Large scale EV charge scheduling under contractual power constraints: a priority rule-based semi-online algorithm

Pierre Blanchart, Marine Depecker-Quéchon, Cédric Auliac

Large scale EV charge scheduling under contractual power constraints: a priority rule-based semi-online algorithm

Pierre Blanchart[a], Marine Depecker-Quéchon[a], Cédric Auliac[a*]

[a]CEA, LIST, 91191 Gif-sur-Yvette CEDEX, FRANCE

Abstract

The scope of this research work is to study the implementation of Electric Vehicles (EVs) based Virtual Power Plants (VPPs) through optimized charging process of a fleet of vehicles spread over multiple charging stations. In this context, this paper focuses on smart charge scheduling for an ensemble of EVs operated by a fleet manager. We consider a semi-online setting where precise information about EVs' arrival times, parking durations and energy needs isn't available before the EVs’ connection to the charge spots ; only global statistics (computed at the fleet level) for these quantities are made available beforehand. In this context we propose, as a first step, to learn a priority function to be used in a second step to perform online scheduling for forthcoming EVs’ requiring a charging service, according to a "highest priority, processing first" scheme. As we ignore most about the shape of the priority function, we propose to learn this function in an offline way as a function of charging information collected from the EV fleet and consolidated in a historical database.

Keywords: Electric Vehicle; Virtual Power Plant; Charge Scheduling; Constrained Optimization; Statistical Learning

Résumé

L’objectif des travaux présentés dans cet article est d’étudier la faisabilité d’une centrale électrique virtuelle basée sur l’optimisation des processus de (dé)charge d’une flotte de Véhicules Electriques (VE). Cet article porte plus précisément sur l’optimisation et la planification des processus de charge pour une flotte de VE. Nous nous plaçons dans un contexte semi temps réel, où les informations relatives à l’heure d’arrivée des VE, leurs temps de stationnement ou encore leurs besoin en énergie ne sont connues que lors de la connexion du VE à la borne de recharge. Avant cet instant, nous ne disposons que d’une statistique globale (calculée à l’échelle de la flotte) relative à ces quantités. Dans ce cadre, nous proposons d’abord d’apprendre une règle de priorité, et de l’utiliser ensuite pour la planification en temps réel des processus de charge des VE, en favorisant la charge des véhicules prioritaires. La règle de priorité, inconnue *a priori*, est modélisée en mode hors-ligne à partir de l’historique des données de charge de la flotte, agrégée par leur gestionnaire de flotte.

Mots-clé: Véhicule Electrique; Centrale Electrique Virtuelle; Planification de Charge; Optimisation sous Contraintes; Apprentissage Statistique

[*] Tel.: +33-16908-8855
E-mail address: cedric.auliac@cea.fr.

1. Introduction

The main goal of this work is to support the sustainable large scale integration of Electric Vehicles (EVs) into the electricity distribution grid, based on the efficient, flexible and reliable management of their loads. Indeed, based on the optimization of EVs charging processes, EV fleets can support distribution grids and charging infrastructures operations. In this paper, focus is made on EV fleets managed by a company. In the considered business model, a fleet manager will act as an aggregator in order to provide energy services to electric utilities and grid operators participating into the electricity market. A well-known service which may be provided by an EV fleet is load shifting, relying on coordinated EVs charge scheduling so as to avoid charging during peak periods. Thus, a fleet manager would sell the fleet's capability to charge a given amount of energy over a given time horizon in a flexible manner, exploiting the fleet as a Virtual Power Plant (VPP). Concretely, if the EV-based energy services offered by the fleet manager meet the demand of a market player, the two parties would agree on a specific load profile to be implemented by the EV-based VPP. This contracted load profile defines the total amount of energy at the disposal of the fleet for each time slot of 15 minutes duration discretizing the contract-period (assumed to last the whole day).

Selling services based on EV fleet management involves different steps. At first, the fleet operator should provide a service offer consistent with the expected energy needs and charging capabilities of its fleet. Here, we assume that the fleet manager is able to compute estimates of these quantities based on the charging data collected from the fleet during the EVs' charging sessions (a charging session being a period during which an EV is connected to a Charge Spot and requires charging services). Based on this data, the fleet manager is able to design a panel of products/services to be offered on the energy market. Then, once a contract is settled, it has to be concretely implemented: the target fleet load profile must be decomposed into individual charging schedules to be achieved by each EV of the fleet. It is the responsibility of the fleet manager to ensure that the aggregated load profile of the fleet -obtained by summing up the individual load profiles of all fleet members- respects the load profile agreed by contract. Moreover, deriving from its contractual obligations may result in significant financial penalties for the fleet manager. This decomposition stage is a complex task as many aspects have to be taken into account simultaneously, related both to the end-users requirements and the technical characteristic of both the EVs and the Charge Spots (CS) they are connected to. In particular, EV drivers will connect "randomly" to CS and require different amounts of energy to be charged during variable parking durations. Moreover, once connected to a CS, EVs will have to deal with charging power limitations even if, for sakes of simplicity, we will assume that the fleet is homogeneous in terms of EVs' maximum battery capacity or charging power. Hence, the main difficulty in the decomposition stage will lie in the high variability of individual EV objectives and constraints, mainly related to end-users' needs.

