Reinforcement Learning for charging scheduling in a renewable powered Battery Swapping Station

Reinforcement Learning for charging scheduling in a renewable powered Battery Swapping Station

Daniela Renga, *Senior, IEEE,* Felipe Spoturno and Michela Meo, *Senior, IEEE*

***Abstract*—Battery Swap (BS) technology represents a promising solution to overcome the main obstacles to a widespread adoption of electric vehicles (EVs) in a urban environment, like the limited range of EVs and the long battery charging time. Furthermore, with respect to traditional charging stations, it offers higher flexibility in dynamically managing the EV electricity demand to prevent the risk of power grid overload. Nevertheless, proper scheduling of the battery charge process is crucial to offer effective e-mobility services, trading off cost, Quality of Service and feasibility constraints. In this paper we consider a renewable powered multi-socket Battery Swapping Station (BSS) and design two algorithms based on Approximate Dynamic Programming (ADP) and Reinforcement Learning (RL) to dynamically adapt the scheduling of the battery charging process to the stochastic nature of the system. Both approaches are proved to be effective in remarkably enhancing the service quality in terms of increased capability to satisfy the customer demand for EV battery charging, at a lower cost with respect to benchmark approaches, with RL outperforming ADP under any budget constraint. In particular, under RL the probability of not satisfying the EV demand can be decreased by up to more than 40% with respect to benchmark approaches, and a significant cost reduction of almost 20% can be achieved, jointly with a greener system operation. Furthermore, our results show that a fine tuning of hyper-parameters is fundamental to properly trade off cost and Quality of Service constraints according to varying business needs. Finally, we analyse how the proposed strategies may affect the battery health due to their impact on battery degradation, hence influencing the BSS management cost.**

***Index Terms*—Battery Swapping Stations, e-mobility, Renewable Energy, Dynamic Programming, Reinforcement Learning.**

I. Introduction

Nowadays the transportation sector still relies mainly on oil as its main energy source, capable to satisfy more than 90% of the overall demand so that road transportation alone is responsible of the half the total oil consumption among all sectors [1]. Interestingly, road transportation is the only sector that has been characterized by a considerable growth of the total final oil consumption in the past years. Indeed, achieving an amount of 2000 Mtoe per year, this consumption has become three-fold larger than five decades ago, determining a raising trend that poses remarkable concerns in terms of sustainability [1]. In addition, air pollution represents a relevant critical issue related to traditional transportation, especially in urban environments. In a similar scenario, the adoption of Electric vehicles (EVs) is gradually emerging as a promising solution to address all the discussed concerns raised by road transportation, as long as EV charging is provided by energy supply systems which rely on Renewable Energy Sources (RES) [2]. Nevertheless, several barriers still prevent a wider diffusion of EVs. First, a more extensive penetration of EVs is hindered by the lack of adequate charging infrastructure [3], [4]. Second, the cost per kWh reduction is not yet sufficient to guarantee a purchasing cost for an EV comparable to that of Internal Combustion Engine (ICE) vehicles [5], [6]. Furthermore, the travel range per charge still results rather limited with respect to ICE vehicles, hence leading drivers to experience a significant range anxiety (fear that a vehicle has insufficient range to reach the destination) [7], [8]. Finally, an additional relevant drawback that slows down the diffusion of EVs on a larger scale is represented by the long charging time of EV batteries, since fast chargers represents only about 13% of the newly installed chargers [2], also due to the huge and unpredictable load that may pose on the electricity distribution infrastructure [9], [10].

In this context, Battery Swap technology may play a key role to overcome the highlighted obstacles and speed up a widespread adoption of EVs [11]. Based on this technology, EVs are equipped with energy storage units that can be easily substituted, so that a drained out battery can be replaced at a Battery Swapping Station (BSS) with a fully charged one in a short time. Owned by independent companies, BSSs operate in a similar way to a fuel filling station for ICE vehicles, virtually making the time experienced by drivers to obtain a fully charged battery negligible. The time for battery swap becomes comparable with the time needed to refuel a traditional car and the range anxiety is significantly mitigated. New business model can be conveniently introduced. EVs can be owned by private users or a car sharing company, whereas batteries are owned and managed by a centralized provider, which is in charge of all the operations and cost required for battery maintenance. EV prices can hence be lowered and users relieved by the need to cope with exhausted batteries.

Nevertheless, the full potential of BSS technology cannot be effectively exploited without considering the fundamental need for jointly pursuing possibly conflicting objectives that coexist during the BSS operation: satisfying the EV user demand, limiting the operational cost, preventing the electric grid overload, and reducing the amount of energy drawn from the grid, hence allowing to achieve sustainability goals. To this aim, Battery Swap technology enables the application of smart scheduling schemes for the battery charging process. Since the requests for a fully charged battery and the battery charge process can be decoupled in a BSS, the time constraint on recharging is lessened, hence allowing to more flexibly

M. Meo, D. Renga, and F. Spoturno are with the Department of Electrical, Electronic and Telecommunicatiions Engineering, Politecnico di Torino, Italy.

and dynamically suspend and resume the charging process. This allows, on the one hand, to better adapt the BSS energy demand to the RES production and availability. On the other hand, the BSS operator can enhance the interaction with the Smart Grid (SG), scheduling the battery charging during periods of low electricity prices, high RES availability, and based on incentives offered by the SG to avoid peaks of electricity load.

In this paper, we focus on urban e-mobility based on battery swap technology, considering a renewable powered BSS. We design different novel charging scheduling strategies based on Dynamic Programming (DP) and Reinforcement Learning (RL). We investigate the impact of these strategies on the performance of the BSS operation, evaluating their potential to trade off cost and Quality of Service (QoS) in terms of capability to satisfy the customer demand for EV battery charging. Furthermore, to demonstrate the superior performance of the proposed strategies, the performance achieved under the operation of the presented methods is compared against benchmark charging scheduling approaches. The BSS and the decision making process for the EV battery charge scheduling is modeled as a Markovian Decision Problem. The main contributions of the paper are the following:

- we design two adaptive algorithms, based on Approximate Dynamic Programming and Reinforcement Learning, respectively, to modulate and dynamically adapt the scheduling of the EV battery charging process at a BSS to the stochastic nature of the system. Charging scheduling decisions are taken keeping into account the variability of EV arrival rates, electricity prices, and renewable energy production profiles;
- we extensively evaluate the performance of the proposed charging scheduling algorithms via simulation, showing that they can provide significantly better service (i.e., the customers can more likely find a fully recharged battery at the BSS upon arrival) at a lower cost with respect to benchmark approaches, including a Heuristic control algorithm and Exact Policy based strategies;
- we thoroughly investigate the effects on system performance of varying the hyper-parameters settings of the proposed algorithms. Varying these settings allows to weight differently the energy costs and the probability of missed service (i.e. an EV cannot find a ready battery upon arrival at the BSS). We demonstrate that a fine tuning of the hyper-parameters represents an effective tool to find an operational point that properly trades off cost and Quality of Service according to business needs;
- finally, we analyse how the proposed charging scheduling strategies affect performance indicators that may influence the battery health, showing that the designed scheduling methods outperform benchmark approaches in preserving EV batteries from degradation phenomena.

