



# Hybrid Heuristic and Metaheuristic for Solving Electric Vehicle Charging Scheduling Problem

Imene Zaidi<sup>1</sup>, Ammar Oulamara<sup>2</sup>, Lhassane Idoumghar<sup>1</sup>,  
and Michel Basset<sup>1</sup>

<sup>1</sup> Université de Haute -Alsace, IRIMAS UR 7499, 68100 Mulhouse, France  
{imene.zaidi, lhassane.idoumghar, michel.basset}@uha.fr

<sup>2</sup> Université de Lorraine, LORIA Laboratory UMR7503,  
54506 Vandoeuvre-lès-Nancy, France  
ammar.oulamara@loria.fr

**Abstract.** The electric vehicle (EV) charging scheduling problem has become a research focus to mitigate the impact of large-scale deployment of EV in the near future. One of the main assumptions in literature is that there are enough charging points (CP) in the charging station to meet all charging demands. However, with the deployment of EVs, this assumption is no longer valid. In this paper, we address the electric vehicle charging problem in a charging station with a limited number of heterogeneous CPs and a limited overall power capacity. Before arriving at the station, the EV drivers submit charging demands. Then, the scheduler reserves a suitable CP for each EV and allocates the power efficiently so that the final state-of-charge at the departure time is as close as possible to the requested state-of-charge. We present two variants of the problem: a constant output power model and a variable power model. To solve these problems, heuristic and simulated annealing (SA) combined with linear programming are proposed. Simulation results indicate that the proposed approaches are effective in terms of maximizing the state-of-charge by the departure time for each EV.

**Keywords:** Electric vehicle · Charging scheduling · Optimization · Heuristic · Simulated annealing

## 1 Introduction

The adoption of EVs has been growing rapidly over the past decade, mainly as a result of ambitious government policies to reduce environmental pollution and advances in the EV industry. In 2019, worldwide EV sales reached 2.1 million, bringing the global EV fleet to 7.2 million, an increase of 40% compared to 2018 [6]. However, the future large-scale adoption of EV raises concerns about charging service quality since charging an EV is time-consuming and requires a considerable amount of electrical power. Nowadays, EV drivers tend to choose

the nearest available CPs to plug in their EVs and start charging immediately. This can easily overload charging infrastructures in a large-scale scenario resulting in poor service quality and EV drivers' dissatisfaction.

Recently, there has been growing interest in the development of EV charging scheduling strategies that focus on economical objectives such as minimizing the electricity costs or on power grid reliability by minimizing the power losses [3] and voltage deviations [7]. However, many studies assume that there is a sufficient number of CPs thus focus on the power allocation and neglect the assignment of EV to a suitable CP. Besides, they assume an identical output power of CPs. Yet, in real life, CPs with different charging output power are installed in the same charging station to meet the various type of charging demands and improve the quality of service [12].

In this work, we consider a charging station that regroups several CPs with different charging power levels. Moreover, this charging station has a maximum power capacity that the distribution-level transformer poses. Each EV driver can submit a charging reservation before arriving to avoid queuing. We developed a heuristic based on interval scheduling [9] and a Simulated Annealing (SA) to assign each EV to a suitable CP and allocate the electrical power over the plugin time. The objective of the scheduling is to ensure a final state-of-charge at the departure that is as close as possible to the requested state-of-charge.

The remainder of this paper is organized as follows. In Sect. 2, we present a brief review of the main works on electric vehicle charging scheduling problems. In Sect. 3, we describe in detail the investigated problem and formulate it as a mixed-integer linear programming (MILP). We then propose our optimization methods in Sect. 4 and evaluate their performance in Sect. 5. Finally, we conclude the paper in Sect. 6.

## 2 Related Work

Several studies have been conducted on the EV charging scheduling problem (EVCSP). Here, we discuss some of the relevant literature that addresses the EVCSP in charging stations and parking lots. From the perspective of these charging service providers, the main objectives are to reduce costs [21–23, 25], improve service quality charging by maximizing the energy delivered [14, 16, 23], or maximizing charging incomes [14] while maintaining the physical constraints of the charging infrastructure. One largely used constraint is the charging infrastructure capacity that defines the overall power limit to avoid potential transformer overload and feeder congestion [14, 16, 21, 25]. For the charging power of CPs, some papers consider variable power in which the charging rate varies over time [14, 16, 25] others consider fixed constant power [4].

EV charging demands are usually defined by arrival and departure times and the requested charging energy. [4, 22, 23] consider uncertainly in the arrival time. [21] consider that EVs may arrive with or without an appointment. Departure times can be provided by the EVs drivers [4, 21, 24, 25] or it can be estimated based on historical behavior [23]. In [16], an EV is allowed to leave before its

initially provided departure time. Regarding the requested charging energy, [23] assumes that the EVs drivers will provide their desired state-of-charge when they plug-in their EVs. [20, 21] assume that the EV owners will directly request the energy demand expressed in kWh. Other papers consider charging EVs to the rated battery capacity [14, 16, 24, 25]. The requested charging energy can either be a hard constraint where the desired energy must be reached [4, 20, 21] or a soft constraint where the scheduler tries to achieve satisfactory energy by the departure time [25].

Different optimization methods were used for tackling these problems. [24] developed a charging scheme based on binary programming for demand respond application and a convex relaxation method is proposed to solve the charging scheduling problem in real-time. Authors in [25] propose a two-stage approximate dynamic programming strategy for charging a large number of EVs. [20] provide a model predictive control based algorithm. We can also find different metaheuristics, for example, particle swarm optimization [16, 21], a GRASP-like algorithm [4], memetic algorithm [4].

Although the above-mentioned studies have examined various aspects of the EV charging scheduling, they have essentially assumed there are a sufficient number of chargers for all EVs and thus the scheduler doesn't decide to which CP each EV is assigned. In this paper, hybrid heuristics and meta-heuristic are proposed to jointly assign the EVs to CP and schedule the EV charging. We consider different charging levels.

### 3 Problem Description and Formulation

We consider a charging station with  $m$  charging points (CPs). The switching on and off of each CP can be controlled. An electric vehicle (EV) can be connected to the CP but does not necessarily have to charge immediately. Each charging point  $i$ ,  $i = 1, \dots, m$  has a constant output power  $p_i$  (kW). We also consider the variable power model where the output power of each CP  $i$  can vary over time from 0 to  $p_i$ . The charging station has a maximum power supply of  $P^{max}$  (kW) which is insufficient to sustain simultaneous activation of all CPs. Thus, the sum of the output power of the CPs cannot exceed  $P^{max}$  (kW) at any time. The scheduling time horizon of one day and is divided into  $H$  time slots of length  $\tau$  (minutes). A set of  $n$  EVs that need charging in the whole day. Each EV  $j$ ,  $j = 1, \dots, n$ , submits a charging demand by providing the following information: the desired arrival time to the station  $r_j$ , the estimated initial state-of-charge at the arrival ( $e_j^0$ ), the desired state-of-charge at the departure ( $e_j^d$ ), the battery capacity  $B_j$  (kWh), and the departure time  $d_j$ . The scheduler collects all charging demands and determines a day ahead optimized charging schedule by assigning EV to each CP. Since there are a limited number of CP, the actual starting time for each EV can exceed the desired arrival time. An EV will occupy a CP from the assigned starting time until its departure time and cannot be plugged out during this period. The preemption of charging operation is allowed. Ideally, each EV should be charged to its desired state-of-charge by the departure time,

however, this may not be possible due to the limited number of CP and the charging station capacity limit. So, the objective of the scheduler is to minimize the difference between the desired state-of-charge and the final state-of-charge at the departure time.