In this context, our aim is to propose an automatic tool for smart resource allocation over EV fleets, enabling finding optimal schedules complying with both the contract established by the fleet manager and the end-user requirements in terms of energy needs and parking duration, as well as respecting the EVs related constraints. This paper is organized as follows: we provide an introductory state-of-the-art on constrained optimization and resource allocation in Section 2. Our semi-online rule-based scheduling approach is then detailed in Section 3. Eventually, we provide a proof of concept based on numerical experiments in Section 4.

2. A short overview of scheduling techniques

Optimizing the charging schedule of an EV fleet actually boils down to solve a resource allocation problem. The global power load capacity granted to the fleet has to be distributed among the different EVs available for charging in a smart manner so as to respect the settled contract and to adapt to the EVs needs and constraints. Hence, EV resource allocation can be seen as a service provided by the fleet manager both to the grid stakeholders and to the EV users. Concretely, computing a smart charge scheduling for a fleet of EVs yields to solve a constrained optimization problem of the form:

$$min\, f_0(x)\ \ subject\ to\ f_i(x) \leq b_i, i = \{1, \cdots, m\} \qquad (1)$$

Where f_0 is the *loss function* to be optimized under a set of *constraints* $f_i(x) \leq b_i, \forall\, i$. In our context, the loss function could be a measure of the user dissatisfaction while the constraints would imply e.g. respecting the technical limitations of the battery and in the meantime fulfilling the contract settled on the market.

2.1. Generic optimization methods

In the literature, classic optimization approaches have been used to solve such scheduling issues. When it can be formulated as a linear programming problem, *i.e.* involving a linear loss function subject to linear equality and inequality constraints, classical optimization algorithms can be used such as the Simplex algorithm (Dantzig & Thapa, 2003) or the Interior Point Method (Nemirovski & Todd, 2008). This is the point of view adopted in Huang et al. (2012) and Sortomme & El-Sharkawi (2011), for Plug-in Hybrid Electric Vehicles (PHEVs) charging optimization. Yet, EV charging optimization is a complex optimization problem that cannot always be formulated in a linear way. For example, in He et al. (2012) and Sortomme et al. (2011), EV charge scheduling is addressed from a non-linear perspective. Although non-linear optimization problems are usually more complex to solve, convenient formulations can be considered making the problem easier to cope with. In particular, non-linear but yet convex optimization problems have been extensively studied and can be efficiently solved through deterministic resolution methods such as gradient descent or Newton-Raphson algorithms, or heuristics such as the extension of the Simplex algorithm proposed in Wolfe (1959). Eventually, when the EVs' charge scheduling problem remains under a non-convex form, approximated methods can be used; see for example Xu & Wong (2011) and Clement-Nyns et al (2010). Contrary to deterministic approaches, in this setting, the solution space is explored through numerical sampling in a stochastic manner. This approach allows performing a more extensive search for solutions (when exhaustive search is impossible) in order to avoid remaining trapped in a local optimum. The main advantage of such methods is that they need few or no assumptions regarding the optimization problem at hand. On the other hand, they usually can't guarantee finding any optimal solution and come with a high computational cost. Nevertheless, meta-heuristics such as Genetic Algorithms (Goldberg, 1989) or Particle Swarm Optimization (Olsson, 2011), are often privileged to find approximate solutions for complex optimization problems.

2.2. Resource allocation methods

In parallel, *resource allocation* problems, also termed *scheduling* problems have their own growing literature. Hence, dedicated approaches have been proposed to solve this specific class of optimization issues, in particular in the context of "jobs" allocation over several CPUs (Central Processing Units). Automatic scheduling or resource allocation introduces a new perspective of tackling such constrained optimization problems; we refer to Brucker, Möhring, Neumann & Pesch (1999) for a detailed review. Resource allocation aims at distributing a limited resource over several users. Two main settings are usually distinguished, online (dynamic) and offline (static) resource allocation issues, involving different resolution methods. The offline setting refers to the fact that all the characteristics of the problem (users' demands and constraints) are known in advance and can be used to compute an optimized allocation *a priori*. On the contrary, in *an online resource allocation problem* neither of these inputs is known in advance; they become known independently when each task is released or when each user arises. The problem we consider in this paper appears to stand in the latter category.

Dynamic Programming (DP), an iterative method that aims at solving a collection of nested decision problems, is commonly used to solve resource allocation problems in the context of CPUs job allocation (see Bertsekas, 2012). Provided that the optimization problem to be solved is well suited for such a recursive approach, dynamic programming usually tends to have a lower computational cost than classic (meta-)heuristics, depending however on the dimension of the problem. Indeed, classical DP programs highly suffer from the curse of dimensionality (i.e. number of control variables to optimize). Approximate Dynamic Programming (ADP) methods have been introduced to cope with this issue (see e.g. Powell, 2007) by providing an approximate solution based on the estimation of the loss function through classic learning algorithms such as neural networks. However, like many heuristics, (A)DP methods tend to be very time consuming. Thus, they seem to be more adapted for solving offline rather than online scheduling issues. Indeed, resource allocation is very complex to solve in an online setting, because of the lack of visibility over the covered time horizon, and of additional constraints related to computational efficiency. In an online setting, delays for computing the allocation for next step are usually very short, limiting the choice of the resolution method. Coordinated EVs charge scheduling has been nevertheless tackled from an ADP perspective, e.g. in Han et al. (2010) and Xu and Wong (2011). Yet, such approaches don't appear to be suitable for the large scale EV scheduling issue considered in this paper.