II. Related work

The emerging research interest on battery swap technology is proved by several studies investigating its potential to foster the penetration of EV technology [11], [12], [13], [14]. Furthermore, several research efforts focus on deploying charging scheduling strategies to provide optimal operation of BSS systems [15], [16], [17]. The study in [15] proposes a dynamic scheduling mechanism based on Rolling-Horizon optimization to maximize the daily profit for a BSS serving different types of EVs that exploit a heterogeneous battery stock. Further charging and swapping scheduling approaches in the literature rely on cuckoo search algorithms [16], [17] and particle swarm optimization [16] to improve cost-effectiveness and enhance the operation efficiency of charging facilities.

Only few studies exploit Dynamic Programming (DP) for battery charge scheduling in traditional charging stations either to manage the charging and discharging of EV batteries integrated in a smart residential framework [18] or to schedule a large-scale EV charging in a power distribution network under stochastic renewable generation and electricity prices [19]. Similarly, various studies exploit RL based methods to properly schedule and manage the battery charging process. Some works in the literature exploit deep RL methods to properly schedule the charging of a single EV, based on Q-learning [20], [21], [22] or safe deep RL approaches [23]. A SAC method is proposed by [24], that focus on the individual EV charging scheduling problem from the users' perspective, deploying a deep RL based approach to derive an optimal charging control strategy, in order to trade off charging cost and driver's anxiety on the driving range and uncertain events. Like in [19], more recent studies deploy RL based solutions to address the charging management of multiple EVs [25], [26], [27]. In [25] a bilevel graph RL method is proposed for the jointed management of an EV fleet routing and charging, with the twofold objective of reducing charging cost and traveling time for drivers. Similarly, [26] proposes a Deep RL methodology for constraint-based routing while simultaneously considering EV charging policies. Authors in [27] deploy a RL based method to perform optimal energy management in a microgrid environment, that includes EV charging stations and distributed energy resources.

Nevertheless, only a limited amount of studies consider DP and RL to specifically address the charging scheduling problem in battery swapping services in urban e-mobility [28], [29], [30]. The study in [28] proposes a RL based charging model that adapts to the incoming vehicle arrival rates to trade off revenues deriving from swapping operations and the cost for battery charging. In [29] a multi-agent deep RL algorithm to tackle the optimization charging scheduling problem in a scenario consisting of centralized battery charging stations (BCSs) and distributed BSSs, where fully recharged batteries are delivered from BCSs and made available to EVs. Authors in [30] propose a model-free optimal dynamic operations framework based on deep RL to minimize the operational costs of a BSS by controlling the charging/discharging and swapping actions. Considering the charging scheduling problem in renewable powered BSSs, the literature offers some solutions based on heuristics and optimization methods based on integer linear programming to address this issue [13], [31]. However, despite the research efforts devoted to deploy RL based scheduling algorithms in BSS scenarios, the application of similar scheduling approaches to operate renewable powered BSSs does not appear well investigated in the literature.

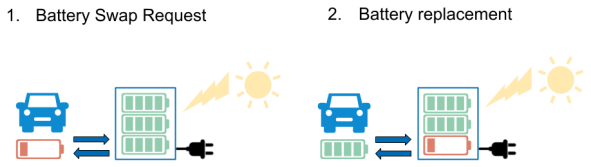


Fig. 1: Renewable powered Battery Swapping Station.

Moreover, in the scenarios investigated in the available studies, battery swap services are typically offered to fleets of electric buses providing urban electric mobility [12], [14]. Conversely, in our work we specifically consider a scenario where a renewable powered multi-socket BSS provides battery charging and replacement service to EVs in a urban environment. Furthermore, the small EVs considered in our work differ from the electric buses used for public transportation, since our EVs are characterized by a smaller battery capacity, entailing a lower energy demand to fully recharge a battery unit, and by less predictable routing dynamics. In addition, we design real-time algorithms based on DP and RL to optimize the battery charging scheduling, taking into account the variability of renewable energy production, electricity prices and EV demand. Differently from the currently available literature, the proposed scheduling strategies jointly trade off cost reduction and QoS, in terms of capability of serving the EV demand. Furthermore the performance analysis of the proposed scheduling methods includes the evaluation of their impact on various performance metrics that may affect battery health.

III. Sustainable urban mobility scenario

We investigate a scenario where a fleet of EVs owned by a private company either offers goods delivery service or car sharing service over a urban environment. A number of BSSs are distributed in the city to provide battery charging service to the fleet of EVs. As depicted in Fig. 1, we focus on a single BSS featuring a number of sockets that is denoted by M. The main notations adopted in this work are reported in Table I.
EVs are equipped with a removable lithium-ion battery unit that can be replaced as required by a fully recharged battery, with a fast swap operation performed at the BSS. The discharged battery is then plugged to a BSS socket to begin the recharge process. When an EV arrives at the BSS and no fully recharged battery (or anyway featuring a predefined charge level) is currently available at the BSS, the EV cannot be served by the considered BSS and another BSS must be reached. We highlight that in a similar context the new concept of *Battery-as-a-Service* is introduced. The pool of batteries that can be plugged to the BSS sockets to complete the charging process is owned by the BSS system operator, while EV owners can subscribe the service offered by the BSS operator, consisting in the possibility to replace an EV discharged storage unit with a battery recharged up to the desired level, swapped at the BSS. In this way, EV owners are not burdened by the remarkable cost required to directly purchase and manage the EV battery. We assume battery units featuring a capacity of 20 kWh, as suggested from the literature for a typical electric city car [32]. The nonlinear charging power of lithium-ion battery technology makes it difficult to accurately predict the duration of the charging process for each storage unit [11]. Slow charge rates are recommended to guarantee slower battery aging and longer cycle life [33]. For an optimal recharging process, the maximum nominal charging rate is typically limited to $\frac{C^N}{2}$ per hour, where C^N is the nominal battery capacity [33]. In our scenario, a maximum charging rate of 10 kW is hence assumed.
The charging rate of the EV battery is highly affected by the battery state of charge (SOC), that represents the fraction of

TABLE I: Notation.

