Real-Time Control of an Electric Vehicle Charging Station While Tracking an Aggregated Power Setpoint

Roman Rudnik, *Student Member, IEEE*, Cong Wang, *Student Member, IEEE*, Lorenzo Reyes-Chamorro, *Member, IEEE*, Jagdish Achara, *Member, IEEE*, Jean-Yves Le Boudec, *Fellow, IEEE*, and Mario Paolone, *Senior Member, IEEE*

***Abstract*—We consider the problem of controlling the charging of electric vehicles (EVs) connected to a single charging station that follows an aggregated power setpoint from a main controller of the local distribution grid. To cope with volatile resources such as load or distributed generation, this controller manages in real time the flexibility of the energy resources in the distribution grid and uses the charging station to adapt its power consumption. The aggregated power setpoint might exhibit rapid variations due to other volatile resources of the local distribution grid. However, large power jumps and minicycles could increase the EV battery wear. Hence, our first challenge is to properly allocate the powers to EVs so that such fluctuations are not directly absorbed by EV batteries. We assume that EVs are used as flexible loads and that they do not supply the grid. As the EVs have a minimum charging power that cannot be arbitrarily small, and as the rapid fluctuations of the aggregated power setpoint could lead to frequent disconnections and reconnections, the second challenge is to avoid these disconnections and reconnections. The third challenge is to fairly allocate the power in the absence of the information about future EVs arrivals and departures, as this information might be unavailable in practice. To address these challenges, we formulate an online optimization problem and repeatedly solve it by using a mixed-integer-quadratic program. To do so in real time, we develop a heuristic that reduces the number of integer variables. We validate our method by simulations with real-world data.**

***Index Terms*—Battery protection, dispatching of active distribution networks, electric vehicles (EVs), fairness, mixed-integer programming, real-time control.**

I. Introduction

THE penetration of electric vehicles (EVs) in the market is expected to significantly increase in the next decade. For example, given an expected growth in sales of 20%, there will be more than four million EVs in the USA by 2024 [1]. This will affect the planning and operation of electrical grids, with a particular emphasis on distribution networks. Indeed, uncoordinated and random EV charging might severely impact supply quality and continuity. In such a case, power flows and voltage-quality patterns throughout the grid will be affected considerably [2] and might increase the risk of local blackouts due to overloads. Studies [3]–[7] show how uncontrolled charging of EVs might jeopardize the operation of the power grid, causing voltage deviations or increasing power-system losses [8], [9]. A possible solution is grid reinforcement, which is often very expensive [10], especially in urban areas. An alternative solution is to dynamically control the power consumed by charging stations (CSs) and to keep the grid in safe operating conditions.

Consider a potentially common situation of a distribution network that contains local generation (e.g., photovoltaic (PV) panels) and an EV CS, both connected to the main grid through a transformer. When EVs are charged mostly by the PV production, a rapid PV power-drop (that could reach up to 60% of the rated power in few seconds [11]) will suddenly increase the power flow through the transformer. This might cause the transformer to exceed its rated power. Alternatively, the CS can reduce its charging power to compensate for the solar drop. However, given external conditions, this requires the CS to constantly update, the maximum charging power it can consume. To cope with such situations, the naive approach would be for the CS to know the precise amount of PV injected power, and the transformer rated power. In the literature, this problem is typically formulated as a cost-driven demand response optimization, where the forecast of arrivals and departures, and of the electricity price is usually fundamental. Yet, these solutions are not scalable for different setups due to the innate complexity of the problem. We take a different approach, which is fundamental for our proposal. Indeed, the optimal economic-decision should be taken by an entity (controller) that has a broader view of the system operation (e.g., [12]). If all flexible devices in the grid could be manipulated by a grid controller, their flexibility could be used to minimize the (local) electricity cost. In other words, we transfer this objective to the grid controller: Aggregated setpoints received by a CS from the distribution-grid controller are assumed to reflect electricity

Manuscript received October 16, 2019; revised February 8, 2020; accepted March 14, 2020. Date of publication April 2, 2020; date of current version September 18, 2020. Paper 2019-AAAE-1268.R1, approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Advanced Approaches and Applications for Electric Vehicle Charging Demand Management Committee of the IEEE Industry Applications Society. This work was supported by SCCER-FURIES. *(Corresponding author: Roman Rudnik.)*

Roman Rudnik, Cong Wang, Jagdish Achara, Jean-Yves Le Boudec, and Mario Paolone are with the École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland (e-mail: roman.rudnik@epfl.ch; cong.wang@epfl.ch; jagdish.achara@epfl.ch; jean-yves.leboudec@epfl.ch; mario.paolone@epfl.ch).

Lorenzo Reyes-Chamorro is with the Facultad de Ciencias de la Ingeniería, Universidad Austral de Chile, Valdivia 5090000, Chile (e-mail: lorenzo.reyes@uach.cl).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIA.2020.2984409

cost. This ensures that controllers have a clear separation of concerns when computing their optimal operation point. In this case, the CS needs to follow a power setpoint and to allocate this *aggregated* power setpoint among the connected EVs. For the grid controller to compute valid setpoints, it should be informed about the flexibility of all controlled resources. As the flexibility of the CS depends on the situation (number and type of connected EVs, state of energy of EV batteries, etc.), it has to be updated repeatedly.

The allocation of power to EVs is a difficult task because, as previously mentioned, an aggregated power setpoint can change dramatically in a few seconds. The naive power allocation, which would transfer these variations directly to the EVs, could increase their battery aging by creating large power jumps, mini-cycles, and frequent ON–OFF switching of EVs. Furthermore, the power should be allocated fairly considering that each EV has its own energy demand and departure time. Indeed, the aggregated power setpoint might not be enough to satisfy the demand of all EVs. Sortomme and El-Sharkawi [13] minimize the battery-degradation cost associated with additional cycling, assuming that there is sufficient amount of power to satisfy the EVs demand, and fairness issues are not addressed, whereas studies in [14]–[16] propose charging schemes that consider fairness of the power allocation among EVs, but without accounting for battery wear. Leehter *et al.* [17] use an ON–OFF strategy stating that a constant power minimizes the battery wear. However, the large power jump, from no charge to maximum charge, could represent a significant impact on the battery's lifetime. In our formulation we penalize the ON/OFF transition and the power change, so that the charge smoothness is guaranteed. We develop a method that considers both battery wear and fair-demand satisfaction and that tracks the aggregated power setpoint. Also, the battery size, charging rate, initial state of energy, and desired state of energy at departure can be different for every EV.

Deilami *et al.* [18] propose a load-management control strategy for minimizing the power losses and for improving the voltage profile during peak hours by assuming that EVs are scheduled in three different types of charging periods. Ma *et al.* [19] develop a decentralized control scheme, using concepts from noncooperative games, showing optimality when the EVs characteristics are identical (same departure time, energy demand and maximum charging power) and all charging schedules are agreed upon with the CS one-day ahead. He *et al.* [20] propose an online charging algorithm, assuming that no EVs will arrive when a charging schedule is made. In [21], a chance-constrained optimization problem is formulated to minimize the charging cost. However, the problem is computationally heavy and might not be applicable in actual real time. We propose to take a different approach without the need for scenarios but only with updated information, thus significantly reducing the computational burden and using the minimum amount of information to represent historical and future events in the decision taken at each time step. Studies [2], [22], and [23] assume that all the EVs have the same charging rate. But, such assumptions do not hold in practice. In contrast, our method considers the EVs heterogeneity. We do not have any information about future arrivals and departures, nor can we know the real amount of time any charging will take; we can only estimate it.

A common assumption in the literature is that the charging power of an EV is a continuous value between 0 and the maximum power (e.g., [16], [22], and [23]). However, in reality, this is not the case because an EV can be either switched OFF and consume no power, or charge at a power that lies between nonzero bounds, where the minimum charging power cannot be arbitrarily small. Liu *et al.* [24] developed a distributed control scheme that supports ON–OFF states, but it is limited to a constant power when ON, whereas we consider both switch ON and OFF possibilities and not arbitrarily small minimum charging power.

The proposed method has the following objectives:

1) follow an aggregated power setpoint;
2) minimize the battery degradation of each EV;
3) fairly allocate the power proportional to the EVs needs.

In order to achieve these objectives, we define novel metrics and use them to construct a dedicated optimization problem. As the charging power is discontinuous (the minimum charging power is not arbitrarily small), our optimization problem is mixed integer. As a mixed-integer optimization problem is difficult to perform in real time, we propose a heuristic for reducing the number of integer variables, thus reducing the complexity of the problem.

Our main contributions are as follows.

1) We assume that the control scheme has no internal information about battery charging (e.g., ramping rates, current state of energy), nor about the actual departure time of EVs. This is more realistic, as modern CSs are myopic to this kind of parameters.
2) Our method provides subsecond-scale control. While most of the existing methods work on a minute scale, ours enables the CS to react faster to changes in the grid.
3) We use a realistic model for the battery charging power, i.e., an EV is either switched OFF (charging power is 0 W), or its power lies within nonzero bounds.
4) We minimize the battery wear by avoiding large power jumps and by reducing the number of cycles.

The article is organized as follows. In Section II, we state the charging control problem. In Sections III and IV, we describe all the details of the developed control strategy. We provide the numerical examples with performance metrics to validate our method in Section V. Finally, Section VI concludes this article.

II. Problem Statement

A. CS Model

We consider a CS that can host N EVs. Time is discretized in constant interval, indexed by k. The CS keeps track of the number of connected EVs at every step k. A newly arrived EV cannot begin charging before being instructed by the CS. Each EV, say i, upon its arrival, is assumed to inform the CS of the following:

1) charging-power bounds $P_i^{\min}$ and $P_i^{\max}$;
2) energy demand E_i^{dem};
3) expected departure time k_i^{dep}.

Information about future arrivals, future expected departures, and future demands is unknown. Also, the CS has access to the measured power $\hat{P}_i[k]$ of EV i at every time k. The CS is able

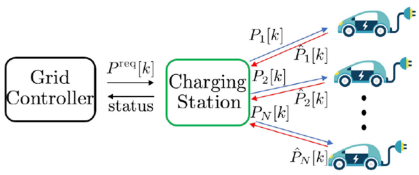


Fig. 1. General setup of the considered CS.

to control the charging power of an EV by sending the setpoint $P_i[k]$ to EV i at time k (see Fig. 1). As the CS is connected to a three-phase system, we assume that the EV charging power is evenly balanced on the three-phases, and we do not target phase-balancing. The CS receives an aggregated setpoint $P^{\text{req}}[k]$ from the grid controller. In return, it sends its updated status (see Section V for details).