Hence, rule-based scheduling may remain the best alternative, although most of real-word problems are too complex to be efficiently solved (at least in an optimal way) through such naïve approaches. Rule-based scheduling aims at allocating an available resource in an adapted manner, following basic rules defined *a priori*. We refer to Pinedo (2012) for a detailed review of rule-based scheduling in different settings. Most of these rules are defined in the offline setting though, when all information regarding users' demands, tasks' definitions, machines' capacities are known in advance. Indeed, it is easier to define smart rules with *a priori* information than running completely blind as in the online setting. Such rules can show poor performances in the online setting. In this paper we propose to get around that issue by building smart rules based on the estimation of priority functions from historical EV charging data that will drive a dispatching algorithm complying with EVs requirements and constraints.

3. A semi-online rule-based scheduling approach

In this section, we introduce our resource allocation problem formulation and provide a rule-based solution to the online scheduling problem corresponding to charging EVs when precise information about their arrival times, parking durations and energy needs is not available before they connect to a charge spot. Thus, we consider a semi-online setting exploiting day-ahead global statistics of the fleet (allowing anticipating the consumption patterns for the next day) in order to learn a priority function. The later will be used by an online rule-based scheduling algorithm in order to allocate energy to EVs requiring a charging service, according to an optimized "highest priority processing first" scheme. This will allow flexible allocation of the global amount of energy contracted by the fleet manager among the fleet's vehicles while satisfying end users' requirements and respecting basic technical constraints.

3.1. Problem formulation

Context definition - In this paper, we consider that time is discretized according to 15 minutes time slots, during which every ongoing process remains constant. Any change occurs only at the transition between two consecutive time slots. Typically, this implies that the charging power provided to an EV's battery by a CS does not change during 15 minutes time slots. We denote by T the total number of time slots during a day (in our case, $T = 96$) and by N the number of EVs requiring charging services during the day.

Charging session description – We first define the concept of charging session, which is the period during which an EV is connected to a CS and requires a specific amount of energy to charge its battery. A charging session is characterized by a set of parameters $\{\alpha_n, \omega_n, R_n, E_n^{max}\}$ that are supposed known (transmitted by the EV to the fleet manager) as soon as the EV connects to the charging infrastructure, where:

1. $\alpha_n \in \{1, \cdots, T\}$ is the index of the time slot where the charging session starts (connection to a CS)
2. $\omega_n \in \{1, \cdots, T\}$ is the index of the time slot where the charging session ends (disconnection from a CS)
3. R_n denotes the amount of energy requested by the end-user at the beginning of the charging session
4. E_n^{max} represents the technical limitations of the EV, i.e. the maximum amount of energy (related to the maximum charging power) that its battery can receive during one time slot.

For the sake of simplicity, we will assume that each car does not participate in more than one charging session per day. Therefore N reflects the number of charging sessions occurring during a single day and the size of the fleet considered by the fleet manager during that day. Of course, the value of N may vary from one day to the other. While the above pieces of information are released only at the beginning of a charging session and therefore are a priori unknown, we consider that the fleet manager can anticipate forthcoming charging sessions thanks to a model of its fleet's dynamic. We assume the existence of a simple model based on a three-dimensional histogram $h(t, g, l)$ which gives the number of vehicles connecting to a CS at time slot t with an energy need g (this modeling implies discretizing the drivers' energy requirements) and an expected parking duration l. Thus, we do not have a statistic associated with a particular vehicle but a global statistic at the scale of the fleet. Such statistic may be based on the analysis of historical charging data, on CS booking data (when such service is available) or on prior knowledge of charging sessions (for a delivery fleet for instance). The fourth parameter (E_n^{max}) is not modeled as it is supposed to be known by the fleet manager as a technical characteristic of its EVs.

In this paper, we are interested in designing the $T \times N$ schedule matrix $E = \{E_n\}, n \in \{1, \cdots, N\}$, the n-th line of which, $E_n = \{E_n(t)\}, t \in \{1, \cdots, T\}$, representing the charge plan (also referred to as "load profile") to be implemented for the n-th vehicle, i.e. the set of sequential energy transfers between the CS and the EV participating into the n-th charging session. Since the power delivered by the CS to the EV's battery is assumed to take on continuous values which will remain constant during 15 minutes time slots, we will consider $E_n(t)$ to be a continuous amount of energy transferred between a CS and an EV during charging session n at time slot t.

Our goal is to determine these energy transfers in an optimal way, so as to maximize the end-users' satisfaction while respecting the energy contract settled by the fleet manager as well as crucial technical limitations.

Respecting the energy contract – The first goal of this work is to design the charge plans so that the fleet's energy consumption over the day will match the contract settled by the fleet manager on the energy market. In order to respect this constraint, we impose that for each time slot, the total amount of energy provided to all the vehicles of the fleet remains below the energy quantity negotiated on the market for that time slot. As the contract value for a specific time slot is denoted by $C(t)$, we have:

$$C(t) = \sum_{n=1}^{N} E_n(t)\,, t \in \{1, \cdots, T\} \qquad (2)$$

By definition, $\forall n \in \{1, \cdots, N\}$ if $t < \alpha_n$ or $t > \ \omega_n$ then: $E_n(t) = 0$. For this approach to make sense, the total contract negotiated for a day has to be a good approximation of the overall energy need of the fleet over that day.