SOC	State of charge of a battery
SOH	State of health of a battery
C^N	Nominal battery capacity of the EV battery
C	Actual battery capacity of the EV battery under real SOH conditions
D_{max}	Maximum Depth of Discharge of the battery
L	Fraction of the battery capacity C corresponding to the battery charge level of an EV battery upon arrival at the BSS
a_B	Age of battery
E_0	Maximum energy demand required to fully recharge an EV battery
B_{th}	Minimum charge level allowing to release a plugged battery to replace the discharged battery of an EV upon arrival
r^N	Nominal battery charging rate, assuming ideal SOH conditions and SOC = 50%
r^A	Actual battery charging rate under real SOC and SOH operation conditions
M	Number of sockets of a BSS
λ	Average EV arrival rate
x_k	Amount of energy required at time step k to fully charge a plugged battery
u_k	Amount of energy used to recharge a plugged battery during time slot k
ω_k^λ	Bernoulli random variable that indicates whether during time slot k an EV arrival occurs ($\omega_k^\lambda = 1$) or not ($\omega_k^\lambda = 0$)
p_k^λ	Probability that an EV arrival occurs during time slot k
ω_k^{PV}	Amount of energy produced by the PV panel during time slot k
a_k	Unitary cost for the energy taken from the grid at time step k
β	Hyper-parameter to weight the trade-off between the cost for the energy drawn from the grid and the cost for missing an EV service demand
$\mathbf{x_k}$	Column vector representing the state variable x_k for each plugged battery in a multi-socket BSS
$\mathbf{u_k}$	Column vector representing the control variable u_k for each plugged battery in a multi-socket BSS
$\mathbf{w_k^\lambda}$	Binary vector indicating the number of cars (from 0 to M) arriving in the time slot k at a multi-socket BSS
c_E	Mean grid energy cost
$\hat{c_E}$	Normalized grid energy cost
p_G	Mean service loss probability
$\hat{p_G}$	Normalized service loss probability
g_E	Grid energy consumption
c_S	Cost per service
t_S	Mean time spent by a battery in the BSS
n_S	Mean number of times that the charging of a battery is suspended and resumed during the charging period
r	Monetary reward for the locally produced extra renewable energy that is injected and sold to the SG
C^Y	Yearly operational cost, including the battery replacement cost

energy stored in a battery unit with respect to its full capacity. The highest charging rate is observed under SOC between 25% and 80% [34], a range in which a constant current can be assumed to quickly recharge the battery [35]. Conversely, as the battery SOC gets closer to the extreme values, that correspond to the battery depletion or to the full recharge, the charging rate tends to progressively decrease [34].
In addition, the battery state of health (SOH) is impacted by degradation phenomena, that mainly depend on the battery age and temperature [36]. The SOH degradation, in its turn, leads to a progressive reduction of the actual available battery capacity over time with respect to the nominal one.
Let us denote by r^N the nominal battery charging rate, observed for a new battery under ideal SOH conditions and for intermediate SOC, i.e., 50%. Let C and r^A represent the actual battery capacity and the actual battery charging rate, respectively, under real SOC and SOH operation conditions. The actual battery charging rate is computed as follows:

$$r^A = \frac{C}{2} \cdot f_{SOC} \tag{1}$$

where f_{SOC} is a scaling factor to take into account the impact of the battery SOC on the charging rate. C is derived as $C^N \cdot f_{SOH}$, where f_{SOH} is a scaling factor to take into account the effect of the battery SOH on the actual available battery capacity. Based on [34], the value of f_{SOC} is defined as follows:

$$f_{SOC} = \begin{cases} 0.69 & \text{if } SOC \leq 0.25 \\ 1 & \text{if } 0.25 < SOC < 0.80 \\ 0.69 & \text{if } SOC \geq 0.80 \end{cases}$$

A linear approximation of f_{SOH} as a function of the battery age at a given temperature is derived in [36] as:

$$f_{SOH} = \begin{cases} 1 - b \cdot a_B & \text{if } a_B \leq 80 \ days \\ c \cdot a_B + d & \text{if } a_B > 80 \ days \end{cases}$$

where a_B represents the battery age expressed in days, with parameters b, c and d set as follows:

$$(b,c,d) = \begin{cases} (9.9985 \cdot 10^{-1},\ 3.0 \cdot 10^{-5},\ 9.904 \cdot 10^{-1}) & if\ T = 25°C \\ (9.9970 \cdot 10^{-1},\ 8.5 \cdot 10^{-5},\ 9.828 \cdot 10^{-1}) & if\ T = 40°C \end{cases}$$

Besides the powering provided to the BSS by the electric grid, a renewable energy (RE) supply, represented by a set of photovoltaic (PV) panels, is envisioned to recharge the battery units with locally produced solar energy, as shown in Fig. 1. Assuming one of the most efficient crystalline silicon technologies currently on the market, characterized by 19% efficiency, the physical area occupancy required to install PV modules is about 5 m^2 per kWp of PV module capacity [37]. Real traces for RE generation are derived from the PVWatts tool [38] for a city in Northern Italy during the typical meteorological year. To evaluate the BSS operational cost, we consider real electricity prices derived from the Day-Ahead Market, provided by *Gestore dei Mercati Energetici* (GME), the Italian company responsible for the electricity market management [39].

EVs are assumed to arrive at the BSS to replace their discharged battery according to an inhomogeneous Poisson process, characterized by average arrival rate λ varying with the time of the day on an hourly basis, as typically done in the the literature for scenarios envisioning traditional EVs [40], [41], [42]. The dynamics of the EV usage and of the battery charging process at the BSS may be different with respect to those observed in a scenario with traditional EVs and charging stations, possibly leading to very different EV arrival patterns. Therefore, the patterns based on real data about EV arrivals at traditional charging stations may not be suitable to properly represent the actual behavior of EV arrivals in a BSS system [43]. We hence adopt the daily traffic profile reported in Fig. 2, as in our previous work [13], that is derived taking inspiration from typical models of EV arrival rates that are adopted in the literature [43], [44], [41], but at the same time accounting for a possibly different behavior in a BSS scenario, hence obtaining a plausible pattern that reflects traffic variations during the day, showing traffic peaks at the beginning of the working day, during lunchtime, and in the evening. We assume a maximum value of the Depth of Discharge of the battery, D_{max}, with respect to its actual capacity, C, in order to limit the battery degradation and improve its lifetime [45]. The minimum allowed battery charge level is hence equal to $C \cdot D_{max}$. Under this constraint, the risk of fully running out of battery is avoided, allowing the EV to reach another BSS in case no storage units are currently ready at the BSS to replace the EV battery. The battery charge level of the EVs that arrive at the BSS is denoted $L \cdot C$, and it is expressed as a fraction, L, of the battery capacity C. Finally, to reduce the probability of service unavailability during peaks of EV demand, a battery under charge at the BSS may be made available even if it is not fully recharged yet. In particular, battery units may become available for swapping as soon as they are recharged up to a threshold level, denoted B_{th}, corresponding to a fraction of the battery capacity C. EV users may exhibit different requirements for the desired battery charge level after the storage unit replacement. Nevertheless, to limit the system complexity, we assume a constant value for B_{th} for any user during the entire system operation, that reasonably reflects an average threshold setting based on different user needs, and accounts for the fact that short-medium range routes of EVs in urban environment usually do not require a fully recharged battery. On top of this, we assume that the charge of the battery units plugged at the BSS may be temporarily postponed, in order to both optimize the renewable energy utilization and take advantage of lower electricity prices, still satisfying the EV battery replacement demand. To this aim, different strategies are implemented to dynamically schedule the battery charging process, taking proper scheduling decisions that allow to trade