B. Constraints of the EVs

We assume that the CS has the ability to stop the charge of an EV. The individual power flexibility of EV i is defined by the set $\{0\} \cup [P_i^{\min}, P_i^{\max}]$. We denote the ON/OFF *decision* for EV i at time k by $\omega_i[k]$. Specifically, $\omega_i[k] = 1$ (respectively, 0) means that we decide to switch ON (respectively, OFF) EV i at time k. We assume that an EV is initially switched OFF, upon arrival.

When receiving new setpoints from the CS, the EVs cannot immediately change their charging power due to delays.

1) *Reaction* delay is the time an EV takes to start modifying its power after receiving a new setpoint.
2) *Implementation* delay is the time an EV takes to reach a new setpoint, which depends on the EV charger ramping rate.

We say that an EV is *locked* if it is in the process of reacting or implementing a setpoint. As the specific delays are usually different for every type of EV, it is difficult to know their exact values. Therefore, we take a conservative upper bound T^L (20 s in this article). Specifically, we consider that, after receiving a setpoint, any EV will be locked for the *locking period* T^L.

Note that the locking of the EVs temporarily shrinks the flexibility of the CS as the amount of EVs that can change power varies from one control cycle to another. Thus, this information is supposed to be constantly sent to the grid controller. As the ramping rates and delays are unknown, it is impossible to know *a priori* exactly how the charging power will change when an EV is locked.

C. Power Allocation to EVs

The CS needs to allocate the time-varying aggregated power setpoint to the connected EVs. The purpose of the power-allocation strategy is to allocate the consumed power in such a way that the satisfaction of the EV demands are maximized and that their batteries are subjected to minimal wear. In particular, it might not be possible to satisfy all EV demands if the power capacity is not sufficient. In such a case, power should be allocated fairly. Furthermore, fast variations of the aggregated setpoint should be smoothed, otherwise its direct implementation can degrade the EV batteries. In summary, the objectives of the allocation strategy are as follows:

1) track the aggregated setpoint from a grid operator;
2) minimize the wear of EV batteries;
3) maximize the EVs energy-demand satisfaction while maintaining fairness;
4) minimize the number of times the CS stops the charging of an EV while it is plugged-in.

In this article, we consider all four objectives together. In the following section, we formulate a specific mixed-integer program and show how we solve it in real time.

III. Control Scheme at the CS

The CS computes setpoints for all EVs that are not locked at time k. At each time k, the CS has the following inputs, states, and outputs.

1) Inputs (as introduced in Section II):
   a) aggregated power setpoint $P^{\text{req}}[k]$;
   b) measured powers $\hat{P}_i[k]$ for every EV i;
   c) ON/OFF decisions $\omega_i[k-1]$ for every EV i.
2) States (will be detailed Section III-B):
   a) history of charging power changes $\lambda_i[k]$;
   b) the desire of an EV to be charged $\rho_i[k]$.
3) Outputs (as introduced in Section II):
   a) the power setpoints $P_i[k]$;
   b) ON/OFF decisions $\omega_i[k]$.

According to the objectives described in Section II-C, the CS solves the following optimization problem:

$$\min_{\mathcal{P}[k],\Omega[k]} \; c_0 f_0(\mathcal{P}[k], P^{\text{req}}[k]) + c_1(f_1(\mathcal{P}[k], \Lambda[k]) + f_2(\Omega[k-1], \Omega[k], R[k], \hat{\mathcal{P}}[k])) + f_3(\mathcal{P}[k]) \quad (1)$$

$$\text{s.t.} \quad P_i^{\min}\omega_i[k] \le P_i[k] \le P_i^{\max}\omega_i[k] \quad (2)$$

$$\omega_i[k] \in \{0,1\} \quad \forall i \in \mathcal{C}[k] \quad (3)$$

where the following conditions hold:

1) $\mathcal{C}[k]$ is the collection of EVs that are unlocked at time k (just before starting a new computation of setpoints);
2) $\mathcal{P}[k]$ is the collection of setpoints $P_i[k]$ that will be computed for each EV in $\mathcal{C}[k]$;
3) $\hat{\mathcal{P}}[k]$ collects measured powers $\hat{P}_i[k]$ for all EVs;
4) $\Omega[k]$ is the collection of ON/OFF decisions that will be computed for each EV in $\mathcal{C}[k]$;
5) $\Lambda[k] = (\lambda_i[k])_{i=1,2..}$ and $R[k] = (\rho_i[k])_{i=1,2...}$;
6) functions f_0, f_1, f_2, f_3, and parameters $c_0, c_1 > 0$ will be described in the following sections.

Note that this optimization is a mixed-integer problem due to the presence of the collection of binary control variables $\Omega[k]$. In addition, this problem is online, with $\lambda_i[k]$ and $\rho_i[k]$ being the proxies for the history and future, respectively.

A. Aggregated Power Setpoint Tracking

The first term in (1) is responsible for tracking the aggregated power setpoint $P^{\text{req}}[k]$. As the locked EVs either react to or implement a previous setpoint, they cannot follow a setpoint and should be removed from the aggregated setpoint, i.e., $\tilde{P}^{\text{req}}[k] = P^{\text{req}}[k] - \sum_{i\in\mathcal{L}[k]} P_i[k]$, where $\mathcal{L}[k]$ collects all the locked EVs. In this case, $P_i[k]$ represents the very last setpoint that a locked

EV has received. This impedes the CS from reallocating the same power in the unlocked EVs. Finally, f_0 can be expressed as

$$f_0(\mathcal{P}[k], P^{\text{req}}[k]) = \left(\widetilde{P}^{\text{req}}[k] - \sum_{i \in \mathcal{C}[k]} P_i[k] \right)^2. \quad (4)$$

Note that, in this case, the aggregated setpoint might not be followed exactly, due to the delays mentioned in Section II-B. However, such uncertainty can be informed to the grid controller, in order to use it in its setpoint computation process. In Section V, we explain how this uncertainty is computed.

B. Battery Wear

In order to minimize the impact of changing power in the EV batteries, we use f_1 and f_2 in the objective function. f_1 penalizes the deviation between the setpoint and the measured power, together with the changes in the measured power. f_2 penalizes sudden a switch OFF of the EVs caused by the CS. To formalize f_1 and f_2, let us introduce new variables. As our method is online, we introduce two *nonlinear* integral terms to account for the following conditions:

1) the past behavior of EVs' charging power;
2) the desire of an EV to be charged.

The first of these terms, $\lambda_i[k] \in [0.5, 1]$ per EV i, quantifies how long ago and how large the power changes were. This is used as a priority metric: the smaller λ_i is, the more priority to change power there is. Let k'_i be the time of the most recent change of the setpoint for EV i before k (so that $P_i[\kappa] = P_i[k'_i]$ for $\kappa = k'_i, k'_i + 1, \ldots, k-1$). Note that k'_i is also a function of k but, for the ease of notation, we drop this dependence. When EV i arrives, $P_i[k'_i]$ is set to zero. Consequently, we take

$$\lambda_i[k] = \begin{cases} \lambda_i[k'_i] + \left(\frac{|\hat{P}_i[k] - \hat{P}_i[k'_i]|}{P_i^{\max}} \right) (1 - \lambda_i[k'_i]) \\ \qquad \text{if } |\hat{P}_i[k] - P_i[k'_i]| > \epsilon \text{ and } k - k'_i < T^L \\ (\lambda_i[k-1] - 0.5)\delta + 0.5, \text{otherwise.} \end{cases} \quad (5)$$

The first case of (5) occurs when EV i is locked and the setpoint is not yet implemented.[1] In this case, $\lambda_i[k]$ increases linearly with respect to the implemented power change (gray area on Fig. 2). It is defined by the following conditions:

1) if $\hat{P}_i[k] = \hat{P}_i[k'_i]$, then $\lambda_i[k] = \lambda_i[k'_i]$;
2) if $|\hat{P}_i[k] - \hat{P}_i[k'_i]| = P_i^{\max}$, then $\lambda_i[k] = 1$.

In the second case, $\lambda_i[k]$ decreases exponentially with a decay δ (see Fig. 2). Observe that the right-hand side of (5) is always in [0.5, 1].

f_1 uses the term $P_i[k] - \hat{P}_i[k]$ and $f_1(\mathcal{P}[k])$ as follows:

$$f_1(\mathcal{P}[k], \Lambda[k]) = \sum_{i \in \mathcal{C}[k]} (P_i[k] - \hat{P}_i[k])^2 \lambda_i[k]. \quad (6)$$

The second term, $\rho_i[k] \in [0.5, 1]$, expresses the desire of an EV i to charge. It is also used as a priority metric: the larger the ρ_i is, the higher the priority to increase the power is. Note that, the CS can keep track of the remaining energy demand $\Delta E_i^{\text{dem}}[k]$ of

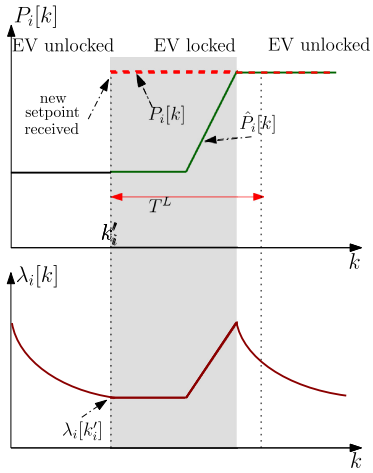


Fig. 2. Evolution of $\lambda_i[k]$.