Maximizing the end-users satisfaction – In order to assess the drivers' satisfaction, we need to define D_n^t, the amount of energy actually delivered by the charge spot to the n-th EV until time slot $t \in\ [\alpha_n,\ \omega_n]$ as:

$$D_n(t) = D_n(t-1) + E_n(t) = \ \sum_{i=\alpha_n}^{t} E_n(i), \forall\ t \in\ ]\alpha_n, \omega_n] \qquad (3)$$

Therefore, $D_n^{\omega_n}$ denotes the total amount of energy delivered by the CS at the end of the charging session:

$$D_n(\omega_n) = \sum_{t=\alpha_n}^{\omega_n} E_n(t) \qquad (4)$$

We make the assumption that all the energy supplied by the charge infrastructure is used to charge the battery, i.e. that it is not used by any other electrical systems such as air (or battery) conditioning.
We seek to minimize the global end-users' dissatisfaction defined as the square of the normalized charging shortage (i.e. the square of the percentage of the requested charge which couldn't be performed) summed up over all charging sessions. The square function is introduced so that stronger dissatisfactions get more penalized as we consider that having many users experiencing a very light dissatisfaction is better than having a few users experiencing a strong dissatisfaction. Thus we wish to optimize the following problem:

$$min_E \sum_{n=1}^{N} \left(\frac{R_n - D_n(\omega_n)}{R_n}\right)^2 \ = \ min_E \sum_{n=1}^{N} \left(\frac{R_n - \sum_{t=\alpha_n}^{\omega_n} E_n(t)}{R_n}\right)^2 \qquad (5)$$

Additional constraints - Finally, in order to tackle the optimization problem described above, we also have to consider the following additional constraints:

1. an EV can't be charged more than required by the driver → $D_n(t) \leq R_n, \forall n, \forall\ t$
2. there is an upper bound on the amount of energy an EV can charge during a time slot → $E_n(t) \leq E_n^{max}$

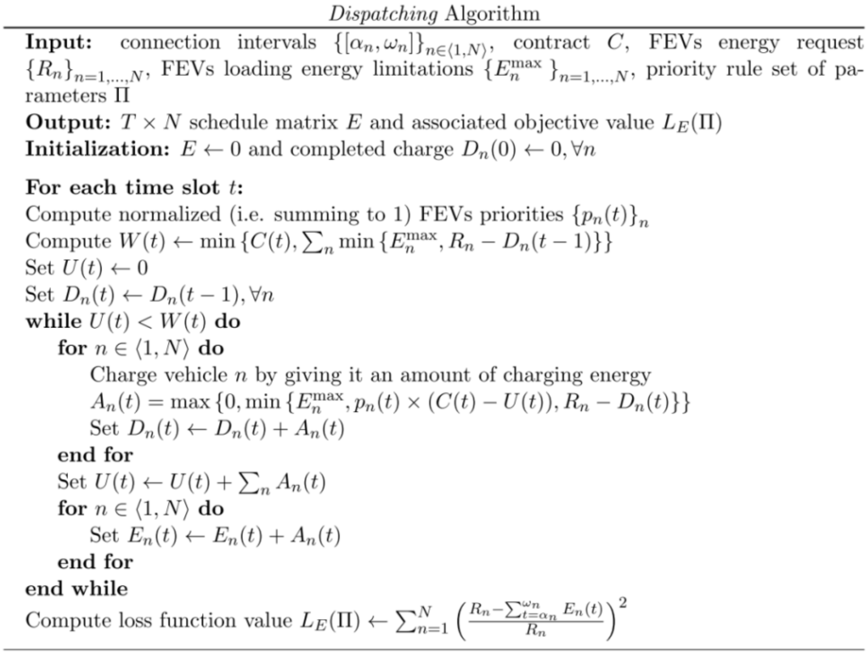
The first constraint may be relaxed in order to exploit any charging capacity left into the battery. The second constraint refers to the technical limitations of the EV and the charge spot it is connected to and does not take into account power supply limitations coming from the charge/grid infrastructure (which may vary through time). Note, that we also made the implicit assumption that drivers do not require energy above the maximum capacity of the battery of their vehicle.

3.2. An online Dispatching algorithm

As explained in section 2, the above scheduling problem has to be envisaged in an online fashion. Therefore, we propose to apply a classic resource allocation procedure, relying on a "highest priority processing first" rule-based approach and introducing slight modifications in the way the negotiated contract is dispatched between the vehicles. The "priority" determines the portion of the contracted amount of energy an EV will obtain during a time slot: the higher the priority, the higher the amount of energy allocated to the EV. In the following procedure, we adopt a rather straightforward way of inferring this amount of energy from priority values: the energy provided to an EV's battery will be its normalized priority value (ranging from 0 to 1) multiplied by the contracted amount of energy in the current time slot. The nature of the priority function will be detailed in the next sub-section and at this stage we consider a generic approach taking as input the set of parameters $\boldsymbol{\Pi}$ defining a priority function.

The physical limitations of the batteries or the eventuality of a vehicle finishing its charge conducted us to slightly modify the regular dispatching scheme detailed in Figure 1. Thus, during time slot t, a connected vehicle n is given the amount of charge: $max\{0, min\{E_n^{max}, p_n(t) \times (C(t) - U(t)), R_n - D_n(t)\}\}$ where $p_n(t)$ is the priority coefficient associated with the n-th vehicle for the t-th time slot and $U(t)$ is the amount of contracted energy that has been already dispatched for the t-th time slot. Note that $W(t)$ must not be confused with $C(t)$ as the former represents the total amount of energy that can be dispatched at the t-th time slot considering charging limitations. Therefore, $W(t)$ is a restriction over $C(t)$: $W(t) = min\{C(t), \sum_{n \in N} min\{E_n^{max}, R_n - D_n(t-1)\}\}$.