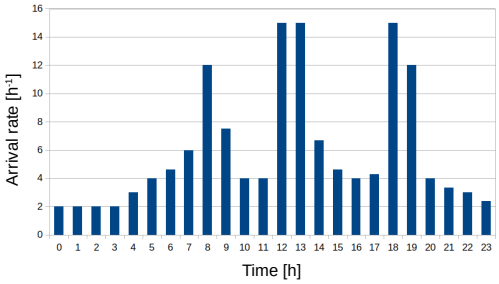


Fig. 2: Daily profile of EV arrival rates [13].

off operational cost, RE utilization, QoS requirements, and sustainability goals.

IV. Modeling BSS operation

We now model the BSS operation and formalize the cost minimization problem, considering both the case of a single-socket BSS and the case of a multiple-socket BSS.

A. Single-socket BSS modeling

In this study, a Markovian Decision Process is used to model the operation of a BSS with M sockets and no waiting queue, i.e., assuming that an EV arriving at the BSS picks the battery under charge only if its charge level is higher than the defined threshold B_{th}, otherwise it leaves the station without replacing the battery. We first consider the single-socket BSS case (M=1). The system is represented through a discrete-time and discrete state-space model, where each state x_k corresponds to the energy demand that is required to fully charge the plugged battery at time step k, with state $x_k = 0$ indicating a fully charged battery and $x_k = C$ corresponding to an empty battery. The control variable u_k represents the amount of energy that is used to recharge the battery during time slot k. A value of u_k equal to 0 means that the charging process of the considered battery is deactivated (OFF), whereas under positive values of u_k the charging process is activated (ON). An EV arrives at the BSS at time slot k with probability p_k^λ that depends on the current value of λ. We denote ω_k^λ the Bernoulli random variable used to indicate whether during time slot k at least an EV arrives or not, assuming $\omega_k^\lambda = 1$ with probability p_k^λ and $\omega_k^\lambda = 0$ with probability $1 - p_k^\lambda$. In the unlikely event of more than M electric vehicles arriving during the same time slot (whose duration is set to 5 minutes), we assume that the additional EVs cannot be served by the BSS. These missed services, that are rather due to an underdimensioned BSS infrastructure than to the management of the EV battery charging scheduling, are neglected. Indeed, our study focuses on the deployment of effective EV charging scheduling strategies, whereas for a more specific focus on the BSS dimensioning in terms of number of sockets, M, with respect to the EV demand, the reader can refer to our previous work [13]. A further random variable, denoted ω_k^{PV}, represents the amount of energy produced by the photovoltaic panel during time slot k. The system evolves in time according to the following state transitions:

$$x_{k+1} = \begin{cases} C & \text{if } \omega_k^\lambda = 1 \text{ and } [x_k - u_k]^+ \le E_0 \\ [x_k - u_k]^+ & \text{otherwise} \end{cases} \tag{2}$$

where $E_0 = C \cdot (B_{th} - L_{min})$, $k = 1, 2, \dots N-1$, with N representing the state horizon of the system. Basically the state of the system depends on the battery charging process, and an empty battery is left on the system after an EV has replaced its battery upon arrival. The state transitions from (2) can also be written as:

$$\begin{aligned} x_{k+1}\left(\omega_k^\lambda, u_k\right) &= \omega_k^\lambda \cdot \mathbb{I}\left\{[x_k - u_k]^+ < E_0\right\} \cdot C \\ &+ \left(1 - \omega_k^\lambda \cdot \mathbb{I}\left\{[x_k - u_k]^+ \le E_0\right\}\right) \cdot [x_k - u_k]^+ \end{aligned} \tag{3}$$

where $\mathbb{I}\{\cdot\}$ stands for the indicator function.

Considering the variability of renewable energy production and of electricity prices, decisions about battery charging scheduling are made at each time step in order to minimize both the cost for the electricity bought from the grid and the cost for a missed service in case an EV cannot replace its empty battery. We first define the *step cost function* for the analyzed system at time step k as:

$$\begin{aligned} g_k\left(x_k, u_k, \omega_k^\lambda, \omega_k^{PV}\right) &= (1-\beta) \cdot a_k \cdot \left[u_k - \omega_k^{PV}\right]^+ \\ &+ \beta \cdot \omega_k^\lambda \cdot \mathbb{I}\left\{[x_k - u_k]^+ > E_0\right\} \end{aligned} \tag{4}$$

where a_k is the deterministic unitary cost of the energy taken from the grid at time k, $\beta \in [0,1]$ is an hyper-parameter used to weight the trade-off between the cost of the energy drawn from the grid and the cost for not being able to serve an EV. Low values of β give priority to minimizing the energy cost, whereas high values of β weight more the reduction of missing an EV service demand.

B. Multiple-socket BSS modeling

A multiple-socket BSS ($M > 1$) can be thought as a single-socket station in which the state and the control variables of the system are represented by column vectors $\mathbf{x_k}$ and $\mathbf{u_k}$. It is convenient to sort the elements of vector $\mathbf{x_k}$ (i.e., $\mathbf{x_k}^{(i)}$, with $0 \le i \le M-1$) based on the increasing state of energy demand of the batteries, so that the state of the system is actually $\mathbf{x'_k} = f(\mathbf{x_k})$, where f is a permutation of the states. In this way $\mathbf{x'_k}^{(i)} \le \mathbf{x'_k}^{(j)}$ for each $i \le j$. Note that the first element of the vector, $\mathbf{x'_k}^{(0)}$, corresponds to the most charged battery, hence the storage unit that is going to replace the empty battery of the next arriving EV. As an extension of the case of a single socket system, the random variable ω_k^λ becomes now a binary vector $\mathbf{w_k^\lambda}$ indicating the number of EVs (from 0 to M) arriving during time slot k:

$$\mathbf{w_k^{\lambda(i)}} = \begin{cases} 1 \text{ (arrivals} > i) & \text{with probability } p_k^{\lambda(i)} \\ 0 \text{ (arrivals} \le i) & \text{with probability } 1 - p_k^{\lambda(i)} \end{cases} \tag{5}$$

As it will be further detailed in Equation 17 and Equation 18, the value of $p_k^{\lambda(i)}$ corresponds to $1 - F(i)$, where $F(i)$ is the cumulative distribution function of the number of EV arrivals in time slot k.