EV i at time k, and expected remaining charging time $k_i^{\text{dep}} - k$. Therefore at time k, the CS computes the power that EV i needs to satisfy its demand as $\Delta E_i^{\text{dem}}[k]/(k_i^{\text{dep}} - k)$. And, for $k = k_i^{\text{arr}}$ this power equals $\Delta E_i^{\text{dem}}[k_i^{\text{arr}}]/(k_i^{\text{dep}} - k_i^{\text{arr}})$. With this, we compute the unitless quantity per EV as follows:

$$\zeta_i[k] = \frac{1}{P_i^{\max}} H \left(\frac{\Delta E_i^{\text{dem}}[k_i^{\text{arr}}]}{k_i^{\text{dep}} - k_i^{\text{arr}}}, \frac{\Delta E_i^{\text{dem}}[k]}{k_i^{\text{dep}} - k} \right) \quad (7)$$

where H represents the harmonic mean. By the property of the harmonic mean, $\zeta_i[k] \in [0, \frac{2\Delta E_i^{\text{dem}}[k_i^{\text{arr}}]}{P_i^{\max}(k_i^{\text{dep}} - k_i^{\text{arr}})}]$, which depends on the initial state of an EV. Moreover, $\zeta_i[k]$ is monotonically increasing function of $\Delta E_i^{\text{dem}}[k]/(k_i^{\text{dep}} - k)$. Consequently

$$\rho_i[k] = 0.5 + \zeta_i[k] / \left(2 \max_{i \in \mathcal{C}[k]} \zeta_i[k] \right). \quad (8)$$

f_2, which penalizes the switch OFF of EVs, is expressed as

$$f_2(\Omega[k-1], \Omega[k], R[k], \hat{\mathcal{P}}[k]) = \sum_{i \in \mathcal{C}[k]} (1 - \omega_i[k]) \omega_i[k-1] \rho_i[k] \hat{P}_i^2[k]. \quad (9)$$

We multiply each term by $\rho_i[k]$ to enforce EVs with larger values to be switched OFF at last. We also multiply by $\omega_i[k-1]$ to exclude EVs that are switched OFF.

C. Fair Allocation of Charging Power

The aggregated power setpoint P^{req} must be allocated fairly among EVs. In order to anticipate the future information, we allocate the power by using ζ_i as a weight for EV i. To this end, at time k, we compute *reference powers*, $P_i^{\text{ref}}[k] \in [0, P_i^{\max}]$ for all EVs, ideally fair such that $\sum_{i \in \mathcal{C}[k] \cup \mathcal{L}[k]} P_i^{\text{ref}}[k] = P^{\text{req}}[k]$. Commonly used fair allocations are *weighted-proportional* and *weighted max–min* [25]. It is sometimes difficult to choose between the two. In our specific case, we show that both are identical.

Let us first describe the weighted max–min fair allocation. As the set of constraints is convex and compact,[2] we know that

[1] $\epsilon = 100$ W is a pertinent safety margin.

[2] i.e., closed and bounded in Euclidean space.

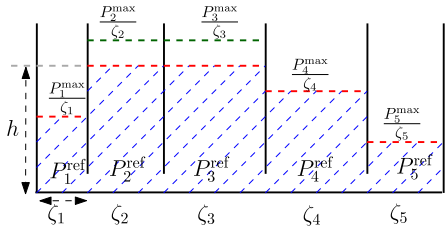


Fig. 3. Result of the water-filling algorithm for 5 EVs. EVs 1, 4, and 5 are fully filled, whereas 2 and 3 have reference powers of $h\zeta_2$ and $h\zeta_3$, respectively. The reference powers $P_1^{\text{ref}}, P_2^{\text{ref}}, P_3^{\text{ref}}, P_4^{\text{ref}}, P_5^{\text{ref}}$ are in kW.

this allocation exists and is unique [25]. In order to find such an allocation, the *water-filling* algorithm is used, which works as follows. The power of all EVs is increased at the same pace, until one or more powers reach their maximum. The powers that reach their maximum are frozen, and the others continue to increase at the same pace. The algorithm is repeated until $\sum_{i\in\mathcal{C}\cup\mathcal{L}} P_i^{\text{ref}} = P^{\text{req}}$ (henceforth, the time index k is omitted for simplicity of notation). For details, see Fig. 3. Here, we use again ζ_i to prioritize the EVs that need to be charged to satisfy their demand. Each EV i is represented as a water tank of width ζ_i and height $\frac{P_i^{\max}}{\zeta_i}$, the volume of the tank is $P_i^{\max}$. The volume of the water in tank is either $P_i^{\max}$ or $h\zeta_i$, where h is the common height of the nonsaturated tanks. Note that the EV with $\zeta_i = 0$ is charged fully and, therefore it no longer needs to be considered in the allocation.

Another possibility is to consider weighted-proportional fairness. We find a proportionally fair allocation of power by solving the following convex optimization problem in (**A**):

$$(\mathbf{A}) \qquad \max_{P_i^{\text{ref}}} \quad \sum_{i\in\mathcal{C}\cup\mathcal{L}} \zeta_i \log P_i^{\text{ref}}$$

$$\text{s.t.} \quad 0 < P_i^{\text{ref}} \le P_i^{\max} \tag{10}$$

$$\sum_{i\in\mathcal{C}\cup\mathcal{L}} P_i^{\text{ref}} = P^{\text{req}}. \tag{11}$$

We next prove that the above weighted-proportional and weighted max–min fair allocations are equivalent. This means that it is possible to find the fair allocation by either solving the optimization problem (**A**) or water filling.

Theorem 1: Weighted-proportional and weighted max–min fair allocations, as defined above, are equivalent.

Proof: Let P_i^{ref} be a weighted-proportional fair allocation. To compute this solution, we first get the Lagrangian of (**A**)

$$L(x,\eta) = -\sum_{i\in\mathcal{C}\cup\mathcal{L}} \zeta_i \log P_i^{\text{ref}} - \eta\left(\sum_{i\in\mathcal{C}\cup\mathcal{L}} P_i^{\text{ref}} - P^{\text{req}}\right) \tag{12}$$

where $0 < P_i^{\text{ref}} \le P_i^{\max}$ and η is the Lagrange multiplier. The solution of the problem is

$$P_i^{\text{ref}} = \begin{cases} \frac{\zeta_i}{\eta} & \text{if } \frac{\zeta_i}{\eta} \in [0, P_i^{\max}] \\ P_i^{\max} & \text{otherwise.} \end{cases} \tag{13}$$

Next, we show that the solution in (13) follows the definition of weighted max–min fairness in [25]. Mainly, we show that increasing one component i is at the expense of decreasing other component j such that $P_j^{\text{ref}}/\zeta_j \le P_i^{\text{ref}}/\zeta_i$. By (13), P_i^{ref} is either $\frac{\zeta_i}{\eta}$ or $P_i^{\max}$. If $P_i^{\text{ref}} = P_i^{\max}$, this component cannot be increased. Let us consider the case when $P_i^{\text{ref}} = \frac{\zeta_i}{\eta}$. If we increase the power of EV i, we have to decrease the power of at least one other EV, say j, to satisfy constraint (11). There are also two cases for EV j: either $P_j^{\text{ref}} = \frac{\zeta_j}{\eta}$ or $P_j^{\text{ref}} = P_j^{\max}$.

If $P_j^{\text{ref}} = \frac{\zeta_j}{\eta}$, then $\frac{P_j^{\text{ref}}}{\zeta_j} = \frac{P_i^{\text{ref}}}{\zeta_i} = \frac{1}{\eta}$. Hence, the inequality in weighted max–min fairness definition holds. If $P_j^{\text{ref}} = P_j^{\max}$, then $P_j^{\text{ref}} \le \frac{\zeta_j}{\eta}$. Dividing each part by ζ_j gives that: $\frac{P_j^{\text{ref}}}{\zeta_j} \le \frac{1}{\eta} = \frac{P_i^{\text{ref}}}{\zeta_i}$. Therefore, in both cases the inequality holds, proving that the two types of fairness give the same results.

Finally, after computing the fair allocation $P_i^{\text{ref}}[k]$, we construct, at every time step k, f_3 as follows:

$$f_3(\mathcal{P}[k]) = \sum_{i\in\mathcal{C}[k]} (P_i[k] - P_i^{\text{ref}}[k])^2. \tag{14}$$

D. Full Formulation

By combining (1), (4), (6), (9), and (14) with constraints (2) and (3), the optimization problem to be solved, at each time k, is

$$(\mathbf{P}) \quad \min_{P_i[k],\omega_i[k]} \quad c_0\left(\widetilde{P}^{\text{req}}[k] - \sum_{i\in\mathcal{C}[k]} P_i[k]\right)^2 \rightarrow \text{reference tracking}$$

$$\text{battery wear} \leftarrow \begin{cases} +c_1\left(\sum_{i\in\mathcal{C}[k]} (P_i[k] - \hat{P}_i[k])^2 \lambda_i[k] \right. \\ \left. + \sum_{i\in\mathcal{C}[k]} (1-\omega_i[k])\omega_i[k-1]\rho_i[k]\hat{P}_i^2[k]\right) \end{cases}$$

$$\text{fair allocation} \leftarrow \quad + \sum_{i\in\mathcal{C}[k]} (P_i[k] - P_i^{\text{ref}}[k])^2 \tag{15}$$

$$\text{s.t. (2)–(3).} \tag{16}$$

IV. Real-Time Implementation Aspects

A. Reducing the Number of Integer Variables

As (**P**) is mixed integer, its complexity grows exponentially with the number of integer variables [26] (here ω_i). To reduce the problem complexity, we propose a heuristic that runs at every time k and limits the number of integer variables. The heuristic partitions the collection of unlocked EVs, $\mathcal{C}[k]$, into three collections: EVs that are forced to be switched (or remain) ON ($\mathcal{S}^{\text{on}}[k]$), EVs that are forced to be switched (or remain) OFF ($\mathcal{S}^{\text{off}}[k]$), and EVs for which the ON/OFF decision is decided by the optimization problem ($\mathcal{S}[k]$). We require that $|\mathcal{S}[k]| \le m$, where m is fixed small number (e.g., $m \le 10$).

In other words, we define a new problem (**H**) that has at most m integer variables. All other $\omega_i[k]$ remain fixed

$$(\mathbf{H}) \quad \min_{P_i[k],\omega_i[k]} (15)$$

$$\text{s.t.} \quad (2), (3)$$

$$\omega_i[k] = 1 \quad \forall i \in \mathcal{S}^{\text{on}}[k], \qquad \omega_j[k] = 0 \quad \forall j \in \mathcal{S}^{\text{off}}[k]. \tag{17}$$

The constraints in (17) force the corresponding EVs to be switched ON/OFF.

Note that, with this consideration, the flexibility that the problem (**H**) considers is, however, smaller than that of (**P**). Specifically, the power to be allocated among the unlocked EVs, $\widetilde{P}^{\mathrm{req}}[k]$, might not be able to be tracked, depending on the partition of $\mathcal{C}$. Let us thus define the full flexibility of the CS at time k, as $\mathcal{F}$ (see Section V-B), and the reduced flexibility (the one available for (**H**)), as the interval $[P^{\mathrm{lb}}, P^{\mathrm{ub}}]$ with

$$P^{\mathrm{lb}} = \sum_{i \in \mathcal{S}^{\mathrm{on}}[k]} P_i^{\min}, \quad P^{\mathrm{ub}} = \sum_{i \in \mathcal{S}^{\mathrm{on}}[k] \cup \mathcal{S}[k]} P_i^{\max}. \tag{18}$$

Thus, the partition $\{\mathcal{S}[k], \mathcal{S}^{\mathrm{on}}[k], \mathcal{S}^{\mathrm{off}}[k]\}$ should ensure that $\widetilde{P}^{\mathrm{req}}[k] \in [P^{\mathrm{lb}}, P^{\mathrm{ub}}]$. Note that we compute the bounds excluding locked EVs. Their power is already defined, as explained in Section III-C.

