.00%	0.00%	12.05
nv_120_ce_4_a_1_tt_2	0.00%	*	*	*	100.00%	2.10%	2.10%	1.61	100.00%	0.00%	0.00%	14.22
nv_120_ce_4_a_2_tt_1	30.00%	0.00%	0.00%	730.19		0.33%	0.33%	0.80	100.00%	0.00%	0.00%	8.13
nv_120_ce_4_a_2_tt_2	0.00%	*	*	*	0.00%	*	*	*	100.00%	0.13%	0.00%	8.58
nv_160_ce_1_a_1_tt_1	100.00%	374.04%	146.35%	3603.88	100.00%	20.50%	0.56%	1.37	100.00%	19.84%	0.00%	5.87
nv_160_ce_1_a_1_tt_2	50.00%	223.20%	155.19%	3603.36	100.00%	25.20%	0.12%	1.99	100.00%	25.06%	0.00%	8.22
nv_160_ce_1_a_2_tt_1	90.00%	882.71%	450.83%	3602.62	100.00%	19.36%	0.00%	1.35	100.00%	19.36%	0.00%	1.65
nv_160_ce_1_a_2_tt_2	50.00%	150.93%	100.99%	3604.58	100.00%	23.86%	0.24%	2.35	100.00%	23.56%	0.00%	6.68
nv_160_ce_2_a_1_tt_1	10.00%	294.76%	240.23%	3606.24	100.00%	16.82%	0.68%	1.41	100.00%	16.03%	0.00%	63.09
nv_160_ce_2_a_1_tt_2	0.00%	*	*	*	100.00%	19.70%	0.89%	2.49	100.00%	18.90%	0.00%	132.13
nv_160_ce_2_a_2_tt_1	40.00%	223.84%	181.29%	3603.72	100.00%	13.97%	0.67%	1.51	100.00%	13.22%	0.00%	40.83
nv_160_ce_2_a_2_tt_2	0.00%	*	*	*	100.00%	19.31%	0.83%	2.45	100.00%	18.33%	0.00%	113.56
nv_160_ce_3_a_1_tt_1	0.00%	*	*	*	100.00%	6.16%	4.39%	1.57	100.00%	1.69%	0.00%	168.93
nv_160_ce_3_a_1_tt_2	0.00%	*	*	*	100.00%	19.14%	6.68%	2.84	100.00%	11.69%	0.00%	152.09
nv_160_ce_3_a_2_tt_1	0.00%	*	*	*	100.00%	6.27%	4.33%	1.57	100.00%	1.86%	0.00%	159.00
nv_160_ce_3_a_2_tt_2	0.00%	*	*	*	100.00%	19.20%	6.80%	2.75	100.00%	11.61%	0.00%	129.82
nv_160_ce_4_a_1_tt_1	0.00%	*	*	*	90.00%	0.06%	0.95%	1.60	100.00%	0.00%	0.00%	26.24