The system hence evolves according to the following equation:

$$\begin{aligned} \mathbf{x'_{k+1}} &= C \cdot \mathbb{I}\left\{f\left(\left[\mathbf{x'_k} - \mathbf{u'_k}\right]^+\right) < E_0\right\} \circ \mathbf{w_k^\lambda} \\ &+ \left[\mathbf{1} - \mathbb{I}\left\{f\left(\left[\mathbf{x'_k} - \mathbf{u'_k}\right]^+\right) < E_0\right\} \circ \mathbf{w_k^\lambda}\right]^T \cdot f\left(\left[\mathbf{x'_k} - \mathbf{u'_k}\right]^+\right) \end{aligned} \tag{6}$$

where $\circ$ stands for the element-wise product (or Hadamard product), and $\mathbb{I}\{\cdot\}$ is applied to a vector verifying the condition for each component. The cost of each step becomes:

$$\begin{aligned} g_k\left(\mathbf{x'_k}, \mathbf{u'_k}, \mathbf{w_k^\lambda}, \omega_k^{PV}\right) &= (1-\beta) \cdot a_k \cdot \left[\sum_i \mathbf{u'_k}^{(i)} - \omega_k^{PV}\right]^+ \\ &+ \beta \cdot \mathbb{I}\left\{f\left(\left[\mathbf{x'_k} - \mathbf{u'_k}\right]^+\right) > E_0\right\}^T \cdot \mathbf{w_k^\lambda} \end{aligned} \tag{7}$$

V. Dynamic charging scheduling

We first introduce a Stochastic Dynamic Programming based algorithm to dynamically schedule the charging of EV batteries plugged at the sockets of a BSS. The algorithm aims

at minimizing the BSS operational cost, by means of proactively suspending and resuming the charging process when it is more convenient, depending on EV demand, electricity cost and RE availability. We then consider the application of this scheduling approach both in the simple case of a single-socket BSS and in the more challenging case of a multiple-socket BSS. To tackle the feasibility issues raised in the multiple-socket BSS scenario and reduce the computational complexity of the DP based algorithm, we propose a set of strategies, based on Approximate Dynamic Programming and Reinforcement Learning. Finally, besides introducing an *Heuristic scheduling strategy*, we describe *Exact policy solutions* that allow to analytically compute the optimal charging scheduling policy operation, representing a benchmark against which the performance of the other approaches can be compared.

A. Stochastic Dynamic Programming

The Stochastic DP based scheduling approach is here described. We detail how it can be applied in the single socket BSS scenario, considering first the Finite-Horizon DP and motivating the need for introducing an Infinite-Horizon DP approach. Furthermore, we introduce the limits of applying a DP scheduling in a multiple socket BSS scenario, that require the deployment of other types of scheduling algorithms.

1) Finite-Horizon DP Algorithm: A Stochastic DP approach is adopted to take proper scheduling decisions with the purpose of minimizing the overall operation cost cumulated over the entire time period of the system evolution, i.e., from $k=0$ to $k=N$. To this aim, we first define a cost function that is additive over time:

$$J(x_0) = \sum_{k=0}^{N-1} g_k\left(x_k, u_k, \omega_k\right) + g_N\left(x_N\right) \tag{8}$$

where $g_k\left(x_k, u_k, \omega_k\right)$ is the step cost function, whereas $g_N\left(x_N\right)$ is a terminal cost incurred at the end of the process, thus it depends only on the final state N. Due to the presence of ω_k, the total cost is generally a random variable. We assume a finite horizon of N time slots, covering the duration of an entire day.

Let us denote a generic scheduling policy with $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$, where μ_k maps states x_k into controls $u_k = \mu_k\left(x_k\right)$ to determine the charging scheduling at each time step. Under this policy, the evolution of the system becomes:

$$x_{k+1} = f_k\left(x_k, \mu_k\left(x_k\right), \omega_k\right) \tag{9}$$

The associated expected value of the cumulative system cost under the policy π starting from a generic initial state x_0, that we denote $J_\pi(x_0)$, is defined as:

$$J_\pi\left(x_0\right) := \mathbb{E}\left[\sum_{k=0}^{N-1} g_k\left(x_k, \mu_k\left(x_k\right), \omega_k\right) + g_N\left(x_N\right)\right] \tag{10}$$

The DP objective is to identify an optimal policy π^* minimizing J over the set of all the possible policies Π, that is:

$$J_{\pi^*}\left(x_0\right) = \min_{\pi \in \Pi} J_\pi\left(x_0\right) \tag{11}$$

To achieve this objective, DP relies on the *principle of optimality* [46]. Let $\pi^* = \left\{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\right\}$ be an optimal policy for the basic problem, and assume that, under policy π^*, a given state x_i occurs at time i with positive probability. Consider the subproblem whereby the system is in state x_i at time i and we want to minimize the *cost-to-go* from time i to time N, that we denote $J_\pi(x_i)$, which represents the cost incurred by the system in the considered time interval:

$$J_\pi(x_i) = \mathbb{E}\left[\sum_{k=i}^{N-1} g_k\left(x_k, \mu_k\left(x_k\right), \omega_k\right) + g_N\left(x_N\right)\right] \tag{12}$$

Clearly, for the considered subproblem, the truncated policy $\left\{\mu_i^*, \mu_{i+1}^*, \ldots, \mu_{N-1}^*\right\}$ results optimal. The principle of optimality suggests that an optimal policy can hence be constructed by solving first the tail subproblem involving the last stage x_N, then extending stage by stage the tail subproblem until an optimal policy is constructed for the entire problem. This strategy of building the solution backwards in time is known as the *Dynamic Programming Algorithm*. More in detail, for any initial state x_0, the optimal cost $J^*\left(x_0\right)$ of the basic problem is equal to $J_0\left(x_0\right)$, and it can be derived from the last step of the following algorithm, which computes $J_k\left(x_k\right)$ proceeding recursively and backwards in time starting from $k = N-1$ up to $k = 0$:

$$J_k\left(x_k\right) = \min_{u_k\left(x_k\right)} \left\{ \mathop{\mathbb{E}}_{\omega_k} \left[g_k\left(x_k, u_k, \omega_k\right) + J_{k+1}\left(f_k\left(x_k, u_k, \omega_k\right)\right)\right]\right\} \tag{13}$$

where $g_k\left(x_k, u_k, \omega_k\right)$ represents the *step cost function* at time k, while the term $J_{k+1}\left(f_k\left(x_k, u_k, \omega_k\right)\right)$ corresponds to the *cost-to-go function* [46]. We highlight that the control space $\mathbf{U_k}\left(x_k\right) = \{0, u\left(x_k\right)\}$ is influenced by the SOC of the battery according to the following formula:

$$u(x_k) = f_{SOC} \cdot u_{max} \tag{14}$$

where u_{max} represents the maximum amount of energy that can be injected into a battery under charge during a time slot, assuming the maximum allowed charging rate. Note that u_{max}, in its turn, depends on the battery SOH, according to the following equation:

$$u_{max} = 0.5 C^A \cdot \triangle t \tag{15}$$

where $C^A = C^N \cdot \mathbb{E}[f_{SOH}]$. Whereas the algorithm operation takes into consideration the influence of the SOC variability on the charging rate of an EV battery, as it appears evident from (14), we remark that, in order to limit the complexity of the proposed charging scheduling algorithms, a constant value of C is considered. As specified in (15), C is computed assuming a constant average value for f_{SOH}, that is derived based on [36] and according to the configuration settings of the considered scenario, that are detailed in Section VII.