Algorithm 1: Heuristic for Partitioning $\mathcal{C}[k]$.

Input: $\mathcal{C}[k]$, $m \geq 1$, $0 \leq \widetilde{P}^{\mathrm{req}}[k] \leq \sum_{i \in \mathcal{C}[k]} P_i^{\max}$
Output: partition $\mathcal{S}[k], \mathcal{S}^{\mathrm{on}}[k], \mathcal{S}^{\mathrm{off}}[k]$ of $\mathcal{C}[k]$, such that $|\mathcal{S}[k]| \leq m$ and $\widetilde{P}^{\mathrm{req}}[k] \in [P^{\mathrm{lb}}, P^{\mathrm{ub}}]$ computed in (18)

1: **if** $|\mathcal{C}[k]| \leq m$ **then**
2: $\mathcal{S}[k] = \mathcal{C}[k]$, $\mathcal{S}^{\mathrm{on}}[k] = \mathcal{S}^{\mathrm{off}}[k] = \emptyset$,
3: stop algorithm.
4: **else**
5: $\mathcal{S}[k] = \mathrm{top}(\mathcal{C}[k], m)$,
6: $\mathcal{S}^{\mathrm{on}}[k] = \{i \mid i \in \mathcal{R}, \omega_i[k-1] = 1\}$,
7: $\mathcal{S}^{\mathrm{off}}[k] = \{i \mid i \in \mathcal{R}, \omega_i[k-1] = 0\}$,
8: let $\mathcal{R} = \mathcal{C}[k] \setminus \mathcal{S}[k]$.
9: **end if**
10: compute reduced flexibility bounds as in (18).
11: **while** $\widetilde{P}^{\mathrm{req}}[k] \notin [P^{\mathrm{lb}}, P^{\mathrm{ub}}]$ **and** $\mathcal{R} \neq \emptyset$ **do**
12: $i = \mathrm{top}(\mathcal{R}, 1)$, $j = \mathrm{top}(\mathcal{S}[k], 1)$
13: **if** $\widetilde{P}^{\mathrm{req}}[k] > P^{\mathrm{ub}}$ **then**
14: $\mathcal{S}^{\mathrm{on}}[k] = \mathcal{S}^{\mathrm{on}}[k] \cup \{j\}$
15: **else if** $\widetilde{P}^{\mathrm{req}}[k] < P^{\mathrm{lb}}$ **then**
16: $\mathcal{S}^{\mathrm{off}}[k] = \mathcal{S}^{\mathrm{off}}[k] \cup \{j\}$
17: **end if**
18: update $\mathcal{S}[k] = \mathcal{S}[k] \cup \{i\} \setminus \{j\}$
19: **if** $i \in \mathcal{S}^{\mathrm{on}}[k]$ **then**
20: remove i from $\mathcal{S}^{\mathrm{on}}[k]$
21: **else**
22: remove i from $\mathcal{S}^{\mathrm{off}}[k]$
23: **end if**
24: recompute bounds according to (18)
25: update $\mathcal{R} = \mathcal{R} \setminus \{i\}$
26: **end while**

We now describe the heuristic, detailed in Algorithm 1. First, we define a metric that takes into account both the past behavior of the EVs power and their desire to be charged as follows:

$$\mu_i[k] = \lambda_i[k] + (1 - \omega_i[k-1])(1.5 - \rho_i[k]) + \omega_i[k-1]\rho_i[k] \tag{19}$$

with $\mu_i[k] \in [1, 2]$, unitless, and consisting of the following three parts.

1) $\lambda_i[k]$ contains information about the past behavior of the charging power. Smaller $\lambda_i[k]$ means that EV i is more likely to change its power.
2) $(1 - \omega_i[k-1])(1.5 - \rho_i[k])$ identifies the propensity of a switched-OFF EV to switch ON.
3) $\omega_i[k-1]\rho_i[k]$ identifies the propensity of a switched-ON EV to switch OFF.

Therefore, $\mu_i[k]$ quantifies the propensity of EV i to change its ON/OFF decision and charging power. Smaller $\mu_i[k]$ indicates more propensity.

Second, we rank the EVs according to their individual operational margins. As the maximum power an EV i can consume is $P_i^{\max}$ and the minimum is 0, its positive margin is $P_i^{\max} - \hat{P}_i[k]$ and its negative margin is $\hat{P}_i[k]$. By dividing these values by $P_i^{\max}$, we obtain normalized margins. Therefore, we introduce the ranking metric $r_i[k]$, which combines the operational margins with $\mu_i[k]$

$$r_i[k] = \begin{cases} \frac{\hat{P}_i[k]}{P_i^{\max}\mu_i[k]} & \text{if } \Delta P^{\mathrm{req}}[k] < 0 \\ \frac{P_i^{\max} - \hat{P}_i[k]}{P_i^{\max}\mu_i[k]} & \text{otherwise} \end{cases} \tag{20}$$

where $\Delta P^{\mathrm{req}}[k] = \widetilde{P}^{\mathrm{req}}[k] - \sum_{i \in \mathcal{C}[k]} \hat{P}_i[k]$. Finally, we define the function $\mathrm{top}(\mathcal{X}, m)$ that returns the index of the m elements with the largest $r_i[k]$ metric, from a collection $\mathcal{X}$. In the rest of this section, for sake of clarity, we omit the time index k.

The purpose of the heuristic is to limit the number of integer variables to m. If the amount of unlocked EVs is initially less than m, then all these EVs can change their ON/OFF decision (lines 2–3). Otherwise, we take the m EVs with the largest metric r_i (lines 5–8). This choice is sufficient in most of the cases as, according to r_i, these EVs are the best to be selected. However, it can happen that $\widetilde{P}^{\mathrm{req}} \notin [P^{\mathrm{lb}}, P^{\mathrm{ub}}]$. In which case, we loop until fulfilling this constraint. If $\widetilde{P}^{\mathrm{req}}$ lies above the bounds, we force the EV from $\mathcal{S}$ with highest rank to be switched ON and replace it with the highest ranked EV in $\mathcal{R} = \mathcal{C} \setminus \mathcal{S}$ (lines 12–14, 18–19). Doing this, we automatically increase P^{ub}, eventually reaching $\widetilde{P}^{\mathrm{req}}$ (see Theorem 2). Similarly, if $\widetilde{P}^{\mathrm{req}}$ lies below the bounds, we switch OFF the highest ranked EV from $\mathcal{S}$ (lines 15–17) and replace it with the highest ranked EV in $\mathcal{R}$.

Theorem 2. (Correctness of the heuristic): Given that $m \geq 1$, Algorithm 1 finds a partition $\mathcal{S}, \mathcal{S}^{\mathrm{on}}, \mathcal{S}^{\mathrm{off}}$ of $\mathcal{C}$, such that $\widetilde{P}^{\mathrm{req}} \in [P^{\mathrm{lb}}, P^{\mathrm{ub}}]$, $|\mathcal{S}| \leq m$. Algorithm 1 takes at most $|\mathcal{C}| - m$ iterations.

Proof: Let ℓ be the number of iterations in the while loop between lines 11–26. $\ell = 0$, when the while loop is not executed. If Algorithm 1 enters the loop, we increment ℓ just before executing line 12. Denote $\mathcal{S}^{(\ell)}, \mathcal{S}^{\mathrm{on},(\ell)}, \mathcal{S}^{\mathrm{off},(\ell)}, \mathcal{R}^{(\ell)}$ the state of collections and $[P^{\mathrm{lb},(\ell)}, P^{\mathrm{ub},(\ell)}]$ the state of bounds, at the end of the ℓth iteration. $\mathcal{S}^{(0)}, \mathcal{S}^{\mathrm{on},(0)}, \mathcal{S}^{\mathrm{off},(0)}, \mathcal{R}^{(0)}, [P^{\mathrm{lb},(0)}, P^{\mathrm{ub},(0)}]$ are initial states of collections and bounds.

We now prove the following five statements (S1–S5).

(S1) For every $\ell \geq 0$

$$\mathcal{R}^{(\ell)} \subseteq \mathcal{S}^{\mathrm{on},(\ell)} \cup \mathcal{S}^{\mathrm{off},(\ell)}. \tag{21}$$

Indeed, for $\ell = 0$, (21) holds by lines 7 and 8. Assume that (21) holds at $\ell - 1$. We remove one element from $\mathcal{R}^{(\ell-1)}$ and either

from $\mathcal{S}^{\text{on},(\ell-1)}$ or $\mathcal{S}^{\text{off},(\ell-1)}$ (lines 20 and 22). Thus, (21) holds at ℓ.

(S2) For every $\ell \geq 0$, $\mathcal{S}^{(\ell)}, \mathcal{S}^{\text{on},(\ell)}, \mathcal{S}^{\text{off},(\ell)}$ is partition of $\mathcal{C}$

$$\mathcal{C} = \mathcal{S}^{(\ell)} \cup \mathcal{S}^{\text{on},(\ell)} \cup \mathcal{S}^{\text{off},(\ell)} \tag{22}$$

$$\mathcal{S}^{(\ell)} \cap (\mathcal{S}^{\text{on},(\ell)} \cup \mathcal{S}^{\text{off},(\ell)}) = \emptyset \tag{23}$$

$$\mathcal{S}^{\text{on},(\ell)} \cap \mathcal{S}^{\text{off},(\ell)} = \emptyset. \tag{24}$$

Indeed, for $\ell = 0$, $\mathcal{S}^{(0)}, \mathcal{S}^{\text{on},(0)}, \mathcal{S}^{\text{off},(0)}$ is a partition of $\mathcal{C}$ by construction. Assume that (22)–(24) hold at $(\ell-1)$. Then, $\mathcal{S}^{(\ell)} = \mathcal{S}^{(\ell-1)} \cup \{i\} \setminus \{j\}$ (line 18) and $\mathcal{S}^{\text{off},(\ell)} \cup \mathcal{S}^{\text{on},(\ell)} = \mathcal{S}^{\text{off},(\ell-1)} \cup \mathcal{S}^{\text{on},(\ell-1)} \cup \{j\} \setminus \{i\}$ (lines 13–17, 19–23). Since $j \in \mathcal{S}^{(\ell-1)}$ and, by (21), $i \in \mathcal{S}^{\text{on},(\ell-1)} \cup \mathcal{S}^{\text{off},(\ell-1)}$ (22) and (23) holds at ℓ. Also, we take j from $\mathcal{S}^{(\ell-1)}$ and move it either to $\mathcal{S}^{\text{on},(\ell-1)}$ (line 14) or $\mathcal{S}^{\text{off},(\ell-1)}$ (line 16), then (24) holds at ℓ.