	Exact method				SH Algorithm				GH Algorithm			
Instance	Success	Gap_{UB}	Gap_{max}	Time(s)	Success	Gap_{UB}	Gap_{max}	Time(s)	Success	Gap_{UB}	Gap_{max}	Time(s)
nv_160_ce_4_a_1_tt_2	0.00%	*	*	*	30.00%	1.00%	0.94%	2.59	100.00%	0.05%	0.00%	137.98
nv_160_ce_4_a_2_tt_1	0.00%	*	*	*	100.00%	0.00%	0.00%	1.51	100.00%	0.00%	0.00%	29.09
nv_160_ce_4_a_2_tt_2	0.00%	*	*	*	20.00%	1.54%	1.54%	2.69	100.00%	0.00%	0.00%	121.15
nv_200_ce_1_a_1_tt_1	60.00%	390.39%	390.39%	3607.04	100.00%	21.75%	0.43%	2.11	100.00%	21.23%	0.00%	16.83
nv_200_ce_1_a_1_tt_2	0.00%	*	*	*	100.00%	22.28%	0.32%	4.32	100.00%	22.06%	0.00%	15.26
nv_200_ce_1_a_2_tt_1	20.00%	558.04%	447.35%	3600.00	100.00%	20.92%	0.80%	2.16	100.00%	20.22%	0.00%	15.11
nv_200_ce_1_a_2_tt_2	0.00%	*	*	*	100.00%	22.56%	0.80%	4.79	100.00%	21.58%	0.00%	16.73
nv_200_ce_2_a_1_tt_1	0.00%	*	*	*	100.00%	22.42%	4.01%	3.06	100.00%	17.70%	0.00%	184.33
nv_200_ce_2_a_1_tt_2	0.00%	*	*	*	100.00%	26.08%	3.79%	6.40	100.00%	21.48%	0.00%	109.74
nv_200_ce_2_a_2_tt_1	0.00%	*	*	*	100.00%	22.39%	4.06%	3.03	100.00%	17.62%	0.00%	185.07
nv_200_ce_2_a_2_tt_2	0.00%	*	*	*	100.00%	29.07%	4.07%	6.17	100.00%	24.03%	0.00%	127.91
nv_200_ce_3_a_1_tt_1	0.00%	*	*	*	100.00%	13.01%	6.66%	2.96	100.00%	5.95%	0.00%	164.35
nv_200_ce_3_a_1_tt_2	0.00%	*	*	*	30.00%	27.48%	7.13%	6.79	100.00%	18.99%	0.00%	173.80
nv_200_ce_3_a_2_tt_1	0.00%	*	*	*	100.00%	12.59%	6.55%	2.88	100.00%	5.67%	0.00%	122.31
nv_200_ce_3_a_2_tt_2	0.00%	*	*	*	30.00%	28.17%	7.24%	7.09	100.00%	19.52%	0.00%	188.96
nv_200_ce_4_a_1_tt_1	0.00%	*	*	*	100.00%	0.00%	0.00%	2.60	100.00%	0.00%	0.00%	183.20
nv_200_ce_4_a_1_tt_2	0.00%	*	*	*	70.00%	0.29%	0.29%	5.22	100.00%	0.00%	0.00%	109.60
nv_200_ce_4_a_2_tt_1	0.00%	*	*	*	100.00%	0.00%	0.00%	13.27	100.00%	0.00%	0.00%	110.67
nv_200_ce_4_a_2_tt_2	0.00%	*	*	*	70.00%	0.25%	0.25%	6.05	100.00%	0.00%	0.00%	156.08

From table 2, we can make the following observations:

- The exact algorithm solves small and medium instances (fleets with 40 and 80 vehicles), but fails to solve large instances (fleets with 120, 160 and 200 vehicles). For large instances with a small number of EVs, i.e. $ce = 0.25$, the exact algorithm succeeds to find feasible solutions. Furthermore, when the EVs fleet is heterogeneous, i.e. $a = 2$, the exact algorithm performs poorly in finding feasible solutions compared to the same classes of instances with homogenous fleets of EVs.

- GH heuristic succeeds in solving all instances, whereas SH heuristic fails to solve 44 instances out of 800. This particularly concerns large instances (for example, instances of classes $nv_200_ce_3_a_2_tt_1$ and $nv_200_ce_4_a_1_tt_2$). This is due to the fact that SH algorithm computes a charging schedule for each EV sequentially without anticipating the existence of a feasible charging schedule for the last EVs.

- Concerning the performance of the proposed heuristics, Table 2 shows that both heuristics exhibit an excellent performance. GH algorithm solves to optimality 180 instances out of 800 instances, and provides the best performances with GAP_{max} less than 1% comparing to the best solutions found by the exact and the SH heuristic and a gap GAP_{UB} of at most 20% compared to the upper bound found by CPLEX. SH heuristic succeeds in finding optimal solutions for 89 instances and provides good solutions with a GAP_{max} of at most 7% and a GAP_{UB} of at most 30% compared to the upper bound found by CPLEX for large instances.

- The running times of the SH and GH heuristics are relatively short. About 190 seconds are needed to solve large instances with up to 200 vehicles using the GH heuristic. A shorter running time, with at most 13 seconds, is needed to find good solutions for large instances using SH heuristic.