Note that $J_N(x_N) = g_N(x_N)$. Furthermore, if $u_k^* = \mu_k^*\left(x_k\right)$ minimizes the right side of Equation 13 for each possible x_k and k, the policy $\pi^* = \left\{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\right\}$ is optimal. From the application of the DP Algorithm to the single-socket BSS system, based on Equation 4 and Equation 13 we obtain:

$$\begin{aligned} J_k^*\left(x_k\right) = \min_{u_k\left(x_k\right)} \Bigg\{ \mathop{\mathbb{E}}_{\omega_k^\lambda, \omega_k^{PV}} \Big[& (1-\beta) \cdot a_k \cdot \left[u_k - \omega_k^{PV}\right]^+ \\ & + \omega_k^\lambda \cdot \beta \cdot \mathbb{I}\left\{\left[x_k - u_k\right]^+ > E_0\right\} + J_{k+1}^*\left(x_{k+1}\right) \Big] \Bigg\} \end{aligned} \tag{16}$$

TABLE II: Example of a control table after running the DP Algorithm.

	$k=0$	$k=1$	$\cdots$	$k=N$
$x_k=0$	$\left(u_0^*(x_0), J_0^*(x_0)\right)$	$\left(u_1^*(x_0), J_1^*(x_0)\right)$	$\cdots$	0
$x_k=1$	$\left(u_0^*(x_1), J_0^*(x_1)\right)$	$\left(u_1^*(x_1), J_1^*(x_1)\right)$	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x_k=C$	$\left(u_0^*(C), J_0^*(C)\right)$	$\left(u_1^*(C), J_1^*(C)\right)$	$\cdots$	0

Since ω_k^λ and ω_k^{PV} are independent random variables, we can split the expected value. Furthermore, considering that ω_k^λ is a Bernoulli variable, then $\mathbb{E}\left[\omega_k^\lambda\right] = p_k^{\lambda,n}$ at each time step, where $p_k^{\lambda,n}$ represents the probability that n or more EVs arrive at the BSS during the time window corresponding to time step k given the average arrival rate λ. It can be computed as:

$$p_k^{\lambda,n} = 1 - \sum_{i=0}^{n-1} {p'}_k^{\lambda,i} \tag{17}$$

where ${p'}_k^{\lambda,i}$ represents the probability that i EVs arrive at the BSS during time step k, given the arrival rate λ. Since interarrival times are exponentially distributed, the stochastic counting process of arrivals can be proved to be a Poisson variable, hence ${p'}_k^{\lambda,i}$ can be derived as follows:

$${p'}_k^{\lambda,i} = \frac{(\lambda_k \cdot \Delta t)^i \cdot e^{-\lambda_k \cdot \Delta t}}{(i)!} \tag{18}$$

Consequently, Equation 16 becomes:

$$J_k^*(x_k) = \min_{u_k(x_k)} \left\{ (1-\beta)\cdot a_k \cdot \left[u_k - \mathbb{E}\left(\omega_k^{PV}\right)\right]^+ + p_k^\lambda \cdot \beta \cdot \mathbb{I}\left\{[x_k - u_k]^+ > E_0\right\} + \mathop{\mathbb{E}}_{\omega_k^\lambda}\left[J_{k+1}^*(x_{k+1})\right] \right\} \tag{19}$$

Finally, from Equation 3 we derive the term $\mathop{\mathbb{E}}_{\omega_k^\lambda}[J_{k+1}^*(x_{k+1})]$:

$$\begin{aligned}\mathop{\mathbb{E}}_{\omega_k^\lambda}\left[J_{k+1}^*(x_{k+1})\right] &= p_k^\lambda \cdot J_{k+1}^*(x_{k+1}(1,u_k)) \\ &+ (1-p_k^\lambda)\cdot J_{k+1}^*(x_{k+1}(0,u_k))\end{aligned} \tag{20}$$

Based on Equations 19-20, the algorithm can be recursively run, to optimize the scheduling of the battery charging, knowing different time dependent system parameters: p_k^λ, $\mathbb{E}\left(\omega_k^{PV}\right)$, and a_k. The optimal solution that is obtained off-line is hence mapped into a lookup table that stores the pair $\left(u_k^*, J_k^*\right)$ for each discrete state x_k, as shown in Table II. This table can then be inspected on-line by the controller during the BSS operation to find the optimal scheduling strategy to apply given the current state of the system, hence determining whether in a given time slot it is convenient to recharge the battery. The DP Algorithm can be applied considering a *Finite Horizon*, i.e., the number of stages is finite. In our case, the $N-$th time step corresponds to the end of the day. We set $J_N(x_N) = 0$ so that every final state is equally penalized at the end of the day.

2) Infinite-Horizon DP through Value Iteration: As shown in Sec. VII-C, the application of a Finite Horizon DP algorithm on a daily basis may cause some misbehavior of the system as the end of the day approaches. The DP Algorithm solves the optimization problem starting from $k = N$, that in our case corresponds to the end of the day. The starting condition, $g_N(x_k)$, is set equal to 0, assuming that no particular cost is assigned for the state of the battery at the end of the day. Since the final cost is zero and given the limit on the maximum charging rate, as the end of the day approaches the current charging level of some batteries may not allow to achieve a full recharge by the end of the day. Hence, the DP algorithm may lead to stop the charging process of some batteries close to the end of the day, although this is unlikely in a realistic scenario. To overcome this critical issue, an infinite number of stages can be assumed and *Infinite Horizon* DP can be adopted, as long as the system is stationary. Although the system is not stationary on an hourly basis, it can be assumed approximately stationary on a daily basis, since solar radiation, EV arrival rates, and electricity prices may be similar from one day to the next one, in the same season. Infinite-Horizon DP relies on the concept of *Value Iteration* [46], that entails multiple iterations to converge to the optimal cost-to-go function, solving the Bellman equation and selecting a discount factor for the cost function, denoted α, that represents a weighting factor for the cost that will be accumulated in the future time steps. The optimal cost can hence be derived as follows:

$$J_k^*(x_k) = \min_{u_k(x_k)} \left\{ \mathop{\mathbb{E}}_{\omega_k^\lambda, \omega_k^{PV}} \left[g_k\left(x_k, u_k, \omega_k^\lambda, \omega_k^{PV}\right) + \alpha \cdot J_{k+1}^*(x_{k+1}) \right] \right\} \tag{21}$$

To properly implement Value Iteration on our system, the DP Algorithm is iteratively performed on a daily basis, starting from $k = N$. Once the beginning of the day is reached ($k = 0$), the accumulated cost $J_0^*(x)$ is set as the initial condition for a new DP Algorithm iteration, setting $J_N^*(x) := J_0^*(x)$. After the iteration is performed for a limited number of times, the system is able to *forget* the very initial condition that was set in Sec. V-A1, assuming $J_N^*(x) := 0$.