We iterate this dispatching rule for each time slot t as long as there is a possibility to dispatch a part of the contracted energy quantity over the plugged vehicles. This allows fitting the contract as much as possible while never exceeding it.



```
*Dispatching* Algorithm
Input: connection intervals $\{[\alpha_n, \omega_n]\}_{n \in \langle 1,N \rangle}$, contract $C$, FEVs energy request $\{R_n\}_{n=1,\dots,N}$, FEVs loading energy limitations $\{E_n^{\max}\}_{n=1,\dots,N}$, priority rule set of parameters $\Pi$
Output: $T \times N$ schedule matrix $E$ and associated objective value $L_E(\Pi)$
Initialization: $E \leftarrow 0$ and completed charge $D_n(0) \leftarrow 0, \forall n$

For each time slot $t$:
Compute normalized (i.e. summing to 1) FEVs priorities $\{p_n(t)\}_n$
Compute $W(t) \leftarrow \min\{C(t), \sum_n \min\{E_n^{\max}, R_n - D_n(t-1)\}\}$
Set $U(t) \leftarrow 0$
Set $D_n(t) \leftarrow D_n(t-1), \forall n$
while $U(t) < W(t)$ do
    for $n \in \langle 1, N \rangle$ do
        Charge vehicle $n$ by giving it an amount of charging energy
        $A_n(t) = \max\{0, \min\{E_n^{\max}, p_n(t) \times (C(t) - U(t)), R_n - D_n(t)\}\}$
        Set $D_n(t) \leftarrow D_n(t) + A_n(t)$
    end for
    Set $U(t) \leftarrow U(t) + \sum_n A_n(t)$
    for $n \in \langle 1, N \rangle$ do
        Set $E_n(t) \leftarrow E_n(t) + A_n(t)$
    end for
end while
Compute loss function value $L_E(\Pi) \leftarrow \sum_{n=1}^{N} \left( \frac{R_n - \sum_{t=\alpha_n}^{\omega_n} E_n(t)}{R_n} \right)^2$
```

Figure 1 - Dispatching scheme to compute the schedule from an arbitrary priority rule

3.3. Learning the priority function

In this subsection, we propose to define EVs' priority as a function of the remaining parking time and the remaining energy to be charged. Intuitively, in order to fulfill the end-users' requirements, the priority should be higher for vehicles with shorter remaining parking time and higher "remaining-to-complete" charge. Though we can anticipate these simple considerations to be true, we ignore most about the shape of the priority function which may be dependent on the problem at hand. We thus propose to learn this function in an offline way, based on the past observations of the EVs remaining parking time and remaining-to-complete SOC. We assume that the fleet manager consolidates charging data from its fleet to compute a reliable fleet's dynamics model based on the histogram $h(\boldsymbol{t}, \boldsymbol{g}, \boldsymbol{l})$ previously introduced in subsection 3.1.

In the following, we consider:

- the remaining parking time denoted by $\delta park_n(t) = max\big(0, min(\omega_n - \alpha_n + 1, \omega_n - t + 1\,)\big)$
- the remaining charge before completion denoted by $\delta g_n(t) = R_n - D_n(t-1)$

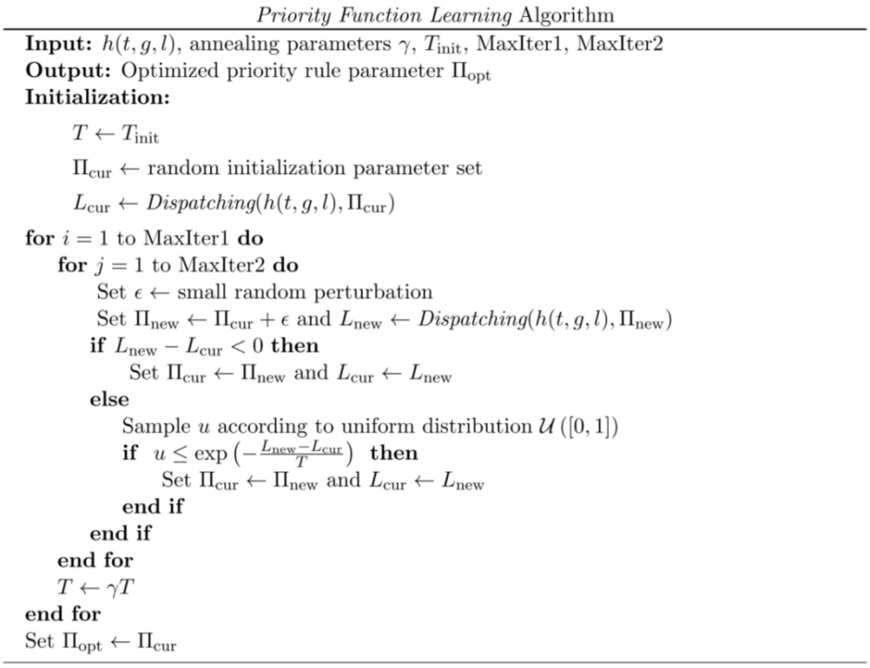
We wish to learn the two-dimensional priority function p depending on $\frac{\delta park_n(t)}{T}$ and $\frac{\delta g_n(t)}{100}$ which are normalizations of the quantity mentioned previously. In the following, we denote by $p_n(t)$ the value of the priority function p evaluated in $\left(\frac{\delta park_n(t)}{T}, \frac{\delta g_n(t)}{100}\right)$.