In the following, a MILP model is proposed to jointly optimize the assigned of the EV to CP and the power allocation at each time slot.

In the case of a constant power model, the decision variables are:

$$x_{ij}^t = \begin{cases} 1 & \text{if EV } j \text{ is charging by the CP } i \text{ at time slot } t \\ 0 & \text{otherwise.} \end{cases}$$

$e_j^f$  : final state-of-charge of EV  $j$  at it departure time

- Objective: minimize the difference between the state-of-charge at the departure ( $e_j^f$ ) and the desired state-of-charge  $e_j^d$ .

$$\min \sum_{j=1}^n (e_j^d - e_j^f)$$

- Constraints:

$$\sum_{i=1}^m x_{ij}^t \leq 1 \quad \forall j, t \quad (1)$$

Constraints (1) ensure that each EV  $j$  is assigned to one CP at each time slot  $t$ .

$$\sum_{j=1}^n x_{ij}^t \leq 1 \quad \forall i, t \quad (2)$$

Constraints (2) ensure that each CP  $i$  charges one EV at each time slot  $t$ .

$$x_{i'j}^{t'} \leq 1 - x_{ij}^t \quad \forall i, j, t, i' \neq i \text{ and } t' < t \quad (3)$$

Constraints (3) ensure that each EV  $j$  is charged by one CP  $i$  i.e., The EV assigned to a CP cannot be moved to another CP.

$$e_j^0 \leq e_j^f \leq e_j^d \quad \forall j \quad (4)$$

$$e_j^f = e_j^0 + \frac{\tau \sum_{t=r_j}^{d_j} p_i \times x_{ij}^t}{B_j} \quad \forall j \quad (5)$$

Constraints (4) and (5) calculate the final state-of-charge of each EV  $j$ .

$$x_{ij}^t = 0 \quad \forall i, j \quad \forall t, t < r_j \text{ and } t \geq d_j \quad (6)$$

Constraints (6) ensure each EV  $j$  can only be charged between its desired arrival time  $r_j$  and departure time  $d_j$ .

$$x_{ij}^t + x_{i'j'}^{t'} \leq 1 \quad \forall i, j, t, j' : j' \neq j, t' \in [t, d_j] \quad (7)$$

Constraints (7) ensure that the CP  $i$  is reserved to the EV  $j$  from the time it begins to charge to its departure time  $d_j$ .

$$\sum_{i=1}^m \sum_{j=1}^n p_i \times x_{ij}^t \leq P^{max} \quad \forall t \quad (8)$$

Constraints (8) ensure that the total output power doesn't exceed the charging station limit at each time slot.

When considering variable power model, we add the decision variables  $p_{ij}^t$  that represent the power delivered by the CP  $i$  to EV  $j$  at time  $t$ . The constraints (5) and (8) will be replaced by the following constraints:

$$e_j^f = e_j^0 + \frac{\tau \sum_{t=r_j}^{d_j} p_{ij}^t}{B_j} \quad \forall j \quad (9)$$

$$\sum_{i=1}^m \sum_{j=1}^n p_{ij}^t \leq P^{max} \quad \forall t \quad (10)$$

We also add the following constraints to ensure that the delivered power to an EV  $j$  by CP  $i$  doesn't exceed its maximum output power  $p_i$ :

$$p_i \times x_{ij}^t \geq p_{ij}^t \quad \forall i, j, t \quad (11)$$

## 4 Proposed Methods

Solving the optimal charging scheduling problem with an exact method cannot be done in polynomial time [18]. Thus, heuristics and the Simulated Annealing (SA) metaheuristic combined with an exact method were developed.

### 4.1 Solution Representation

Solving the scheduling problem consists of determining the assignment of each Ev to CP and then, in case of a constant charging model, choose the appropriate time slots of charging. In the case of a variable power model, choose the appropriate charging rate at each time slot. Therefore, a feasible solution consists of the assignment of EVs to the CPs and the power allocation. The assignment of EVs to CPs is represented as a vector  $(\pi_1, \dots, \pi_n)$  where  $\pi_i$  is the sequence of EVs assigned to CP  $i$ . Once we have the EV-CP assignment, we solve the power allocation by determining the amount of power delivered by each CP to each EV at each time slot. To this end, we define the solution variables  $a_{ij}^t$  of the EV-CP assignment, which is equal to 1 if the EV  $j$  is plugged to the CP  $i$  at time  $t$ . The  $a_{ij}^t$  values will be used as inputs for solving the power allocation problem. To get  $a_{ij}^t$  from  $(\pi_1, \dots, \pi_n)$ , we simply schedule all EVs sequentially without idles times while respecting their arrival times. For each EV  $j$  in the sequence  $\pi_i$ , we

select the earliest possible starting time  $st_j = \max(r_j, d_{j'})$  where  $j'$  is the EV scheduled before the EV  $j$  in  $\pi_i$ . In this case, the variable  $a_{ij}^t$  will be equal to 1 for all  $t \in [st_j, d_j]$ . If  $st_j > d_j$ , the charging demand of EV  $j$  will be rejected but penalized in the objective function.

The proposed heuristics and neighborhood search in the SA deals with the EV-CP assignment problem. Then, for each solution, the power allocation problem is formulated as a linear programming model.

## 4.2 Heuristics for Solving the Assignment of EVs to CPs

In the following, we define greedy rules for the assignment of EVs to CPs.

**First Come First Served (FCFS) Heuristic.** First-come-first-served (FCFS) rule is a popular approach to schedule the EVs charging. It assigns the EV with the smallest arrival time to the first available CP.