(S3) If Algorithm 1 visits line 14 at iteration ℓ, then $P^{\text{ub},(\ell-1)} \in [P^{\text{lb},(\ell)}, P^{\text{ub},(\ell)}]$ and, if it visits line 16 at iteration ℓ, then $P^{\text{lb},(\ell-1)} \in [P^{\text{lb},(\ell)}, P^{\text{ub},(\ell)}]$.

Let us show (S3) in the first case, i.e., that $P^{\text{ub},(\ell-1)} \leq P^{\text{ub},(\ell)}$ and $P^{\text{ub},(\ell-1)} \geq P^{\text{lb},(\ell)}$. To this end, let us first check how $\mathcal{S}^{\text{on},(\ell-1)}$ and $\mathcal{S}^{(\ell-1)}$ are changed by Algorithm 1. We take i from $\mathcal{R}^{(\ell-1)}$ (line 12) and add it to $\mathcal{S}^{(\ell-1)}$. Also, we move j from $\mathcal{S}^{(\ell-1)}$ to $\mathcal{S}^{\text{on},(\ell-1)}$. Then, we have two cases, since i can be either from $\mathcal{S}^{\text{on},(\ell-1)}$ or $\mathcal{S}^{\text{off},(\ell-1)}$ (see (21) and (24)). If $i \in \mathcal{S}^{\text{on},(\ell-1)}$, then $\mathcal{S}^{\text{on},(\ell)} = \mathcal{S}^{\text{on},(\ell-1)} \cup \{j\} \setminus \{i\}$ and $\mathcal{S}^{(\ell)} = \mathcal{S}^{(\ell-1)} \cup \{i\} \setminus \{j\}$, hence $\mathcal{S}^{(\ell)} \cup \mathcal{S}^{\text{on},(\ell)} = \mathcal{S}^{(\ell-1)} \cup \mathcal{S}^{\text{on},(\ell-1)}$. According to (18), $P^{\text{ub},(\ell-1)} = P^{\text{ub},(\ell)}$. Rewrite expressions in (18) for $P^{\text{ub},(\ell-1)}$ and $P^{\text{lb},(\ell)}$

$$P^{\text{ub},(\ell-1)} = \sum_{j' \in \mathcal{S}^{\text{on},(\ell-1)}} P^{\max}_{j'} + \sum_{j' \in \mathcal{S}^{(\ell-1)} \setminus \{j\}} P^{\max}_{j'} + P^{\max}_{j} \tag{25}$$

$$P^{\text{lb},(\ell)} = \sum_{j' \in \mathcal{S}^{\text{on},(\ell-1)} \cup \{j\} \setminus \{i\}} P^{\min}_{j'} = P^{\text{lb},(\ell-1)} - P^{\min}_{i} + P^{\min}_{j}. \tag{26}$$

By inspecting (25) and (26) term by term, we see that each term in (25) is not smaller than the corresponding term in (26).

If $i \in \mathcal{S}^{\text{off},(\ell-1)}$ then $\mathcal{S}^{\text{on},(\ell)} = \mathcal{S}^{\text{on},(\ell-1)} \cup \{j\}$ and $\mathcal{S}^{(\ell)} \cup \mathcal{S}^{\text{on},(\ell)} = \mathcal{S}^{(\ell-1)} \cup \mathcal{S}^{\text{on},(\ell-1)} \cup \{i\}$. By (18) $P^{\text{ub},(\ell-1)} \leq P^{\text{ub},(\ell)}$

$$P^{\text{lb},(\ell)} = \sum_{j' \in \mathcal{S}^{\text{on},(\ell-1)} \cup \{j\}} P^{\min}_{j'} = P^{\text{lb},(\ell-1)} + P^{\min}_{j}. \tag{27}$$

Comparing (25) and (27), we get $P^{\text{ub},(\ell-1)} \geq P^{\text{lb},(\ell)}$, which concludes the proof. The proof is similar in the case when algorithm visits line 16.

(S4) If Algorithm 1 visits line 14, it will never visit line 16 and vice versa.

Assume that Algorithm 1 visits line 14 at iteration $\ell - 1$. From (S3) we know that $P^{\text{lb},(\ell-1)} \in [P^{\text{lb},(\ell)}, P^{\text{ub},(\ell)}]$. We will prove (S4) by contradiction. Assume that Algorithm 1 visits line 16 at ℓ, i.e., $\widetilde{P}^{\text{req}} < P^{\text{lb},(\ell)}$. Then, $P^{\text{ub},(\ell-1)} < \widetilde{P}^{\text{req}} < P^{\text{lb},(\ell)}$ [see Fig. 4(a)]. This is impossible because $P^{\text{lb},(\ell)} \leq P^{\text{ub},(\ell-1)}$ according to (S3). Similarly, the case for line 16 can be proven [Fig. 4(b)].

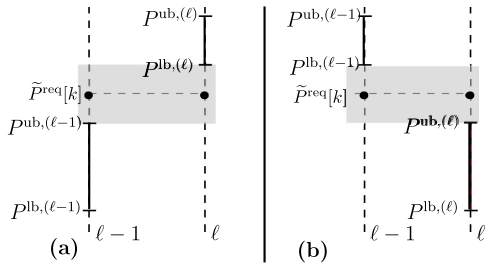


Fig. 4. Proof of statement (S4).

(S5) If Algorithm 1 terminates with empty $\mathcal{R}^{(\ell)}$, then $\widetilde{P}^{\text{req}} \in [P^{\text{lb},(\ell)}, P^{\text{ub},(\ell)}]$.

Let us prove (S5) when algorithm visits line 14 (similar for line 16). We show that $\mathcal{S}^{\text{off},(\ell)} \subset \mathcal{R}^{(\ell)}$ at every iteration ℓ. If $\ell = 0$ it is true by construction. Let us assume that $\mathcal{S}^{\text{off},(\ell-1)} \subset \mathcal{R}^{(\ell-1)}$ at $\ell - 1$. According to (S4), if Algorithm 1 visits line 14 it will never visit line 16, so no new elements are added to $\mathcal{S}^{\text{off},(\ell-1)}$. Next, if $i \in \mathcal{S}^{\text{on},(\ell-1)}$, $\mathcal{S}^{\text{off},(\ell)} = \mathcal{S}^{\text{off},(\ell-1)}$, else (i.e., if $i \in \mathcal{S}^{\text{off},(\ell-1)}$), $\mathcal{S}^{\text{off},(\ell)} = \mathcal{S}^{\text{off},(\ell-1)} \setminus \{i\}$. Also $\mathcal{R}^{(\ell)} = \mathcal{R}^{(\ell-1)} \setminus \{i\}$. Thus, $\mathcal{S}^{\text{off},(\ell)} \subset \mathcal{R}^{(\ell)}$. Let L be the value ofℓ when Algorithm 1 terminates and $\mathcal{R}^{(L)} = \emptyset$, therefore $\mathcal{S}^{\text{off},(L)} = \emptyset$, thus $\mathcal{C} = \mathcal{S}^{\text{on},(L)} \cup \mathcal{S}^{(L)}$ and $P^{ub,(L)} = \sum_{i \in \mathcal{C}} P^{\max}_{i}$. According to the input conditions $\widetilde{P}^{\text{req}} \leq P^{ub,(L)}$. Let us show that $\widetilde{P}^{\text{req}} \geq P^{lb,(L)}$. Indeed, according to (S4) Algorithm 1 visited line 14 on iteration $L-1$, therefore by (S3) $P^{lb,(L)} \leq P^{ub,(L-1)} < \widetilde{P}^{\text{req}}$.

We can now complete the proof by observing that Algorithm 1 terminates either when $\widetilde{P}^{\text{req}} \in [P^{\text{lb},(\ell)}, P^{\text{ub},(\ell)}]$ or because $\mathcal{R}^{\ell} = \emptyset$. However, by (S5) we also have that $\widetilde{P}^{\text{req}} \in [P^{\text{lb},(\ell)}, P^{\text{ub},(\ell)}]$. Moreover, Algorithm 1 removes one element from $\mathcal{R}$ at every iteration. The initial amount of elements in $\mathcal{R}$ is $|\mathcal{C}| - m$, then it will terminate in at most $|\mathcal{C}| - m$ iterations.

After constructing $\{\mathcal{S}, \mathcal{S}^{\text{on}}, \mathcal{S}^{\text{off}}\}$, we solve problem (**H**) by using branch-and-bound [26].

V. Validation

A. Simulation Setup

To validate our method, we consider a grid with an existing 500 kWp PV plant connected to the distribution network through a power transformer rated $S^{r}_{\text{Tr}} = 500$ kVA.[3] The CS has a power rating $P^{r}_{\text{CS}} = 1000$ kW; this is a stress test for our method, as the CS has to opportunistically use all available power. The CS has 60 slots of 22 kW max.

To simulate the PV production, we use real irradiance measurements taken in our laboratory; they are then normalized according to the PV rated power. We simulate the arrival of EVs to the CS as a homogeneous Poisson process with a rate of 30 arrivals/h [see Fig. 5(d)]. We assume that, upon arrival (at k^{arr}_{i}), EV i informs the CS of its energy demand ΔE_i and the expected departure time k^{dep}_{i}. We model the expected staying time $(k^{\text{dep}}_{i} - k^{\text{arr}}_{i})$ to be uniformly distributed between 1.5 and 1.6 h. This assumption has been made only for simulation purposes and

[3]Note that we do not consider grid constraints other than the transformer rated power.