3) DP in the multiple-socket BSS: We now consider the application of DP in the multiple-socket BSS. In this case, to find the optimal cost, the DP algorithm would perform a cost minimization for each possible state, at each time step:

$$J_k^*\left(\mathbf{x'_k}\right) = \min_{\mathbf{u'_k} \in U_k\left(\mathbf{x'_k}\right)} \left\{ \mathop{\mathbb{E}}_{\mathbf{w_k^\lambda}, \omega_k^{PV}} \left[g_k\left(\mathbf{x'_k}, \mathbf{u'_k}, \mathbf{w_k^\lambda}, \omega_k^{PV}\right) \right] + \mathop{\mathbb{E}}_{\mathbf{w_k^\lambda}} \left[J_{k+1}^*\left(f\left(\mathbf{x'_{k+1}}\right)\right) \right] \right\} \tag{22}$$

However, this problem is computationally expensive, since the number of solutions to explore for each time step is exponential on M, involving at least $|\mathbf{x'}| \cdot |u|^M$ operations. The space for $\mathbf{x'}$ is composed by all the sorted combinations for x, which is:

$$|\mathbf{x'}| = \begin{pmatrix} |x| + M - 1 \\ M \end{pmatrix} \tag{23}$$

For example, for $|x| = 25$, $|u| = 2$ (for an on/off controller) and $|M| = 20$, this would imply at least $\sim 1.8 \cdot 10^{18}$ operations per time step for the construction of a Dynamic Programming lookup table. In addition, a critical issue arises due to the memory required to store all the actions to take for each one of the states. Instead of solving the complete table

Algorithm 1 Decoupled problem approximation.

1) At the beginning of the day compute M lookup tables running the DP algorithm, one per each socket.
   Input: x_k, u_k, ω_k^{λ}, ω_k^{PV} for any time slot in a day.
   Output: *Control table* and *cost-to-go table* per each socket.
2) At each time slot k:
   a) Re-order the states computing $\mathbf{x'_k} = f(\mathbf{x_k})$
   b) For each socket i, look in the i^{th} DP lookup table the optimal control $\mathbf{u'}_{\mathbf{k}}^{(i)}$ to apply knowing $\mathbf{x'}_{\mathbf{k}}^{(i)}$
   c) Compute the controls to apply to each socket by performing $\mathbf{u_k} = f^{-1}(\mathbf{u'_k})$

   Input: k, $\mathbf{x_k}$ for the current time slot k.
   Output: $\mathbf{u_k}$, i.e. ON/OFF control decisions to be applied to each plugged battery during time slot k.

produced by the DP Algorithm, which is clearly unfeasible, the computational complexity of the optimization is reduced by applying a set of strategies, based on *Approximate Dynamic Programming* and *Reinforcement Learning*. These strategies are detailed hereafter.

B. Approximate DP: Decoupled problem approximation

The principle at the basis of problem approximation allows to simplify the original problem. To this aim, one of the possible strategies is to decouple variables that may be coupled in the original problem, in order to reduce the dimensionality of the problem. For example, our M-socket BSS, can be thought as a set of M single-socket BSSs. In this case, the algorithm produces, for each socket, a DP lookup table following the classical DP algorithm. Furthermore, the cost is also decoupled into a vector $\mathbf{J_k}(\mathbf{x'_k})$, and the DP algorithm for each component becomes:

$$\mathbf{J}_{\mathbf{k}}^{*(i)}\left(\mathbf{x}_{\mathbf{k}}^{\prime\,(i)}\right) = \min_{\mathbf{u}_{\mathbf{k}}^{\prime\,(i)} \in U_k\left(\mathbf{x}_{\mathbf{k}}^{\prime\,(i)}\right)} \left\{ \mathop{\mathbb{E}}_{\mathbf{w}_{\mathbf{k}}^{\lambda\,(i)}, \omega_k^{PV}} \left[g_k^{(i)}\left(\mathbf{x}_{\mathbf{k}}^{\prime\,(i)}, \mathbf{u}_{\mathbf{k}}^{\prime\,(i)}, \mathbf{w}_{\mathbf{k}}^{\lambda\,(i)}, \frac{\omega_k^{PV}}{M}\right) + \mathbf{J}_{\mathbf{k+1}}^{*(i)}\left(f\left(\left[\mathbf{x'_k} - \mathbf{u'_k}\right]^{+}\right)^{(i)}\right)\right]\right\} \quad (24)$$

Note that in our case the power derived from the PV panel is simply distributed uniformly among all of the sockets, hence excluding the possibility of further optimization through an intelligent distribution of the produced renewable energy. The workflow of the controller is detailed by Algorithm 1.

C. Reinforcement Learning: One step-lookahead + Multi-agent policy iteration + Decoupled problem approximation

Under Approximate DP, lookup tables are computed for each socket at the beginning of the day, to support at any time step the decision process about whether the battery plugged to a given socket should be charged or not. Conversely, under the RL approach the charge scheduling decision at each time step is taken not only based on the socket specific control

Algorithm 2 One step-lookahead + Decoupled problem approximation.

1) At the beginning of the day, for each socket compute M lookup tables running the DP algorithm.
   Input: p_k, ω_k^{λ}, ω_k^{PV}, a_k for any time slot in a day.
   Output: *Control table* and *cost-to-go table* per each socket.
2) At each time step k:
   a) Re-order the states computing $\mathbf{x'_k} = f(\mathbf{x_k})$
   b) Find $\mathbf{u'_k}$ minimizing the cost in Equation (25) by brute-force (this step may take some time to finish)
   c) Compute the controls to apply to each socket by performing $\mathbf{u_k} = f^{-1}(\mathbf{u'_k})$

   Input: k, x_k, p_k, ω_k^{λ}, ω_k^{PV}, a_k for the current time slot k.
   Output: $\mathbf{u_k}$, i.e. ON/OFF control decisions to be applied to each plugged battery during time slot k.

tables generated at the beginning of the day, but also based on additional optimization procedures that are performed at each time step.