**Interval Graph Coloring Based Heuristic (IGCH).** Consider the interval graph  $G = (V, E)$  where each vertex  $v \in V$  represents a charging demand of an EV  $j$ ,  $j \in n$  defined by its arrival time  $r_j$  and its departure time  $d_j$ . There is an edge  $e \in E$  between two vertices if and only if their associated intervals have a nonempty intersection i.e.  $(j, j') \in E$  if  $[r_j, d_j] \cap [r_{j'}, d_{j'}] \neq \emptyset$ . Assigning a set of EVs to a given CP is equivalent to the  $k$ -coloring problem of the graph  $G$ . The  $k$ -coloring problem is to assign a color  $c \in \{1, \dots, k\}$  to each vertex of  $G$  so that no adjacent vertices have the same color. The set of vertices colored with the same color corresponds to the set of EVs assigned to the same CP and it is called a color class. Since an interval graph is a chordal graph, the greedy coloring algorithm delivers an optimal coloring on a chordal graph following the perfect elimination orderings [5]. A perfect elimination ordering in a graph is an ordering of the vertices of the graph such that, for each vertex  $v$  and the neighbors of  $v$  that occur after  $v$  in the order form a clique. We use the lexicographic breadth-first (LexBFS) search proposed by [17] to find the perfect elimination ordering in linear time. We add randomness to the algorithm to generate difference perfect elimination orders by adding a random weight  $w$  to each vertex. Therefore, when two vertices have the same label, we choose the vertex with maximum weight  $w$ . Algorithm 1 shows the pseudocode of LexBFS ordering.

In the case where we have the chromatic number  $k$ , i.e. the number of color classes is less than or equal to the number of CP, we assign the EVs with the same color class to the same CP. We start with the color classes that have the greater cardinality (the highest number of vertices with the same color) and assign them to the CP with the greater charging output power. Otherwise, when  $k > m$ , each remaining non assigned EV  $j$  will be assigned to the CP that has the largest available time from  $r_j$  to  $d_j$ . The overall procedure of the IGCH algorithm is depicted in Algorithm 2.

**Algorithm 1: LexBFS**

```

input : Interval graph  $G = (V, E)$ 
output: A perfect elimination orderings  $\sigma = (v_1, \dots, v_n)$ 
1 for  $v \in V$  do
2 | label  $(v) \leftarrow []$ 
3 |  $w(v) \leftarrow \text{random}()$ 
4 end
5 for  $i = |V|$  down to  $1$  do
6 | choose a vertex  $v \in V$  with lexicographically maximal label with ties being
    | broken by  $w(v)$ 
7 |  $\sigma(i) \leftarrow v$ 
8 | for  $u \in \text{Neighborhood}(v)$  do
9 | | label  $(u) \leftarrow \text{label}(u). \text{concatenate}(u)$ 
10 | end
11 end
12 return  $\sigma$ 

```

**4.3 Exact Methods for Solving the Power Allocation Problem**

After determining the assignment of the EVs to CPs, the objective here is to decide the amount of electric power delivered by each CP at each time slot. We formulate both constant and variable power models as an integer linear programming (ILP) model.

**A Linear Programming for Constant Power Model:** We define the binary decision variables  $y_{ij}^t$ .

$$y_{ij}^t = \begin{cases} 1 & \text{if EV } j \text{ is charging by the CP } i \text{ at time } t \\ 0 & \text{otherwise.} \end{cases}$$

Let  $H_j = \{t | t \in H, a_{ij}^t = 1\}$ . We set  $y_{ij}^t = 0 \quad \forall t \notin H_j$

– Objective :

$$\min \sum_{j=1}^m (e_j^d - e_j^f)$$

– Constraints:

$$e_j^0 \leq e_j^f \leq e_j^d \quad \forall j \tag{12}$$

$$e_j^f = e_j^0 + \frac{\sum_{t=r_j}^{d_j} y_{ij}^t \times \tau \times p_i}{B_j} \quad \forall j \tag{13}$$

Constraints (12) and (13) calculate the final state-of-charge of each EV  $j$ .

$$\sum_{i=1}^n \sum_{j=1}^m y_{ij}^t \times p_i \leq P^{max} \quad \forall t \tag{14}$$

Constraints (14) ensure that the total output power doesn't exceed the charging station limit at each time slot.

**Algorithm 2:** Heuristic using interval graph coloring

---

```

input : Scenario of  $n$  EVs and  $m$  CPs
output: assignment of EVs to CPs
1 Construct the interval graph  $G$  of the scenario
2 Get a perfect elimination ordering  $\sigma$  of  $G$  using Algorithm 1
3  $k \leftarrow 1$ 
4  $\text{Color}(\sigma(0)) \leftarrow k$ 
5 for  $i = n$  down to 1 do
6    $v \leftarrow \sigma(i)$ 
7   for  $c = 0$  to  $k$  do
8     if  $\text{color } c \notin \text{Color}(\text{Neighborhood}(v))$  then
9        $\text{Color}(v) \leftarrow c$ 
10    end
11  end
12  if  $\text{all colors } c \in \text{Color}(\text{Neighborhood}(v))$  then
13     $k \leftarrow k + 1$ 
14     $\text{Color}(v) \leftarrow c$ 
15  end
16 end
17 Sort CPs in non-decreasing order of their output power
18 Sort the color classes in non-decreasing order their cardinality
19 Assign the EVs of the  $k$  first classes to the  $k$  first CPs
20 if  $k > m$  then
21   for each EV  $j$  in the remaining non assigned classes do
22     Choose the CP  $i$  that have the largest sub-interval of  $[r_j, d_j]$  where the
     CP is free (no EV is plugged in) and assign the EV CP  $i$ 
23   end
24 end

```

---

**A Linear Programming for the Variable Power Model:** We define the continuous decision variables  $p_{ij}^t$  which represents the power delivered by the CP  $i$  to EV  $j$  at time  $t$ .

$$\min \quad \sum_{j=1}^m (e_j^d - e_j^f)$$

Constraints :

$$e_j^0 \leq e_j^f \leq e_j^d \quad \forall j \quad (15)$$

$$e_j^f = e_j^0 + \frac{\sum_{t=r_j}^{d_j} p_{ij}^t \times \tau}{B_j} \quad \forall j \quad (16)$$

Constraints (15) and (16) calculate the final state-of-charge of each EV  $j$ .

$$a_{ij}^t \times p_i \geq p_{ij}^t \quad \forall i, j, t \in H_j \quad (17)$$

Constraints (17) ensure that the delivered power by each CP  $i$  doesn't exceed its maximum rated power  $p_i$  at each time  $t$ .



$$\sum_{i=1}^n \sum_{j=1}^m p_{ij}^t \leq P^{max} \quad \forall t \quad (18)$$

Constraints (18) ensure that the total output power doesn't exceed the charging station limit.

#### 4.4 Simulated Annealing

Simulated annealing algorithm, initially proposed by [8], has been successfully adapted to address several optimal resource scheduling problems. The pseudo-code of the SA algorithm is shown in Algorithm 3.

A candidate solution for SA represents the EV-CP assignment part described in Sect. (4.1). We obtain its objective value by solving the LP problem representing the power allocation as described in Sect. (4.3). In preliminary experiments, the population-based meta-heuristics turned out to be worse than SA and time-consuming due to solving an LP for each newly generated solution in the population. Thus, we pursue with SA only.