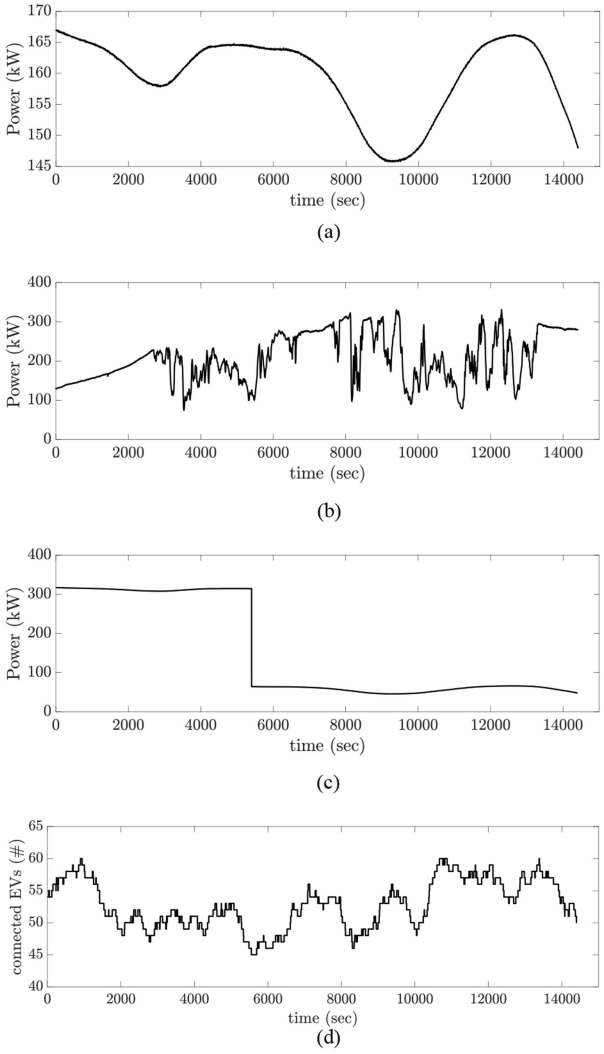


Fig. 5. PV production scenarios and the number of EVs. (a) PV production for regular production. (b) PV production for fluctuating production. (c) PV production for sharp jump. (d) Number of connected EVs at the CS.

serves to model the behavior of the different users. Furthermore, we consider that an EV can leave before the informed time. Thus, the real staying time is uniformly distributed between 1.4 and 1.5 h in our simulation. An EV will leave after the real staying time, regardless of its level of charge. The proposed control scheme does not have access to the information about distributions of expected staying time and real staying time; it uses only the exact value of expected staying time, declared by users. Given the distributions of the arrival time and the staying time, it is highly likely that an EV will find an available slot upon arrival, otherwise this EV is ignored (as in practice this EV will leave for another CS). In all our simulation scenarios, this property was maintained.

We consider two groups of EVs: group A with high and group B with low energy demand. The demand is uniformly distributed between 28 and 32 kWh and between 10 and 14 kWh, respectively. Considered reaction times are also uniformly distributed between 2 and 3 s and the ramping rate is 5 kW/s.[4]

[4]This rate was taken according to the maximum charging power such that the EV will reach its maximum power before the locking period finishes.

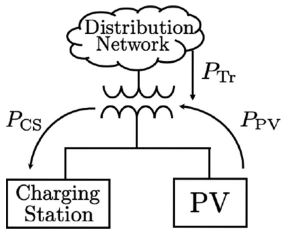


Fig. 6. Structure of the grid. The arrows show the positive directions of the corresponding active-power flows.

We also ran simulations that consider different ramping rates depending on each group: we set 5 kW/s for group A and 15 kW/s for group B. The obtained results show little difference to the results presented in Section V-F, thus ensuring that our method is robust against the change in ramping rates. As these results are almost identical, they are not included in the article. The minimum and maximum powers of the modeled EVs are 2 and 22 kW.

We consider three scenarios, mainly defined by the PV trace, that are representative enough to show all our method features:

1) *regular production*, when the PV production is smooth [see Fig. 5(a)];
2) *fluctuating production*, when the PV production has high-frequency fluctuations [see Fig. 5(b)], e.g., due to clouds;
3) *sharp jump*, when, for emergency reasons, part of the PV plant is suddenly disconnected [see Fig. 5(c)].

Finally, we analyze the influence of the combinations of weights c_0, c_1 in problem (**P**) on the method performance.

B. Model of the Grid Controller

Next, we describe the way we model the decision of the grid controller at time k. We assume that all resources are connected to the same node, thus simplifying the power-balance equation to $P_{\mathrm{Tr}} = P_{\mathrm{CS}} - P_{\mathrm{PV}}$, being P_{Tr} the transformer, P_{PV} the PV plant and P_{CS} the CS powers, respectively (see Fig. 6 for the powers direction convention). The control variable is P_{CS}, whereas the controlled variable is P_{Tr}. The purpose of this controller is to maximize P_{CS}, and to avoid the violation of the transformer rated power, i.e., $|P_{\mathrm{Tr}}| \leq S^r_{\mathrm{Tr}}$, subject to the uncertainty produced by 1) the variation of the injected PV power and 2) the charging of locked EVs. We focus on the case when the violation is produced by an overconsumption of the CS. The case when the violation is produced by an overproduction of the PV plant can be handled similarly. Hence, the controller's decision is computed as

$$P^{\mathrm{req}} = P^r_{\mathrm{Tr}} + P^{\downarrow}_{\mathrm{PV}} - \Delta P^{\uparrow}_{\mathrm{CS}} \tag{28}$$

where $P^{\downarrow}_{\mathrm{PV}}$ is the one-step-ahead minimum expected PV production, computed by a short-term forecasting tool [11]. $\Delta P^{\uparrow}_{\mathrm{CS}}$ is the maximum possible consumption increment of locked EVs, i.e., the difference between their individual setpoint and their current measured power

$$\Delta P^{\uparrow}_{\mathrm{CS}} = \sum_{i \in \mathcal{L}^{\uparrow}} P_i - \hat{P}_i, \quad \mathcal{L}^{\uparrow} = \{i \in \mathcal{L} | P_i - \hat{P}_i \geq 0\}. \tag{29}$$

This term accounts for the uncertainty of EVs at implementing a setpoint due to the unknown ramping properties of each EV.

Finally, the computed setpoint is saturated, depending on the current flexibility of the CS, and it is computed by the CS and sent to the grid controller, represented by the interval

$$\mathcal{F} = \left[\sum_{i\in\mathcal{L}} P_i, \min\left(\sum_{i\in\mathcal{L}} P_i + \sum_{i\in\mathcal{C}} P_i^{\max}, P_{\mathrm{CS}}^r\right)\right]. \quad (30)$$

It is worth noting that the flexibility is lower bounded by the locked EVs and upper bounded by the maximum power of the unlocked EVs. The flexibility is not limited by the minimum power and the handling of any setpoint below $\min_i P_i^{\min}$ is ensured by Theorem 2. Besides, the controller cannot instantly ensure that the transformer rated power will not be violated due to the ramping mechanism of the locked EVs but, in the worst-case scenario, it will take a time T^L (locking period) to regain more flexibility, thus decreasing the consumption.

C. Performance Evaluation Metrics

As our optimization problem in (**H**) is multiobjective, we define the following metrics for the performance evaluation.

1) *Follow request*—measures how well a CS follows the aggregated power setpoint

$$M^{\mathrm{fr}} = \frac{1}{K}\sum_{k=1}^{K}\left|P^{\mathrm{req}}[k] - \hat{P}[k]\right| \quad (31)$$

where K is the amount of discrete time steps during the selected control period and $\hat{P}[k] = \sum_{i\in\mathcal{C}[k]\cup\mathcal{L}[k]} \hat{P}_i[k]$. This metric is lower bounded by 0. Then, the closer M^{fr} is to 0, the better the CS follows the aggregated setpoint.

2) *Nonsatisfied demand*—measures how well the charging demand of EV i is satisfied

$$M_i^{\mathrm{nsd}} = \Delta E_i[k_i^{\mathrm{stop}}]/\Delta E_i[k_i^{\mathrm{arr}}] \quad (32)$$

where $\Delta E_i[k_i^{\mathrm{stop}}]$ is the energy that remains to be satisfied at departure time, and $\Delta E_i[k_i^{\mathrm{arr}}]$ is the initial energy demand. $M_i^{\mathrm{nsd}} \in [0,1]$. If $M_i^{\mathrm{nsd}} = 1$, then EV i did not charge. On contrary, EV i is fully satisfied if $M_i^{\mathrm{nsd}} = 0$.

3) *Battery wear*—measures the changes of the charge power

$$M_i^{\mathrm{bw}} = \frac{1}{2(P_i^{\max})^2}\sum_{k=1}^{K}(P_i[k] - P_i[k-1])^2. \quad (33)$$

This metric shows the effect of the control scheme into the battery life. The closer M_i^{bw} is to 0, the less effect is. If $M_i^{\mathrm{bw}} < 1$, there was no sharp jump of charging power from 0 to $P_i^{\max}$ and back.

4) *Violation*—measures the maximum per-unit operating (hot-spot) temperature of the transformer

$$M^{\mathrm{viol}} = \max_k\left\{\frac{\theta[k]}{\theta^{\mathrm{rated}}}\right\} \quad (34)$$

where $\theta[k]$ and θ^{rated} are the hot-spot and rated hot-spot transformer temperatures,[5] and $\theta^{\mathrm{rated}} = 160\,^{\circ}\mathrm{C}$. As a transformer can be overloaded for short time, this metric shows how the method can exploit this flexibility by respecting the transformer physical limits.

[5]We model the dynamic behavior of the hot-spot temperature $\theta[k]$ as a first-order dynamic system (see [27] and [28]): $\theta[k+1] = \alpha\theta[k] + \beta P_{\mathrm{Tr}}[k] + \gamma\theta^{\mathrm{amb}}[k]$, where θ^{amb} is the ambient temperature. The parameters α, β, and γ are estimated using the curves from [28].