We chose to use a parametric model for this function as a weighted sum of bi-dimensional Gaussian functions:

$$p(X) = \sum_{k=1}^{K} \beta_k \frac{1}{2\pi |V_k|^{1/2}} exp\left((X' - X_k)' {V_k}^{-1} (X' - X_k)\right) \qquad (6)$$

where X is the two-dimensional vector $\left(\frac{\delta park_n(t)}{T}, \frac{\delta g_n(t)}{100}\right)'$. X' denotes the transpose of vector X. The set of parameters of the function $p(X)$, is denoted by $\Pi = \left\{\beta_k, X_k, V_k\right\}_{k=1,\cdots,K}$ where β_k is the weight of the k-th component of the mixture model and X_k, V_k are respectively the mean vector and covariance matrix of the k-th Gaussian component.

Using these notations, learning the priority function boils down to finding the set of parameters Π minimizing the loss function: $L_E(\Pi) = \sum_{n=1}^{N}\left(\frac{R_n - D_n(\omega_n)}{R_n}\right)^2$ which reflects the end-users dissatisfaction while respecting the energy contract and the EVs' charging limitations. In our setting, achieving this learning task amounts to perform a re-parameterization of the initial minimization problem as a function of the priority function parameters Π and to perform the optimization over this parameter. The value of the loss function $L_E(\Pi)$ is indeed obtained by computing the charging schedule E with the dispatching algorithm making use of a priority function of parameters Π. Thus, we cannot use standard mathematical optimization algorithms relying on an analytical expression of the problem functional such as gradient descent, Levenberg-Marquardt or BFGS algorithms. Among the various stochastic optimization algorithms we have tested, the simulated annealing yielded the best results while taking a very reasonable amount of time to execute. The procedure described below implements a SA scheme for our problem. In this procedure, "*Dispatching()*" refers to the dispatching scheme introduced in *Figure 1*. To evaluate $L_E(\Pi)$ in the algorithm, we use the non-normalized histogram $h(t, g, l)$ to generate a "canonical" scenario on which to run the dispatching algorithm presented in *Figure 1*. We proceed in the following way: for a bin $h(t, g, l)$ containing a value m, we generate m charging sessions corresponding to the triplets $\{t, t + l, g\}$ and we repeat this operation for all bins of the histogram. The missing parameter (E_n^{max}) required to fully specify a charging session is drawn randomly among a set of predefined maximum charging values.



```
Priority Function Learning Algorithm
Input: $h(t,g,l)$, annealing parameters $\gamma$, $T_{\text{init}}$, MaxIter1, MaxIter2
Output: Optimized priority rule parameter $\Pi_{\text{opt}}$
Initialization:
    $T \leftarrow T_{\text{init}}$
    $\Pi_{\text{cur}} \leftarrow$ random initialization parameter set
    $L_{\text{cur}} \leftarrow Dispatching(h(t,g,l), \Pi_{\text{cur}})$
for $i = 1$ to MaxIter1 do
    for $j = 1$ to MaxIter2 do
        Set $\epsilon \leftarrow$ small random perturbation
        Set $\Pi_{\text{new}} \leftarrow \Pi_{\text{cur}} + \epsilon$ and $L_{\text{new}} \leftarrow Dispatching(h(t,g,l), \Pi_{\text{new}})$
        if $L_{\text{new}} - L_{\text{cur}} < 0$ then
            Set $\Pi_{\text{cur}} \leftarrow \Pi_{\text{new}}$ and $L_{\text{cur}} \leftarrow L_{\text{new}}$
        else
            Sample $u$ according to uniform distribution $\mathcal{U}([0,1])$
            if $u \leq \exp\left(-\frac{L_{\text{new}} - L_{\text{cur}}}{T}\right)$ then
                Set $\Pi_{\text{cur}} \leftarrow \Pi_{\text{new}}$ and $L_{\text{cur}} \leftarrow L_{\text{new}}$
            end if
        end if
    end for
    $T \leftarrow \gamma T$
end for
Set $\Pi_{\text{opt}} \leftarrow \Pi_{\text{cur}}$
```

Figure 2 - Simulated annealing procedure to learn the priority rule

4. Simulation-based experiments

In this section, we conducted experiments on several simulated scenarios, following two steps: we first learned the priority function based on a "fleet's dynamics" model corresponding to a histogram $h(t, g, l)$ using the simulated annealing algorithm detailed in Figure 2. Then, we evaluated the performance of the dispatching algorithm using the learned priority function, for a wide range of scenarios. These were generated by sampling a large number of triplets from normalized histograms $h(t,g,l)$ modeling different fleet sizes and using the sampled data to build a new histogram, representing a specific scenario deriving from the initial model.

4.1. Hyper-parameters setting

As previously highlighted, the considered algorithms involve several hyper-parameters that have to be calibrated. For the presented simulations, all these parameters have been settled experimentally. In particular, the number of Gaussian components K in the priority function p (see *Equation (6)*) has been fixed to 7, since it proved experimentally to be the smallest number of components capturing efficiently the shape of the priority rule. The parameters of the simulated annealing procedure have also been tuned experimentally by trying different values. The parameters $MaxIter1$ and $MaxIter2$ have been fixed respectively to 500 and 5, the initial temperature T_{init} to 10 and the decay coefficient γ to 0.99. Yet, it has to be noted that the tuning of these parameters may be improved thanks to more systematic/extensive search strategies, yielding better performances for the corresponding algorithms.