First, an initial solution  $S_0$  is generated using the FCFS rule. Starting from the initial solution  $S_0$ , SA generates a neighborhood solution  $S'$ . The difference in the objective value between the new solution  $S'$  and the current solution  $S$  is calculated as  $\Delta f = f(S') - f(S)$ . The neighborhood solution  $S'$  replaces the current solution based on the Metropolis criteria; It will replace the current solution if there is an improvement i.e.  $\Delta f < 0$ . If it also improves the best solution found so far, it will become the new global best solution  $S_{best}$ . Otherwise, a random number  $r$  is generated following the uniform distribution  $U[0, 1]$  and the neighborhood solution  $S'$  will become the current solution if  $r < e^{-\Delta f/T}$  where  $T$  is a parameter called temperature, which regulates the probability of accepting worsening solutions. The temperature is initially set to a value  $T_0$  proportional to the objective function value of the initial solution  $T_0 = \mu f(S_0)$  where  $\mu$  is a fixed parameter.

At each iteration  $l$ , the temperature is gradually decreased by a cooling scheme. The authors in the original SA paper [8] propose the following decreasing geometric cooling scheme:

$$T_{l+1} = \alpha T_l$$

Where  $0 < \alpha < 1$ . Typically, when  $\alpha$  is set to high implying a slow decrease of the temperature. We also consider the Lundy-Mees cooling scheme proposed by [11]:

$$T_{l+1} = \frac{T_l}{a + bT_l}$$

Connolly in [1] develops a variant of the Lundy-Mees scheme that set the parameter  $a$  to 1 and  $b$  in dependence of the initial temperature  $T_0$ , the final temperature  $T_f$  and the size of the neighborhood  $M$ :

$$b = \frac{T_0 - T_f}{MT_0T_f}$$

The algorithm will stop if it reaches the maximum number of iteration or after a fixed number of moves that did not result in accepted solutions. When the stopping criterion is met, the algorithm terminates returning the best solution  $S_{best}$  found so far.

**Neighborhood Generation.** A neighborhood solution is obtained through a perturbation on the assignment of Evs to CPs in the current solution by one of the following moves:

- Swap in the same CP: a neighborhood solution is generated by randomly choosing a CP and interchange the position of two EVs scheduled in this CP. The positions of EVs are randomly selected.
- Swap from two different CP: a neighborhood solution is generated by randomly swapping two EVs between two CPs. The CPs and EVs are randomly selected.
- Insert: a neighborhood solution is generated by randomly choosing an EV in a CP and move it to another position in another CP. The CPs and the position of the EV are randomly selected.
- Shift left: a neighborhood solution is generated by moving an EV at position  $p_1$  from a CP to position  $p_2$  in the same CP where  $p_2 < p_1$ . The positions  $p_1$  and  $p_2$  are randomly selected.
- Shift right: a neighborhood solution is generated by moving an EV at position  $p_1$  from a CP to position  $p_2$  in the same CP where  $p_2 > p_1$ . The positions  $p_1$  and  $p_2$  are randomly selected.

After each move, the power allocation is solved using the ILP models described in Sect. 4.3.

## 5 Experimental Analysis

To evaluate the performance of the proposed methods, several simulations are conducted, and the relevant results are discussed in this section. Note that all algorithms are implemented in C++ programming language. We use CPLEX 12.7 as a solver for the LP models in our heuristics and SA. Note that the CPLEX was not used to solve the MILP presented in Sect. (3) since we couldn't get results for small instances vehicles even after turning for several days due to out of memory errors.

### 5.1 Parameters Tuning for SA

The tuning of the particular optimization problem is essential to obtain good results. We use the IRACE (Iterated Racing for Automatic Algorithm Configuration) package [10] which allows an automatic parameter configuration using the Iterated F-race method. Two scenarios of each case were selected to be the training instances for IRACE. Table 1 presents the parameters of the SA along with the range of values that tested for each parameter. The final choices of the parameter values are also presented and they are used in all experiments in the following sections.

---

**Algorithm 3:** Simulated annealing

---

```

input : Initial solution  $S_0$ , maxNeighbours, maxAccepted, final Temperature
          $T_f$ , maxTrials,  $\mu$ 
output: best solution found  $S_{best}$ 
1  $S_{best} \leftarrow S_0$  ,  $S \leftarrow S_0$  ,  $T \leftarrow \mu f(S_0)$ ,  $M \leftarrow \frac{maxTrials}{maxNeighbours}$ ,  $trail \leftarrow 0$ 
2 do
3    $accepted \leftarrow 0$ ,  $generated \leftarrow 0$ 
4   while  $generated \leq maxNeighbours$  and  $accepted \leq maxAccepted$  do
5      $S' \leftarrow Neighbour(S)$ 
6      $\Delta f \leftarrow f(S') - f(S)$ 
7      $generated \leftarrow generated + 1$ 
8      $trail \leftarrow trail + 1$ 
9     if  $f(S') < f(S)$  or  $U(0,1) \leq e^{-\Delta f/T}$  then
10       $S \leftarrow S'$ ,  $accepted \leftarrow accepted + 1$ 
11      if  $f(S) < f(S_{best})$  then
12         $S_{best} \leftarrow S$ 
13      end
14    end
15  end
16  update T according to the cooling scheme
17 while  $trail \leq maxTrials$  and  $accepted > 0$ ;
18 return  $S_{best}$ 

```

---

**Table 1.** Values tested for the SA parameters tuning.

| Parameter         | value range          | Best      |
|-------------------|----------------------|-----------|
| $\mu$             | [0.01, 0.9]          | 0.12      |
| Max neighbours    | 20,50,100            | 50        |
| Cooling technique | LandyMees, Geometric | LandyMees |
| $\alpha$          | [0.01,0.9]           | -         |
| final temperature | [0.001,0.1]          | 0.01      |
| max accepted      | 0.1,0.2,0.3,0.4      | 0.1       |

### 5.2 Scenarios Generation

Regarding the charging station, we consider three cases with different number of CPs  $m = \{10, 20, 40\}$ . For each case, 30% of CPs deliver 3.7 kW, 30% deliver 11 kW, 30% deliver 22 kW and 10% deliver 43 kW. This charging rates are chosen from the standard IEC 61851 [19] that defines the classification of the different charging modes. The charging station maximum capacity  $P^{max}$  is set to 70% of  $\sum_{i=1}^n p_i$ . The scheduling time horizon is one day and the time slot  $\tau$  is set to 6 min.

To test the model implemented, we need to model the stochastic EVs charging demands. The EV arrivals are randomly occurring and independent events.