7 Conclusion

In this paper, we considered the problem of simultaneously optimizing the allocation of tours to EVs and minimizing the charging costs. This new problem was shown to be NP-hard in the ordinary sense. To solve the EVSCP, a mixed integer linear programming formulation was proposed. An exact method based on two phases was developed to solve the MIP with CPLEX. We also proposed two heuristics. The first heuristic is a sequential one and interleaves two sub-algorithms to assign a set of tours to

each EV using the Maximum Weight Clique Problem, and plan its charging process using a Minimum Cost Flow formulation. The second is a global heuristic. It generates a solution to the assignment problem of tours to all EVs while guaranteeing the feasibility of a charging schedule. After that, it applies a global charging schedule algorithm to minimize charging costs. The computational results showed the efficiency and the effectiveness of the proposed heuristics.

References

[1] Tomic, J., Kempton, W.: Using fleets of electric-drive vehicles for grid support. Journal of Power Sources 2007; 168: 459–468

[2] Kempton, W., Letendre, S.E.: Electric vehicles as a new power source for electric utilities. Transportation Research D 1997; 2: 157–175

[3] Kempton, W., Tomic, J.: Vehicle-to-grid power implementation: from stabilizing the grid to supporting large-scale renewable energy. Journal of Power Sources 2005; 144: 280–294

[4] Kempton, W., Tomic, J.: Vehicle-to-grid power fundamentals: calculating capacity and net revenue. Journal of Power Sources 2005; 144: 268–279

[5] Hutson, C., Venayagamoorthy, G.K., Corzine, K.A.: Intelligent scheduling of hybrid and electric vehicle storage capacity in a parking lot for profit maximization in grid power transactions. In: IEEE Energy 2030 (2008)

[6] Shi, W., Wong, V.W.S.: Real-time vehicle-to-grid control algorithm under price uncertainty. In: SmartGridComm (2011)

[7] Lopez, M.A., Martin, S., Aguado, J.A., De la Torre, S.: V2G strategies for congestion management in microgrids with high penetration of electric vehicles. Electric Power Systems Research 2013; 104: 28–34

[8] Jian, J., Zhu, X., Shao, Z., Niu, S., Chan, C.C.: A scenario of vehicle-to-grid implementation and its double-layer optimal charging strategy for minimizing load variance within regional smart grids. Energy Conversion and Management 2014; 78: 508-517

[9] Soares, J., Sousa, T., Morais, H., Vale, Z., Canizes, B., Silva, A.: Application-Specific Modified Particle Swarm Optimization for energy resource scheduling considering vehicle-to-grid. Applied Soft Computing 2013; 13: 4264-4280

[10] Wang, Y.W.: An optimal location choice model for recreation-oriented scooter recharge stations. Transportation Research Part D: Transport and Environment 2007; 12(3): 231–237

[11] Wang, Y.W.: Locating battery exchange stations to serve tourism transport. Transportation Research Part D: Transport and Environment 2008; 13(3): 193–197

[12] Wang, Y.W., Lin, C.C.: Locating multiple types of recharging stations for battery-powered electric vehicle transport. Transportation Research Part E 2013; 58: 76-87

[13] He, F., Wu, D., Yin, Y., Guan, Y.: Optimal deployment of public charging stations for plug-in hybrid electric vehicles. Transportation Research Part B 2013; 47: 87-101

[14] Rotering, N. and Ilic, M.: Optimal charge control of plug-in hybrid electric vehicles in deregulated electricity markets. IEEE Transactions on Power Systems 2011; 26(3): 1021–1029

[15] Erol-Kantarci, M. and Mouftah, H. T.: Prediction-based charging of PHEVs from the smart grid with dynamic pricing. In: Proceedings of the IEEE Workshop on Smart Grid Networking Infrastructure, Denver, CO; 2010, p. 1032–1039

[16] Deilami, S., Masoum, A. S., Moses, P. S. and Masoum, M. A. S.: Realtime coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile. IEEE Transactions on Smart Grid 2011; 2(3): 456–467