4.2. Experimental design

In these experiments, the energy contract has been generated randomly (based on a uniform distribution), the only constraint being that the total amount of energy contractually offered to the fleet over the day matches the total fleet's users demand. Of course, the extent to which a contract is feasible will depend on its shape.
Besides, 20 different fleet models (corresponding to $h(t,g,l)$ functions), with different sizes ranging from 100 to 100000 EVs, have been generated. For each model, 200 scenarios were sampled and tested independently. Finally, for each model, we have computed the value of the loss function averaged over the 200 scenarios to allow comparisons of the results yielded by the different fleet sizes.

4.3. Experimental results

At first, the proposed scheduling method is compared with a reference approach given by the following priority function:

$$p_{baseline}\left(\frac{\delta park_n(t)}{T},\frac{\delta g_n(t)}{100}\right) = \left(1-\frac{\delta park_n(t)}{T}\right) \times \frac{\delta g_n(t)}{100} \qquad (7)$$

Where $\delta park_n(t)$ and $\delta g_n(t)$ are defined in subsection 3.3. This simple bi-dimensional ad-hoc rule gives more importance to vehicles with a short parking duration left and a big amount of energy remaining to be charged. Plot (a) of Figure 3 represents the graph of this simple rule, while plot (b) of same figure represents the learned rule for the first fleet model (i.e. for 100 EVs). We observe that the intuition implemented in the ad-hoc rule is also verified for a learned priority rule. Yet, in this setting, the remaining parking time appears as the main influencing factor, the remaining-to-charge SOC gaining influence when the parking time left is low.

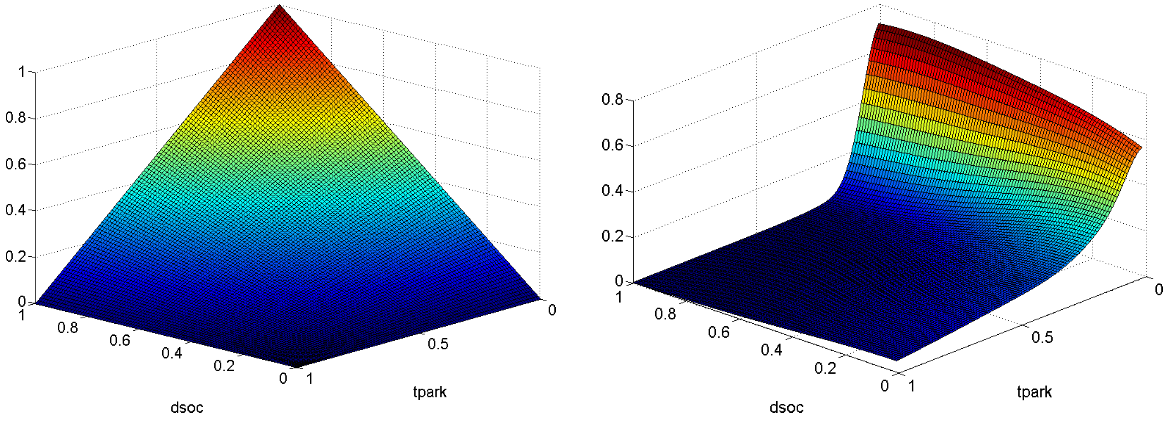


Figure 3 - (a) ad-hoc priority rule - (b) learned priority rule

Plot (a) of Figure 4 summarizes the average loss value reached for each fleet model with the associated variance. We notice that as we increase the fleet size, we obtain better results (i.e. smaller mean value and variability of the loss function across the different scenarios). This may be explained using simple statistical considerations: the actual behavior of a large fleet will be less likely to deviate from a regular pattern than a fleet counting a few vehicles. In the latter case, a few drivers changing their habits may invalidate the estimated fleet's dynamics model.

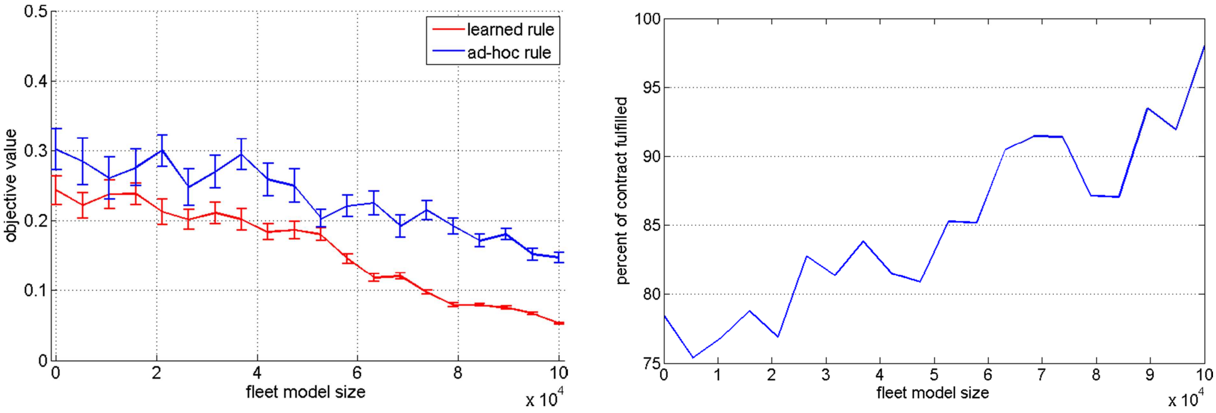


Figure 4 - (a) average loss value reached with the learned priority rule (red curve) versus the ad-hoc rule (blue curve). The vertical bars indicate the variability of the results across the simulated scenarios - (b) average percentage of the contract which could be fulfilled for each fleet model.