**Table 2.** Comparison of results with  $m = 10$ .

| scenario                  | $n$ | $k$ | FCFS |          | IGCH |             |      |      | SA   |             |      |          |
|---------------------------|-----|-----|------|----------|------|-------------|------|------|------|-------------|------|----------|
|                           |     |     | obj  | time (s) | mean | best        | std  | time | mean | best        | std  | time (s) |
| Model with constant power |     |     |      |          |      |             |      |      |      |             |      |          |
| 1                         | 20  | 12  | 0.52 | 0.47     | 0.53 | 0.26        | 0.15 | 0.29 | 0.25 | <b>0.13</b> | 0.09 | 56.15    |
| 2                         | 29  | 18  | 3.74 | 0.42     | 2.92 | 2.41        | 0.22 | 0.35 | 1.26 | <b>0.80</b> | 0.29 | 82.26    |
| 3                         | 24  | 16  | 2.05 | 0.35     | 2.20 | 1.68        | 0.39 | 0.28 | 0.55 | <b>0.25</b> | 0.17 | 68.69    |
| 4                         | 25  | 17  | 1.64 | 0.43     | 1.80 | 1.11        | 0.51 | 0.31 | 0.57 | <b>0.31</b> | 0.15 | 72.62    |
| 5                         | 30  | 16  | 4.79 | 0.44     | 2.57 | 2.34        | 0.36 | 0.35 | 2.01 | <b>1.34</b> | 0.47 | 93.43    |
| 6                         | 27  | 16  | 2.80 | 0.35     | 2.57 | 0.79        | 0.19 | 0.31 | 1.48 | <b>0.66</b> | 0.46 | 104.38   |
| 7                         | 29  | 14  | 2.10 | 0.42     | 2.01 | 1.26        | 0.23 | 0.35 | 0.96 | <b>0.66</b> | 0.21 | 80.72    |
| 8                         | 26  | 16  | 2.26 | 0.34     | 1.39 | 1.08        | 0.22 | 0.31 | 0.85 | <b>0.42</b> | 0.36 | 78.28    |
| 9                         | 22  | 14  | 2.41 | 0.30     | 1.78 | 1.15        | 0.31 | 0.25 | 0.48 | <b>0.10</b> | 0.35 | 91.35    |
| 10                        | 22  | 15  | 0.73 | 0.28     | 0.56 | 0.37        | 0.19 | 0.26 | 0.32 | <b>0.19</b> | 0.10 | 90.17    |
| 11                        | 27  | 9   | 2.21 | 0.50     | 1.32 | 0.79        | 0.24 | 0.37 | 1.02 | <b>0.53</b> | 0.35 | 77.03    |
| 12                        | 25  | 10  | 1.87 | 0.33     | 0.60 | 0.43        | 0.10 | 0.34 | 0.53 | <b>0.29</b> | 0.19 | 75.24    |
| 13                        | 20  | 8   | 2.03 | 0.27     | 0.91 | 0.59        | 0.07 | 0.23 | 0.46 | <b>0.19</b> | 0.22 | 59.76    |
| 14                        | 23  | 9   | 2.21 | 0.31     | 0.95 | 0.83        | 0.06 | 0.28 | 0.86 | <b>0.45</b> | 0.28 | 69.82    |
| 15                        | 22  | 9   | 1.47 | 0.29     | 0.64 | 0.44        | 0.12 | 0.26 | 0.52 | <b>0.32</b> | 0.17 | 67.86    |
| Model with variable power |     |     |      |          |      |             |      |      |      |             |      |          |
| 1                         | 20  | 12  | 0.27 | 0.30     | 0.25 | 0.02        | 0.14 | 0.23 | 0.03 | <b>0.00</b> | 0.05 | 61.50    |
| 2                         | 29  | 18  | 3.34 | 0.39     | 2.53 | 2.09        | 0.23 | 0.33 | 0.80 | <b>0.20</b> | 0.28 | 90.74    |
| 3                         | 24  | 16  | 1.79 | 0.32     | 1.73 | 1.03        | 0.28 | 0.26 | 0.23 | <b>0.00</b> | 0.16 | 76.03    |
| 4                         | 25  | 17  | 1.26 | 0.35     | 1.40 | 0.62        | 0.56 | 0.28 | 0.13 | <b>0.00</b> | 0.09 | 85.89    |
| 5                         | 30  | 16  | 4.51 | 0.38     | 2.52 | 1.98        | 0.56 | 0.33 | 1.48 | <b>0.66</b> | 0.46 | 104.38   |
| 6                         | 27  | 16  | 2.46 | 0.34     | 0.81 | 0.46        | 0.24 | 0.30 | 0.61 | <b>0.13</b> | 0.25 | 105.85   |
| 7                         | 29  | 14  | 1.90 | 0.36     | 1.59 | 0.61        | 0.42 | 0.32 | 0.68 | <b>0.27</b> | 0.23 | 101.39   |
| 8                         | 26  | 16  | 2.00 | 0.33     | 1.07 | 0.28        | 0.35 | 0.29 | 0.48 | <b>0.10</b> | 0.35 | 91.35    |
| 9                         | 22  | 14  | 2.06 | 0.28     | 1.24 | 0.33        | 0.39 | 0.24 | 0.48 | <b>0.09</b> | 0.25 | 97.57    |
| 10                        | 22  | 15  | 0.48 | 0.27     | 0.38 | 0.13        | 0.27 | 0.24 | 0.05 | <b>0.00</b> | 0.05 | 88.00    |
| 11                        | 27  | 9   | 1.89 | 0.34     | 0.76 | 0.32        | 0.28 | 0.34 | 0.67 | <b>0.09</b> | 0.30 | 90.27    |
| 12                        | 25  | 10  | 1.64 | 0.32     | 0.37 | 0.16        | 0.27 | 0.28 | 0.35 | <b>0.04</b> | 0.25 | 87.20    |
| 13                        | 20  | 8   | 1.73 | 0.25     | 0.59 | 0.50        | 0.02 | 0.23 | 0.17 | <b>0.00</b> | 0.16 | 70.88    |
| 14                        | 23  | 9   | 2.04 | 0.29     | 0.59 | 0.57        | 0.05 | 0.26 | 0.60 | <b>0.10</b> | 0.27 | 77.36    |
| 15                        | 22  | 9   | 1.14 | 0.28     | 0.30 | <b>0.00</b> | 0.20 | 0.24 | 0.20 | <b>0.00</b> | 0.10 | 74.38    |