D. Congestion Indication

As the amount of EVs connected to the CS varies in time, this influences the possibility of the CS satisfying the demand of EVs. For instance, there might be many EVs charging at the same time and that have a large energy demand (i.e., congested period). The M_i^{nsd} depends on whether CS is congested or not. If the CS is noncongested, we expect that M_i^{nsd} is close to 0 for all i. Conversely, if $M_i^{\mathrm{nsd}} > 0$, CS is congested. In order to evaluate our method, we define a *congestion* indicator I^{con}. Specifically, I^{con} is an average nonsatisfied demand, assuming that the CS is absolutely fair and attempts to charge every EV at constant power $P_i^{\mathrm{avg}}[k] = \Delta E_i^{\mathrm{dem}}[k_i^{\mathrm{arr}}]/(k_i^{\mathrm{dep}} - k_i^{\mathrm{arr}})$. We compute I^{con} as follows:

$$I^{\mathrm{con}} = \frac{\sum_{k=1}^{K}(P^{\mathrm{avg}}[k] - P^{\mathrm{avail}}[k])\mathbb{1}_{P^{\mathrm{avg}}[k]\geq P^{\mathrm{avail}}[k]}}{\sum_{k=1}^{K} P^{\mathrm{avg}}[k]} \quad (35)$$

where $P^{\mathrm{avail}}[k] = P_{\mathrm{Tr}}^r + P_{\mathrm{PV}}[k]$ is the available power that can be consumed by the CS at time k, and $P^{\mathrm{avg}}[k] = \sum_{i\in\mathcal{C}[k]\cup\mathcal{L}[k]} P_i^{\mathrm{avg}}[k]$. If $I^{\mathrm{con}} \leq 0$, there is no congestion at CS and all EVs should be satisfied. On the contrary, if I^{con} is 1, it means that there is no available power at all.

E. Results for the Default Weight Combination

In this section, we show the results of our method for the default weight combination ($c_0 = 1, c_1 = 1$). We show maximum battery wear ($\max_{i,A}(M_i^{\mathrm{bw}})$, $\max_{i,B}(M_i^{\mathrm{bw}})$), standard deviation ($\sigma_A(M_i^{\mathrm{nsd}})$, $\sigma_B(M_i^{\mathrm{nsd}})$) and mean value ($\mu_A(M_i^{\mathrm{nsd}})$, $\mu_B(M_i^{\mathrm{nsd}})$) of nonsatisfied demand per group A and B. We also show I^{con}, M^{viol}, M^{fr} and an average aggregated power setpoint $\bar{P}^{\mathrm{req}}$. The results for the scenarios are shown in the following.

metrics	regular production	fluctuating production	sharp jump
$\max_{i,A}(M_i^{\mathrm{bw}})$ (p.u.)	0.19	0.34	0.47
$\max_{i,B}(M_i^{\mathrm{bw}})$(p.u.)	0.11	0.44	0.23
$\mu_A(M_i^{\mathrm{nsd}})$ (p.u.)	0.28	0.14	0.45
$\sigma_A(M_i^{\mathrm{nsd}})$ (p.u.)	0.03	0.04	0.03
$\mu_B(M_i^{\mathrm{nsd}})$ (p.u.)	0.27	0.12	0.44
$\sigma_B(M_i^{\mathrm{nsd}})$ (p.u.)	0.03	0.03	0.04
I^{con} (p.u.)	0.26	0.12	0.42
M^{fr} (kW)	0.51	2.61	0.35
M^{viol} (p.u.)	0.52	0.53	0.51
$\bar{P}^{\mathrm{req}}$ (kW)	607	709	369

We see that for all scenarios the $\mu_A(M_i^{\mathrm{nsd}})$ and $\mu_B(M_i^{\mathrm{nsd}})$ are close to I^{con}, $\sigma_A(M_i^{\mathrm{nsd}})$ and $\sigma_B(M_i^{\mathrm{nsd}})$ are less than 0.04. This means that our method efficiently and fairly allocates the power among EVs. The battery wear is also less than 1, meaning that there were no large jumps of the charging power. Also, the CS correctly tracks an aggregated power setpoint since M^{fr} is small compared to the $\bar{P}^{\mathrm{req}}$. Finally, the M^{viol} is around 0.5 for all scenarios, i.e., we never overheat the transformer. The results

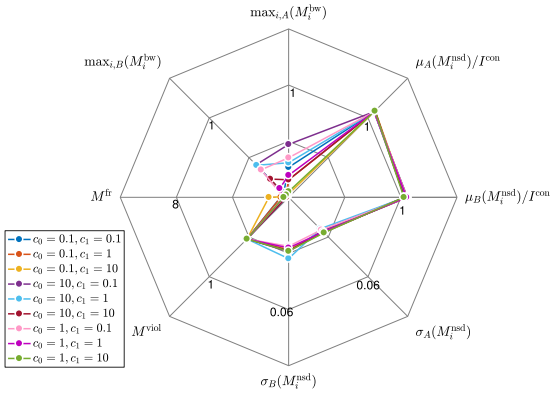


Fig. 7. Metrics for the regular production.

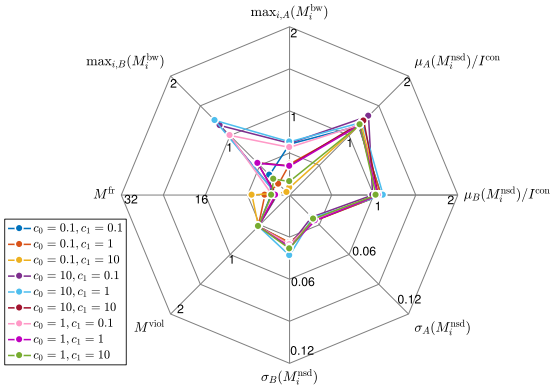


Fig. 8. Metrics for the fluctuating production.

show that our method has good performance in all scenarios with the default combination of weights.

F. Results for the Different Weight Combinations

We studied the behavior of our method for weights by taking values from the set $\{0.01, 0.1, 1, 10, 100\}$. This gives us 25 combinations, but we show only the results of nine of them, as the other combinations have significantly worse performances. For a better visualization, we show the results on eight-dimensional spider plots. We show violation M^{fr}, M^{viol}, maximum battery wear ($\max_{i,A}(M_i^{\mathrm{bw}})$, $\max_{i,B}(M_i^{\mathrm{bw}})$), standard deviation ($\sigma_A(M_i^{\mathrm{nsd}})$, $\sigma_B(M_i^{\mathrm{nsd}})$) and mean value ($\mu_A(M_i^{\mathrm{nsd}})/I^{\mathrm{con}}$, $\mu_B(M_i^{\mathrm{nsd}})/I^{\mathrm{con}}$) of nonsatisfied demand per group A and B.

Fig. 7 shows the results for the regular production. We see that, for all combinations of weights, the power allocated fairly among EVs, as $\mu_A(M_i^{\mathrm{nsd}})/I^{\mathrm{con}}$ and $\mu_B(M_i^{\mathrm{nsd}})/I^{\mathrm{con}}$ are 1.07 and 1.03, respectively, whereas $\sigma_A(M_i^{\mathrm{nsd}})$ and $\sigma_B(M_i^{\mathrm{nsd}})$ are 0.03. Our method avoids large power jumps because the battery-wear metric never exceeds 1 for any weight combination. Overall, all weight combinations have similar performance, which shows that our method is not sensitive to the weight change in case of a smooth change of the power production.

Fig. 8 shows the results for the fluctuating production. We observe that combinations $(c_0 = 10, c_1 = 1)$, $(c_0 = 10, c_1 = 0.1)$ and $(c_0 = 1, c_1 = 0.1)$ have the worst battery-wear performance.

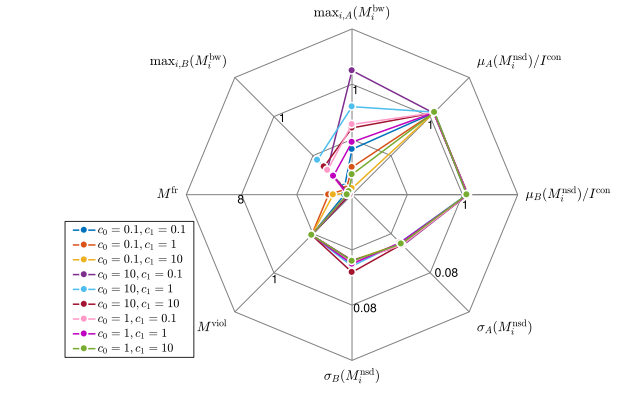


Fig. 9. Metrics for the sharp jump.

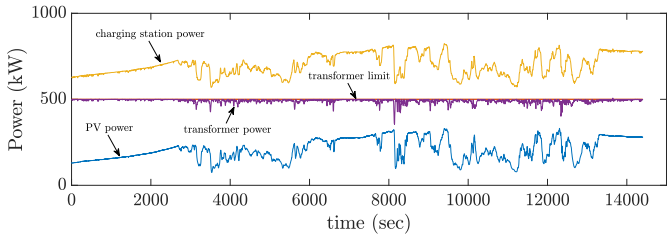


Fig. 10. Fluctuating production, $c_0 = 1, c_1 = 10$. This illustrates that CS can opportunistically use the fluctuated PV power production while respecting the transformer limit.

The combination $(c_0 = 10, c_1 = 0.1)$ has the worst nonsatisfied demand for group A. Other combinations show close and good general performance in all metrics. The best combination is $(c_0 = 1, c_1 = 10)$ in this scenario.

Fig. 9 shows the results for the sharp jump. We see that the combination $(c_0 = 10, c_1 = 0.1)$ has the worst battery-wear performance. The rest of the combinations have close performance in all metrics, which shows that our method is not sensitive to the weights and has a good performance if there is a sharp change of the power production. The best combination is $(c_0 = 1, c_1 = 10)$ in this scenario. We also observe that, for all weight combinations, M^{viol} metric is around 0.5, i.e., we never overheat the transformer. The method potentially leads to an overshooting of the rated power of the transformer, which rapidly increases its temperature. However, as we immediately react to such overshoots, the operating temperature never exceeds the safe limit.

The above analysis shows that the performance of the method is largely insensitive to the weights, and that the default combination $(c_0 = 1, c_1 = 1)$ works well in all scenarios. It also shows that increasing c_1, for example with $(c_0 = 1, c_1 = 10)$, leads to slightly better performance. As an example, we show in Fig. 10 power traces for the fluctuating production scenario for this combination. Additionally, we show in Fig. 11 the evolution of the power setpoint of two EVs: one from group A (with high energy demand) and one from group B (with low energy demand). There are no large fluctuations and minicycles can be seen, and the EV with higher energy demand receives more power according to the fair power allocation.

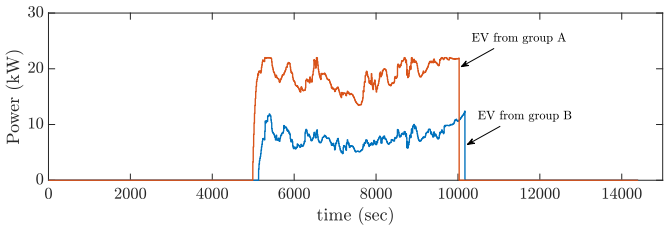


Fig. 11. Fluctuating production, $c_0 = 1, c_1 = 10$. Setpoints change for EVs from group A (red curve) and group B (blue curve).