Another important aspect to envisage in experiments concerns the fulfillment of the negotiated contract. In plot (b) of Figure 4, we represent the average percentage of the contract which has been fulfilled, computed over all time slots of all scenarios for each fleet model (i.e. fleet size). Like for the users' satisfaction, the fit-to-contract grows with the size of the fleet. Indeed, within a generic setting (in this work, we did not designed constrained fleet behavior neither additional power supply limitations coming from the grid), a bigger fleet offers more flexibility in the way we allocate resources among multiple EVs. Besides, the fact that the above percentage isn't increasing steadily and is subject to jumps/drops may be explained by the fact that the contracts are generated randomly for each fleet model and may thus be more or less favorable to the computation of a feasible schedule. An important drop in the curve is generally explained by a generated contract being strongly unfavorable to the sampled charging sessions' scenarios.

The last aspect of our experiments concerns the execution time of the online scheduling. We ran all our experiments under Matlab on a standard workstation in single-threaded mode. We first observed that the computation time of a resources allocation step increases almost linearly with the size of the fleet and remains negligible compared to the duration of a time slot. Indeed, the online *dispatching* algorithm ran in about *one second* for the largest fleets, while the energy contract we considered had a 15 minutes granularity. The *Learning of Priority Functions* had a higher computational cost (around 30 minutes for the worst case scenarios) ; however, this has no impact on the operational use of the method as this learning step is performed offline.

5. Conclusion and perspectives

In this article, we presented a method to learn a priority function from an estimated model of a fleet's dynamics. The main advantage of this method is its ability to be applied in an online mode, where the connection and disconnection times of EVs, as well as their energy needs are not known a priori but may be anticipated, thanks for instance to statistical models. Another advantage is the possibility to schedule the charge of arbitrarily large fleets of vehicles since the only costly part in term of computation time is the offline learning of the priority rule. The scheduling algorithm can be applied in near real-time since it is only the application of a priority rule and thus is computationally efficient. The limitation of the approach mainly relies in the reliability of the fleet's dynamics model. Such a model may indeed be proved wrong by unexpected events, such as a public event causing people to make a much more intensive use of their cars in a certain geographical area. We are currently working on an extension of our approach in order to identify the deviations from the original fleet model and react to them in an online way by adapting the priority function. Besides, we are also considering a more sophisticated formulation of the problem in order to take into account additional constraints such as charging power limitations coming from the charge infrastructure or the distribution grid.

Acknowledgements

The work in this paper was partly funded by the European Commission as part of the 7th Framework Program with grant agreement number 285586. The authors would like to thank their partners in the corresponding EU research project e-DASH (www.edash.eu).

References

Bertsekas, D.P. (2012). Dynamic programming and optimal control, Vol. II: Approximate dynamic programming. Athena Scientific.

Boyd, S., & Vandengerghe, L., (2004). Convex optimization. Cambridge University Press.

Brucker, P.A., Möhring, R.H., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models and methods. European journal of operational research, pp. 3-41.

Clement-Nyns, K et al. (2010). The impact of charging plug-in hybrid electric vehicles on a residential distribution grid, IEEE Transactions on Power Systems.

Dantzig, J.B., & Thapa, M.N. (2003). Linear programming 2: Theory and extensions. Springer Verlag.

Goldberg, D. (1989). Genetic algorithms. Search, optimization and machine learning.

Han, S. et al. (2010). Development of an optimal vehicle-to-grid aggregator for frequency regulation. IEEE Transactions on Smart Grid.

He, Y. et al. (2012). Optimal scheduling for charging and discharging of electric vehicles, IEEE Transactions on Smart Grid.

Huang, F. et al. (2012). Design Optimization of PHEV charging station, IEEE Smart Grid Publications.

Lu, R., Yang, A., Xue, Y., Xu, L., & Zhu, C. (2010). Analysis of the key factors affecting the energy efficiency of batteries in electric vehicle. In World Electric Vehicle Journal Vol. 4 (pp. 9-13), Proceedings of EVS25.

Nemirovski , A.S., & Todd, M.J. (2008). Interior point methods for optimization. Cambridge university press.

Olsson, A. (2011). Particle swarm optimization: Theory, techniques and applications. Engineering, tools, techniques and tables.

Pinedo, M. (2012). Scheduling: Theory, algorithms and systems. Springer.

Powell, W. (2007). Approximate dynamic programming: solving the curses of dimensionality. Wiley Series in Probability and Statistics.

Sortomme, E., & El-Sharkawi, M.A. (2011). Optimal charging strategies for unidirectional vehicle-to-grid, IEEE Transactions on Smart Grid.

Sortomme , E. et al. (2011). Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses, IEEE Transactions on Smart Grid.

Wolfe, P., (1959). The simplex method for quadratic programming. JSTOR Econometrica.

Xu, J., & Wong, V.W.S. (2011). An approximate dynamic programming approach for coordinated charging control at vehicle-to-grid aggregator. IEEE Smart Grid Communications.