[17] Sundstrom, O.: Binding Optimization Methods to Plan the Charging of Electric Vehicle Fleets. In: International Conference on Control, Communication and Power Engineering, 28–29 (2010)

[18] Ramezani, M., Graf, M., Vogt, H.: A simulation environment for smart charging of electric vehicles using a multi-objective evolutionary algorithm. In: Kranzlmuller, D., Tjoa, M. (eds.) ICT-Glow 2011; LNCS vol. 6868, p. 56–63

[19] Lee, J., Park, G-L., Kwak, H-Y., Jeon, H.: Design of an energy consumption scheduler based on genetic algorithms in the smart grid. In: Part I, ICCCI 2011; LNCS vol. 6922, p. 438–447

[20] Kang, J., Duncan, S.J., Mavris, D.N.: Real-time Scheduling Techniques for Electric Vehicle Charging in Support of Frequency Regulation. Procedia Computer Science 2013; 16: 767-775

[21] Artmeier, A., Haselmayr, J., Leucher, M., Sachenbacher, M.: 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)

[22] Erdogan, S., Miller-Hooks, E.: A Green Vehicle Routing Problem. Transportation Research Part E 2012; 48: 100-114

[23] Lin, C., Choy, K.L., Ho, G.T.S., Chung, S.H., Lam, H.Y.: Survey of Green Vehicle Routing Problem: Past and future trends. Expert Systems with Applications 2014; 41: 1118-1138

[24] Bashash, S., Moura, S.J., Forman, J.C., Fathy, H. K.: Plug-in hybrid electric vehicle charge pattern optimization for energy and battery longevity. Journal of power sources 2011; 196:541–549

[25] Kolen, W.J.A., Lenstra, J.K., Papadimitriou, C. H., Spieksma, F.C.R.: Interval Scheduling: A Survey. Naval research and Logistics 2007; 54: 530-543

[26] Kovalyov, M.Y., Ng, C.T., Cheng, T.C.E.: Fixed interval scheduling: Models, applications, computational complexity and algorithms. European Journal of Operational Research 2007; 178: 331–342

[27] Arkin, E.M., Silverberg, E.B.: Scheduling with fixed start and end times. Discrete Applied Mathematics 1987; 18: 1–8

[28] Bouzina, K.I., Emmons, H.: Interval scheduling on identical machines. Journal of Global Optimization 1996; 9: 379–393

[29] Faigle, U. and Nawijn, W.M.: Note on scheduling intervals on-line, Discrete Appl Math 1995; 58: 13–17

[30] Carlisle, M.C. and Lloyd, E.L.: On the k-coloring of intervals, Discrete Appl Math 1955; 59: 225–235

[31] Garey, R., Johnson, S.: Computers and Intractability: A Guide to the Theory of NP-Completeness, A Series of Books in the Mathematical Sciences. W.H. Freeman (1979)

[32] Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering, Structural and Multidisciplinary Optimization 2004; 26 (6):369-395.

[33] M. Ehrgott, Multicriteria Optimization. Springer, Berlin, 2000.

[34] Vaidyanathan, B., Ahuja, R. K.: Fast Algorithms for Specially Structured Minimum Cost Flow Problems with Applications. Operations Research 2010; 58 (6): 1681–1696.

[35] Brijnesh, J.J., Wysotski, F.: The Maximum Weighted Clique Problem and Hopfield Network. In: Proceedings - European Symposium on Artificial Neural Networks (ESANN) Bruges (Belgium), p. 331–336 (2004)

[36] Yamaguchi, K., Masuda, S.: A New Exact Algorithm for the Maximum Weight Clique. In: the 23rd International Technical Conference on Circuits/Systems, Computers and Communications (2008)

[37] Orlin, J. B.: A Faster Strongly Polynomial Minimum Cost Flow Algorithm. Operations Research 1993; 41(2): 338–350

[38] Balas, E., Yu, C.S.: On graphs with polynomially solvable maximum-weight clique problem. Networks 1989; 19: 247–253