VI. Conclusion

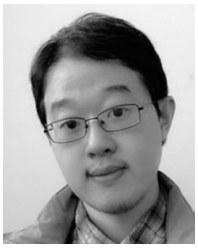
We have proposed a control scheme for controlling the charging of EVs connected to a single CS that follows an aggregated power setpoint in real time. When the CS tracks the aggregated power setpoint, the overall consumed power is allocated fairly among the connected EVs, and the effect on the battery life is minimized. Specifically, we have formulated a mixed-integer-quadratic program based on novel integral terms to cope with time-dependent variables such as battery wear and remaining energy demand. To reduce the problem complexity, we have also proposed a heuristic that reduces the number of integer variables enabling it to be solved in real time. We have evaluated our method in a stressed situation when the CS does not have enough power to charge all EVs at maximum. We have created three scenarios with different production traces and have demonstrated the performance based on rigorous metrics. The results show that the control scheme has the potential for a real-world application.

References

[1] D. Block, J. Harrison, and P. Brooker, "Electric vehicle sales for 2014 and future projections," Florida Solar Energy Center, Cocoa, FL, USA, Rep. FSEC-CR-1998-15, Mar. 2015.
[2] K. Clement, E. J. Haesen, and J. Driesen, "Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids," in *Proc. Power Syst. Conf. Expo.*, 2009, pp. 1–7.
[3] J. A. P. Lopes, F. J. Soares, and P. M. R. Almeida, "Integration of electric vehicles in the electric power system," *Proc. IEEE*, vol. 99, no. 1, pp. 168–183, Jan. 2011.
[4] G. A. Putrus, P. Suwanapingkarl, D. Johnston, E. C. Bentley, and M. Narayana, "Impact of electric vehicles on power distribution networks," in *Proc. IEEE Vehicle Power Propulsion Conf.*, Sep. 2009, pp. 827–831.
[5] P. B. Evans, S. Kuloor, and B. Kroposki, "Impacts of plug-in vehicles and distributed storage on electric power delivery networks," in *Proc. IEEE Vehicle Power Propulsion Conf.*, Sep. 2009, pp. 838–846.
[6] C. Dharmakeerthi, N. Mithulananthan, and T. Saha, "Impact of electric vehicle fast charging on power system voltage stability," *Elect. Power Energy Syst.*, vol. 57, pp. 241–249, 2014.
[7] J. Pillai and B. Bak-Jensen, "Impacts of electric vehicle loads on power distribution systems," in *Proc. IEEE Vehicle Power Propulsion Conf.*, 2010, pp. 1–6.
[8] L. Fernández, T. San Roman, R. Cossent, C. Domingo, and P. Frias, "Assessment of the impact of plug-in electric vehicles on distribution networks," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 206–213, Feb. 2011.
[9] S. Acha, T. Green, and N. Shah, "Effects of optimised plug-in hybrid vehicle charging strategies on electric distribution network losses," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo.*, Apr. 2010, pp. 1–6.
[10] E. Veldman and R. A. Verzijlbergh, "Distribution grid impacts of smart electric vehicle charging from different perspectives," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 333–342, Jan. 2015.
[11] E. Scolari, D. Torregrossa, J.-Y. Le Boudec, and M. Paolone, "Ultra-short-term prediction intervals of photovoltaic AC active power," in *Proc. Int. Conf. Probabilistic Methods Appl. Power Syst.*, Oct. 2016, pp. 1–8.
[12] A. Bernstein, L. Reyes-Chamorro, J.-Y. Le Boudec, and M. Paolone, "A composable method for real-time control of active distribution networks with explicit power set points. Part I: Framework," *Elect. Power Syst. Res.*, vol. 6, pp. 254–264, Aug. 2015.
[13] E. Sortomme and M. A. El-Sharkawi, "Optimal scheduling of vehicle-to-grid energy and ancillary services," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 351–359, Mar. 2012.
[14] M. Liu, P. McNamara, and S. McLoone, "Fair charging strategies for EVs connected to a low-voltage distribution network," in *Proc. IEEE PES Innov. Smart Grid Technol. Eur.*, 2013, pp. 1–5.
[15] S. Xie, W. Zhong, K. Xie, R. Yu, and Y. Zhang, "Fair energy scheduling for vehicle-to-grid networks using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1697–1707, Aug. 2016.
[16] S. Vandael, B. Claessens, M. Hommelberg, T. Holvoet, and G. Deconinck, "A scalable three-step approach for demand side management of plug-in hybrid vehicles," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 720–728, Jun. 2013.
[17] Y. Leehter, W. H. Lim, and T. S. Tsai, "A real-time charging scheme for demand response in electric vehicle parking station," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 52–62, Jan. 2017.
[18] S. Deilami, A. Masoum, P. Moses, and M. A. S. Masoum, "Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 456–467, Sep. 2011.
[19] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 67–78, Jan. 2013.
[20] Y. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1095–1105, Sep. 2012.
[21] Q. Yan, B. H. Zhang, and M. Kezunovic, "Optimized operational cost reduction for an EV charging station integrated with battery energy storage and PV generation," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2096–2106, Mar. 2019.
[22] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, May 2013.
[23] Y. Mou, H. Xing, Z. Lin, and M. Fu, "Decentralized optimal demand-side management for PHEV charging in a smart grid," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 726–736, Mar. 2015.
[24] M. Liu, S. McLoone, S. Studli, R. Middleton, R. Shorten, and J. Braslavs, "On-off based charging strategies for EVs connected to a low voltage distributon network," in *Proc. IEEE PES Asia–Pac. Power Energy Eng. Conf.*, 2013, pp. 1–6.
[25] B. Radunovic and J.-Y. Le Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1073–1083, Oct. 2007.
[26] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*. New York, NY, USA: McGraw-Hill, 2010.
[27] O. E. Gouda, G. M. Amer, and W. A. A. Salem, "Predicting transformer temperature rise and loss of life in the presence of harmonic load currents," *Ain Shams Eng. J.*, vol. 3, no. 2, pp. 113–121, 2012.
[28] D. Susa, M. Lehtonen, and H. Nordman, "Dynamic thermal modelling of power transformers," *IEEE Trans. Power Del.*, vol. 20, no. 1, pp. 197–204, Jan. 2005.

Roman Rudnik (Student Member, IEEE) was born in Moscow, Russia, in 1991. He received the Undergraduate degree in computational mathematics from Lomonosov Moscow State University, Moscow, Russia, in 2013. He is currently working toward the Ph.D. degree with the School of Computer Science and Communication Systems, Swiss Federal Institute of Technology of Lausanne, Lausanne, Switzerland, under the supervision of Prof. J.-Y. Le Boudec.

His current research focuses on real-time control of electrical grids with special reference to charging of electric vehicles.



Cong Wang (Student Member, IEEE) received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China in 2013, and the Ph.D. degree in computer and communication sciences from the School of Computer Science and Communication Systems, Swiss Federal Institute of Technology of Lausanne, Lausanne, Switzerland, in 2019.

His research interests include convex optimization, signal processing, and functional and numerical analysis, with their applications to networked systems.

Lorenzo Reyes-Chamorro (Member, IEEE) was born in Santiago, Chile, in 1984. He received the B.Sc. degree in electrical engineering from the University of Chile, Santiago, Chile, in 2009 and the Ph.D. degree from the Swiss Federal Institute of Technology of Lausanne (EPFL), Lausanne, Switzerland, in 2016.

From 2016 to 2018, he was a Postdoctoral Fellow with the Distributed Electrical System Laboratory, EPFL. He is currently an Assistant Professor with the Institute of Electricity and Electronics, Facultad de Ciencias de la Ingeniería, Universidad Austral de Chile, Valdivia, Chile.

Jagdish Prasad Achara (Member, IEEE) received the B.Tech. degree in communication and computer engineering from LNM Institute of Information Technology, Jaipur, India, in 2010 and the master's degree in computer services, security, and networks from Nancy University, Nancy, France, in 2011, and the Ph.D. degree in web, smartphone, and data privacy from INRIA and Grenoble-Alpes University, Grenoble, France, in 2016.

He is currently a Postdoctoral Researcher with the Swiss Federal Institute of Technology of Lausanne, Lausanne, Switzerland. In parallel, he is also the Founder and CEO of GridSteer. He is currently working on developing advanced monitoring and control techniques for sustainable and renewable energy based electric grids. Before doing the Ph.D., he was a Research and Development Engineer with INRIA for two years (2011–2013).

Jean-Yves Le Boudec (Fellow, IEEE) received the graduate degree from the Ecole Normale Supérieure de Saint-Cloud, Paris, France, where he obtained the agrégation in mathematics, in 1980, and the doctorate degree from the University of Rennes, Rennes, France, in 1984.

He is currently a Professor with the Swiss Federal Institute of Technology of Lausanne (EPFL), Lausanne, Switzerland. From 1984 to 1987, he was with INSA/IRISA, Rennes, France. In 1987, he joined Bell Northern Research, Ottawa, ON, Canada, as a member of scientific staff in the Network and Product Traffic Design Department. In 1988, he joined IBM Zurich Research Laboratory where he was the Manager of the Customer Premises Network Department. In 1994, he became an Associate Professor with EPFL. He coauthored a book on network calculus, which forms a foundation to many traffic control concepts in the Internet, an introductory textbook on Information Sciences, and is also the Author of the book *Performance Evaluation* (EPFL Press, 2010). His research interests include performance and architecture of communication systems and smart grids.

Mario Paolone (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in electrical engineering from the University of Bologna, Bologna, Italy, in 1998 and 2002, respectively.

In 2005, he was an Assistant Professor in Power Systems with the University of Bologna, where he was with the Power Systems Laboratory until 2011. Since 2011, he has been with the Swiss Federal Institute of Technology, Lausanne, Switzerland, where he is currently a Full Professor and the Chair of the Distributed Electrical Systems Laboratory. His research interests include power systems with particular reference to real-time monitoring and operation aspects, power system protections, dynamics, and transients.

Dr. Paolone has authored or coauthored more than 300 papers published in mainstream journals and international conferences in the area of energy and power systems that received numerous awards including the 2013 IEEE EMC Technical Achievement Award, two IEEE Transactions on EMC Best Paper Awards, and the Basil Papadias Best Paper Award at the 2013 IEEE PowerTech. He was the Founder Editor-in-Chief for the Elsevier journal *Sustainable Energy, Grids and Networks*.