**Table 3.** Comparison of results with  $m = 20$ .

| scenario                  | n  | k  | FCFS |          | IGCH |             |      |          | SA   |             |      |          |
|---------------------------|----|----|------|----------|------|-------------|------|----------|------|-------------|------|----------|
|                           |    |    | obj  | time (s) | mean | best        | std  | time (s) | mean | best        | std  | time (s) |
| Model with constant power |    |    |      |          |      |             |      |          |      |             |      |          |
| 16                        | 48 | 22 | 6.04 | 1.86     | 3.33 | <b>2.49</b> | 0.50 | 1.41     | 3.87 | 2.52        | 0.61 | 320.80   |
| 17                        | 50 | 22 | 6.22 | 1.50     | 3.59 | <b>2.09</b> | 0.64 | 1.40     | 4.25 | 3.45        | 0.42 | 340.73   |
| 18                        | 59 | 30 | 4.95 | 1.67     | 3.68 | 2.55        | 0.63 | 1.68     | 3.01 | <b>2.26</b> | 0.42 | 391.15   |
| 19                        | 66 | 28 | 7.39 | 1.98     | 5.90 | 4.78        | 0.71 | 1.91     | 5.39 | <b>3.84</b> | 0.74 | 428.59   |
| 20                        | 51 | 23 | 4.73 | 1.52     | 3.51 | <b>2.08</b> | 0.61 | 1.38     | 3.33 | 2.58        | 0.52 | 323.44   |
| 21                        | 58 | 33 | 6.60 | 1.70     | 5.61 | 3.88        | 0.86 | 1.65     | 4.10 | <b>2.80</b> | 0.71 | 381.73   |
| 22                        | 53 | 26 | 3.76 | 1.59     | 3.49 | 2.40        | 0.69 | 1.41     | 2.66 | <b>1.30</b> | 0.60 | 322.97   |
| 23                        | 68 | 37 | 9.24 | 1.98     | 7.80 | 5.41        | 0.85 | 1.90     | 6.08 | <b>4.60</b> | 0.67 | 448.97   |
| 24                        | 52 | 27 | 4.21 | 1.49     | 3.02 | <b>1.76</b> | 0.78 | 1.43     | 2.57 | 1.96        | 0.37 | 340.59   |
| 25                        | 54 | 28 | 5.89 | 1.59     | 5.17 | 3.82        | 0.81 | 1.46     | 4.02 | <b>3.09</b> | 0.58 | 370.09   |
| 26                        | 40 | 20 | 4.11 | 1.17     | 2.18 | 1.55        | 0.36 | 2.01     | 2.10 | <b>1.29</b> | 0.42 | 283.48   |
| 27                        | 40 | 18 | 4.05 | 1.13     | 2.05 | 1.33        | 0.28 | 2.11     | 2.28 | 1.42        | 0.51 | 259.32   |
| 28                        | 37 | 20 | 3.52 | 1.12     | 1.53 | 1.03        | 0.31 | 1.97     | 1.37 | <b>0.96</b> | 0.30 | 246.28   |
| 29                        | 33 | 15 | 2.88 | 0.91     | 0.90 | 0.62        | 0.14 | 1.58     | 0.85 | <b>0.44</b> | 0.30 | 206.95   |
| 30                        | 39 | 18 | 3.48 | 1.13     | 2.04 | 1.47        | 0.27 | 2.21     | 1.58 | <b>0.88</b> | 0.34 | 240.35   |
| Model with variable power |    |    |      |          |      |             |      |          |      |             |      |          |
| 16                        | 48 | 22 | 5.39 | 1.33     | 2.58 | <b>1.77</b> | 0.53 | 1.25     | 3.87 | 2.52        | 0.61 | 320.80   |
| 17                        | 50 | 22 | 5.79 | 1.37     | 3.02 | <b>1.84</b> | 0.80 | 1.28     | 3.64 | 2.50        | 0.61 | 368.63   |
| 18                        | 59 | 30 | 4.20 | 1.63     | 2.75 | 1.80        | 0.54 | 1.54     | 2.16 | <b>1.25</b> | 0.42 | 427.03   |
| 19                        | 66 | 28 | 6.72 | 1.82     | 5.22 | 3.40        | 0.65 | 1.72     | 4.44 | <b>3.24</b> | 0.64 | 480.56   |
| 20                        | 51 | 23 | 4.24 | 1.41     | 2.52 | <b>1.60</b> | 0.48 | 1.30     | 2.49 | 1.64        | 0.56 | 371.11   |
| 21                        | 58 | 33 | 6.03 | 1.58     | 5.22 | 3.18        | 0.87 | 1.49     | 3.54 | <b>2.20</b> | 0.47 | 427.77   |
| 22                        | 53 | 26 | 3.18 | 1.44     | 2.69 | 1.66        | 0.64 | 1.34     | 2.04 | <b>1.16</b> | 0.42 | 368.22   |
| 23                        | 68 | 37 | 8.75 | 1.86     | 6.66 | 5.12        | 0.73 | 1.77     | 5.46 | <b>4.29</b> | 0.53 | 507.92   |
| 24                        | 52 | 27 | 3.75 | 1.44     | 2.36 | <b>0.99</b> | 0.62 | 1.32     | 2.25 | 1.23        | 0.50 | 373.87   |
| 25                        | 54 | 28 | 5.44 | 1.48     | 4.46 | 2.67        | 0.78 | 1.37     | 3.64 | <b>2.34</b> | 0.57 | 390.13   |
| 26                        | 40 | 20 | 3.61 | 1.06     | 1.29 | <b>0.82</b> | 0.23 | 1.88     | 1.62 | 0.91        | 0.46 | 299.69   |
| 27                        | 40 | 18 | 3.75 | 1.09     | 1.27 | <b>0.38</b> | 0.29 | 1.97     | 1.68 | 0.97        | 0.57 | 280.38   |
| 28                        | 37 | 20 | 3.22 | 1.11     | 0.84 | 0.45        | 0.26 | 1.80     | 0.98 | <b>0.20</b> | 0.32 | 267.22   |
| 29                        | 33 | 15 | 2.60 | 0.86     | 0.41 | <b>0.09</b> | 0.19 | 1.49     | 0.58 | 0.11        | 0.20 | 241.82   |
| 30                        | 39 | 18 | 3.16 | 1.10     | 1.29 | 0.67        | 0.22 | 2.11     | 1.10 | <b>0.62</b> | 0.31 | 297.82   |

**Table 4.** Comparison of results with  $m = 40$ .

| scenario                  | $n$ | k  | FCFS |          | IGCH |             |      |          | SA   |             |      |          |
|---------------------------|-----|----|------|----------|------|-------------|------|----------|------|-------------|------|----------|
|                           |     |    | obj  | time (s) | mean | best        | std  | time (s) | mean | best        | std  | time (s) |
| Model with constant power |     |    |      |          |      |             |      |          |      |             |      |          |
| 31                        | 93  | 45 | 6.27 | 6.30     | 3.92 | <b>2.43</b> | 0.67 | 5.34     | 4.76 | 3.91        | 0.58 | 1621.11  |
| 32                        | 99  | 53 | 7.27 | 6.38     | 6.70 | 5.24        | 0.62 | 5.86     | 5.56 | <b>4.57</b> | 0.55 | 1433.02  |
| 33                        | 79  | 41 | 6.64 | 5.74     | 3.59 | <b>2.19</b> | 0.55 | 4.45     | 4.81 | 3.50        | 0.69 | 1122.35  |
| 34                        | 102 | 52 | 9.59 | 6.69     | 6.76 | <b>4.84</b> | 1.03 | 6.39     | 6.90 | 5.98        | 0.54 | 1442.63  |
| 35                        | 93  | 52 | 7.77 | 5.97     | 5.61 | <b>4.10</b> | 0.70 | 5.51     | 6.11 | 4.83        | 0.48 | 1382.38  |
| 36                        | 96  | 50 | 4.30 | 6.20     | 4.78 | 3.52        | 0.64 | 5.64     | 3.45 | <b>2.84</b> | 0.33 | 1462.46  |
| 37                        | 96  | 52 | 9.47 | 6.47     | 5.09 | <b>4.02</b> | 0.69 | 5.61     | 6.77 | 5.89        | 0.71 | 1480.53  |
| 38                        | 112 | 58 | 9.82 | 7.45     | 7.71 | <b>5.96</b> | 1.06 | 6.95     | 7.96 | 6.68        | 0.64 | 1648.97  |
| 39                        | 95  | 43 | 7.04 | 6.26     | 4.70 | <b>3.53</b> | 0.80 | 6.04     | 5.34 | 4.40        | 0.55 | 1379.94  |
| 40                        | 78  | 38 | 6.56 | 5.83     | 3.40 | <b>2.71</b> | 0.32 | 4.90     | 3.92 | 3.05        | 0.52 | 1127.74  |
| 41                        | 85  | 37 | 6.40 | 5.22     | 4.09 | <b>3.13</b> | 0.68 | 9.35     | 4.44 | 3.20        | 0.55 | 1188.21  |
| 42                        | 82  | 39 | 5.47 | 5.51     | 3.37 | <b>2.10</b> | 0.52 | 8.83     | 3.40 | 2.29        | 0.58 | 1245.61  |
| 43                        | 88  | 40 | 6.36 | 5.47     | 4.27 | <b>2.96</b> | 0.68 | 9.86     | 4.56 | 3.86        | 0.41 | 1241.81  |
| 44                        | 91  | 40 | 6.13 | 5.24     | 3.68 | <b>2.72</b> | 0.52 | 9.56     | 4.46 | 3.33        | 0.70 | 1239.36  |
| 45                        | 79  | 39 | 6.81 | 4.70     | 4.36 | <b>2.98</b> | 0.56 | 8.13     | 4.69 | 3.41        | 0.51 | 1131.08  |
| Model with variable power |     |    |      |          |      |             |      |          |      |             |      |          |
| 31                        | 93  | 45 | 5.32 | 5.75     | 2.59 | <b>1.21</b> | 0.77 | 5.07     | 3.59 | 2.34        | 0.63 | 1663.55  |
| 32                        | 99  | 53 | 6.30 | 7.47     | 5.27 | <b>3.32</b> | 1.03 | 5.46     | 4.87 | 3.36        | 0.62 | 1629.41  |
| 33                        | 79  | 41 | 5.79 | 5.15     | 2.47 | <b>1.44</b> | 0.62 | 4.37     | 3.88 | 2.97        | 0.49 | 1258.61  |
| 34                        | 102 | 52 | 8.16 | 6.60     | 5.32 | <b>2.61</b> | 1.02 | 5.79     | 5.62 | 4.43        | 0.64 | 1645.17  |
| 35                        | 93  | 52 | 6.69 | 5.86     | 4.02 | <b>1.71</b> | 0.84 | 5.18     | 4.75 | 4.01        | 0.39 | 1613.15  |
| 36                        | 96  | 50 | 3.18 | 6.30     | 3.40 | <b>1.27</b> | 0.75 | 5.36     | 2.44 | 1.52        | 0.43 | 1640.23  |
| 37                        | 96  | 52 | 8.31 | 6.27     | 3.39 | <b>1.72</b> | 0.60 | 5.37     | 6.24 | 4.67        | 0.70 | 1569.15  |
| 38                        | 112 | 58 | 8.57 | 7.26     | 5.88 | <b>3.84</b> | 0.84 | 6.43     | 6.93 | 5.41        | 0.65 | 1784.67  |
| 39                        | 95  | 43 | 5.79 | 6.24     | 3.02 | <b>1.21</b> | 0.80 | 5.65     | 4.17 | 3.27        | 0.62 | 1579.64  |
| 40                        | 78  | 38 | 5.61 | 4.71     | 2.12 | <b>1.20</b> | 0.52 | 4.55     | 3.18 | 1.94        | 0.63 | 1307.81  |
| 41                        | 85  | 37 | 5.56 | 5.20     | 2.78 | <b>1.23</b> | 0.76 | 8.74     | 3.47 | 2.49        | 0.52 | 1430.08  |
| 42                        | 82  | 39 | 4.70 | 4.95     | 1.93 | <b>1.01</b> | 0.41 | 8.82     | 2.65 | 1.71        | 0.58 | 1259.53  |
| 43                        | 88  | 40 | 5.33 | 5.19     | 2.93 | <b>1.71</b> | 0.76 | 8.08     | 3.43 | 2.25        | 0.54 | 1376.60  |
| 44                        | 91  | 40 | 5.11 | 5.19     | 2.38 | <b>1.32</b> | 0.59 | 9.29     | 3.49 | 2.33        | 0.63 | 1357.30  |
| 45                        | 79  | 39 | 5.98 | 4.38     | 2.86 | <b>1.60</b> | 0.63 | 8.67     | 3.95 | 2.51        | 0.60 | 1253.41  |

Therefore, the arrival time is modeled using a non-homogeneous Poisson Process with an arrival rate that varies  $\lambda(h)$  at each hour  $h = \{1, \dots, 24\}$ . The arrivals are likely high in the morning and low in the afternoon. The parking times  $pr_j$  follow an exponential distribution with a mean parking duration that also varies over time. There is no correlation between the arrival time and the parking time so the two variables can be generated independently [15]. The departure time  $d_j$  of each EV can be directly obtained  $d_j = r_j + pr_j$ . The initial state-of-charge at the arrival ( $e_j^0$ ) is considered uniformly distributed in the range of [20,70]. The desired state-of-charge of each EV  $j$  ( $e_j^d$ ) is uniformly chosen from  $[e_j^0, 100]$ . The battery capacities are chosen randomly from the list of current real-world EVs battery capacities [2].

We generate 15 scenarios for each case. In the first 10 scenarios, the chromatic number ( $k$ ) of the interval graph of the scenario is greater than the number of CPs while it is less or equal in the last five scenarios.

### 5.3 Simulation Results

Due to the stochastic nature of the IGCH and the SA algorithm, 30 independent executions were done for each scenario to obtain statistically significant results. The objective value is calculated for each scenario and we collect the mean, best, the standard deviation (std), and average execution time in seconds of each algorithm. Table 2, Table 3 and Table 4 shows the comparison of results obtained with  $m = 10$ ,  $m = 20$  and  $m = 40$  respectively. We highlight in bold the best solution found.

About comparison between the model with constant power and the model with variable power, the objective value in the second one was averagely lower by 26.6% by the IGCH and by 20% by the SA than the first model. Thus, charging demands can be satisfied more using the variable power model.

For all scenarios, the best solutions found by the SA and IGCH always outperform the FCFS heuristic. For the scenarios with  $m = 10$ , the SA algorithm outperforms the IGCH in all scenarios whereas it outperforms the IGCH in 18 scenarios out of 30 with  $m = 20$  and in 2 scenarios out of 30 when  $m = 40$ .

We perform the Mann-Whitney U test [13] to compare the results between the IGCH and the SA for each case. The Mann-Whitney U test is a non-parametric statistical test for determining whether two independent samples were drawn from a population with the same distribution. We compare the p-value to a significance level of 0.05. The p-value found was 0.0001, 0.3661 and 0.0038 for results with  $m = 10$ ,  $m = 20$  and  $m = 40$ , respectively. It suggests that there is no significant difference between the results with  $m = 20$  of SA and IGCH algorithms. However, we can conclude that the difference between the SA and IGCH results with  $m = 10$  and  $m = 40$  are statistically significant. The average time taken by the three methods is always greater in scenarios with  $m = 40$  when there are more EV charging demands. Also, the IGCH uses an average of 99.5% less computational time than SA. It is also noted that increment of computation time in the SA algorithm is due to solving multiple LP with CPLEX at each neighbor generation while there are one LP solved for each IGCH execution.

In conclusion, the proposed IGCH is significantly better than the FCFS mechanism and it is preferable to the SA regarding the execution time. Moreover, it can handle large scenario better than the SA.

## 6 Conclusion

In this paper, we addressed the electric vehicle charging scheduling problem (EVCSP) in a charging station with different charging modes to maximize the final state-of-charge of each EV by the departure time. To solve the optimization problem, we designed a heuristic based on interval scheduling and Simulated Annealing (SA) combined with linear programming. Variable power and constant power models were both studied and compared. Different scenarios were presented to evaluate the performance of the proposed algorithms. The results show that the variable power model is better for allocating power. The results also show that the proposed heuristic and SA can achieve an optimal solution and outperform the SA algorithm, and the performance is significantly better than the First Come First Serve (FCFS) mechanism. In this paper, we have assumed that the data on vehicle recharging (arrival time, departure time, state of charge, etc.) are known in advance. This assumption is realistic since many recharging service operators require a reservation of the recharging in advance to avoid queues. However, it is interesting to study the dynamic case of vehicle arrival. In future work, we will consider multi-objective optimization to reduce the charging cost.

## References

1. Connolly, D.T.: An improved annealing scheme for the QAP. *Eur. J. Oper. Res.* **46**(1), 93–100 (1990)
2. EVDB: Ev database (2020). <https://ev-database.org>
3. Franco, J.F., Rider, M.J., Romero, R.: An MILP model for the plug-in electric vehicle charging coordination problem in electrical distribution systems. In: 2014 IEEE PES General Meeting—Conference and Exposition, National Harbor, MD, USA, pp. 1–5. IEEE (2014)
4. García-Álvarez, J., González, M.A., Vela, C.R.: Metaheuristics for solving a real-world electric vehicle charging scheduling problem. *Appl. Soft Comput.* **65**, 292–306 (2018)
5. Gilmore, P.C., Hoffman, A.J.: A characterization of comparability graphs and of interval graphs. *Can. J. Math.* **16**, 539–548 (1964)
6. IEA: Global EV outlook (2020). <https://www.iea.org/reports/global-ev-outlook-2020>
7. Kang, Q., Wang, J., Zhou, M., Ammari, A.C.: Centralized charging strategy and scheduling algorithm for electric vehicles under a battery swapping scenario. *IEEE Trans. Intell. Transp. Syst.* **17**(3), 659–669 (2016)
8. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
9. Kleinberg, J., Tardos, E.: *Algorithm Design*. Pearson Education, India (2006)



10. López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace package: iterated racing for automatic algorithm configuration. *Oper. Res. Perspect.* **3**, 43–58 (2016)
11. Lundy, M., Mees, A.: Convergence of an annealing algorithm. *Math. Program.* **34**(1), 111–124 (1986)
12. Luo, L., et al.: Optimal planning of electric vehicle charging stations comprising multi-types of charging facilities. *Appl. Energy* **226**, 1087–1099 (2018)
13. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* 50–60 (1947)
14. Niu, L., Zhang, P., Wang, X.: Hierarchical power control strategy on small-scale electric vehicle fast charging station. *J. Cleaner Prod.* **199**, 1043–1049 (2018)
15. Pflaum, P., Alamir, M., Lamoudi, M.Y.: Probabilistic energy management strategy for EV charging stations using randomized algorithms. *IEEE Trans. Control Syst. Technol.* **26**(3), 1099–1106 (2018)
16. Rahman, I., Vasant, P.M., Singh, B.S.M., Abdullah-Al-Wadud, M.: On the performance of accelerated particle swarm optimization for charging plug-in hybrid electric vehicles. *Alexandria Eng. J.* **55**(1), 419–426 (2016)
17. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.* **5**(2), 266–283 (1976)
18. Sassi, O., Oulamara, A.: Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. *Int. J. Prod. Res.* **55**(2), 519–535 (2017)
19. IEC 61851-1: 2017 Standard: Electric vehicle conductive charging system-part 1: general requirements. The International Electrotechnical Commission, Geneva, Switzerland, 292, 7 February 2017
20. Tang, W., Zhang, Y.J.A.: A model predictive control approach for low-complexity electric vehicle charging scheduling: optimality and scalability. *IEEE Trans. Power Syst.* **32**(2), 1050–1063 (2016)
21. Wu, H., Pang, G.K.H., Choy, K.L., Lam, H.Y.: Dynamic resource allocation for parking lot electric vehicle recharging using heuristic fuzzy particle swarm optimization algorithm. *Appl. Soft Comput.* **71**, 538–552 (2018)
22. Wu, W., Lin, Y., Liu, R., Li, Y., Zhang, Y., Ma, C.: Online EV charge scheduling based on time-of-use pricing and peak load minimization: properties and efficient algorithms. *IEEE Trans. Intell. Transp. Syst.*(2020)
23. Yang, S.: Price-responsive early charging control based on data mining for electric vehicle online scheduling. *Electric Power Syst. Res.* **167**, 113–121 (2019)
24. Yao, L., Lim, W.H., Tsai, T.S.: A real-time charging scheme for demand response in electric vehicle parking station. *IEEE Trans. Smart Grid* **8**(1), 52–62 (2016)
25. Zhang, L., Li, Y.: Optimal management for parking-lot electric vehicle charging by two-stage approximate dynamic programming. *IEEE Trans. Smart Grid* **8**(4), 1722–1730 (2015)