One step lookahead strategies are halfway between simple problem approximation and solving the complete problem. Since solving the complete problem is infeasible due to its dimensionality, these strategies first perform an on-line cost optimization over all the possible controls to apply, given the current state of the system, and then an approximation is performed for the cost-to-go function, potentially through problem approximation.

At each time step the algorithm computes the following:

$$\mathbf{u'_k}^{*}\left(\mathbf{x'_k}\right) = \mathop{\operatorname{argmin}}_{\mathbf{u'_k} \in U_k\left(\mathbf{x'_k}\right)} \left\{ \mathop{\mathbb{E}}_{\mathbf{w}_{\mathbf{k}}^{\lambda}, \omega_k^{PV}} \left[g_k\left(\mathbf{x'_k}, \mathbf{u'_k}, \mathbf{w}_{\mathbf{k}}^{\lambda}, \omega_k^{PV}\right)\right] + \widetilde{J}_k\left(\mathbf{x'_{k+1}}\right)\right\} \quad (25)$$

where $\widetilde{J}_k(\mathbf{x'_{k+1}})$ represents the cost-to-go approximation, that can be computed exploiting the cost-to-go tables obtained by running the Dynamic Programming at the beginning of the day for the decoupled system and summing the associated cost for each socket as shown in the following equation:

$$\widetilde{J}_k\left(\mathbf{x'_{k+1}}\right) = \sum_i \mathop{\mathbb{E}}_{\mathbf{w}_{\mathbf{k}}^{\lambda\,(i)}} \left[\mathbf{J}_{\mathbf{k+1}}^{*(i)}\left(f\left(\left[\mathbf{x'_k} - \mathbf{u'_k}\right]^{+}\right)^{(i)}\right)\right] \quad (26)$$

In this way an approximated overall cost-to-go of the system is obtained, after applying a control $\mathbf{u'_k}$. At each time step, all the possible actions are evaluated starting from the actual state of the system, which are $|u|^M$ possible controls. These kind of algorithms enable a collaboration between the agents, since for at least one of the time steps the complete problem is considered.

Summarizing, the algorithm combining the One step-lookahead approach and the Decoupled problem approximation would work as reported in Algorithm 2. Considering the previously introduced scenario, featuring $|u| = 2$ and $|M| = 20$, at least $\sim 10^6$ operations are needed for each time step, which are much fewer than those required to solve the complete problem through Dynamic Programming. Nevertheless, the process is still extremely complex and time consuming. In

Algorithm 3 One step-lookahead + Multi-agent policy iteration + Decoupled problem approximation.

1) Compute M lookup tables running the DP algorithm for each one of the sockets at the beginning of the day
   Input: p_k, ω_k^{λ}, ω_k^{PV}, a_k for any time slot in a day.
   Output: *Control table* and *cost-to-go table* per each socket.
2) At each time step k:
   a) Re-order the states computing $\mathbf{x}'_{\mathbf{k}} = f(\mathbf{x}_{\mathbf{k}})$
   b) Initialize the optimal control as $\mathbf{u}'^{*}_{\mathbf{k}} := 0$
   c) For a number of iterations N_{IT} or until convergence:
      i) For each socket i:
         A) Find a new $\mathbf{u}'^{*(i)}_{\mathbf{k}}$ by minimizing the cost in Equation (27), where the rest of the components are kept fixed to $\mathbf{u}'^{*}_{\mathbf{k}}$.
         B) Update the component i of the optimal control $\mathbf{u}'^{*}_{\mathbf{k}}$
   d) Compute the controls to apply to each socket by performing $\mathbf{u}_{\mathbf{k}}{}^{*} = f^{-1}\left(\mathbf{u}'^{*}_{\mathbf{k}}\right)$

   Input: k, x_k, p_k, ω_k^{λ}, ω_k^{PV}, a_k for the current time slot k.
   Output: $\mathbf{u}^{*}_{\mathbf{k}}$, i.e. ON/OFF control decisions to be applied to each plugged battery during time slot k.

similar problems where the control variable is a vector that can be thought as M-agents, each operating on its own socket, the control can be evaluated one agent at a time to further reduce the search space for the optimal control. In this case, to reduce the cost, the algorithm first starts with a base policy, for example applying $\mathbf{u}'_{\mathbf{k}} = 0$, and, subsequently, it minimizes the cost *one-agent-at-a-time* instead of *all-agents-at-once*:

$$\mathbf{u}'^{*}_{\mathbf{k}}\left(\mathbf{x}'_{\mathbf{k}}\right) = \underset{\mathbf{u}'_{\mathbf{k}} \in U_k\left(\mathbf{x}'_{\mathbf{k}}\right)}{\text{seq. argmin}} \left\{ \underset{\mathbf{w}_{\mathbf{k}}^{\lambda}, \omega_k^{PV}}{\mathbb{E}} \left[g_k\left(\mathbf{x}'_{\mathbf{k}}, \mathbf{u}'_{\mathbf{k}}, \mathbf{w}_{\mathbf{k}}^{\lambda}, \omega_k^{PV}\right) \right] + \sum_i \underset{\mathbf{w}_{\mathbf{k}}^{\lambda(i)}}{\mathbb{E}} \left[\mathbf{J}^{*(i)}_{\mathbf{k+1}} \left(f\left(\left[\mathbf{x}'_{\mathbf{k}} - \mathbf{u}'_{\mathbf{k}}\right]^{+} \right)^{(i)} \right) \right] \right\} \tag{27}$$

where *seq.* stands for *sequentially*, to indicate the fact that the minimization is performed one component of $\mathbf{u}'_{\mathbf{k}}$ at a time. Under this approach the complexity is reduced to $|u| \cdot |M| = 40$ operations, for each iteration of the algorithm, which hence becomes linear on $|u|$.
The complete Reinforcement Learning based approach, based on One step-lookahead, Multi-agent policy iteration and Decoupled problem approximation, is detailed in Algorithm 3.

D. Heuristic control algorithm

We now introduce a heuristic based algorithm that implements a charging scheduling strategy. This algorithm is adopted as a baseline reference, against which the performance of the charge scheduling algorithms based on Dynamic Programming and Reinforcement Learning is compared. The heuristic control algorithm, derived from our previous work [13], relies on battery charging postponement to reduce cost. Besides minimizing cost, the algorithm aims at favoring the utilization of the locally produced renewable energy. Based on this approach, the charge of some batteries at the BSS can be postponed by up to T_{max} if the cost for the energy drawn from the grid is expected to be more convenient in the next future. In particular, when (i) no RE is currently produced and (ii) a new EV arrives at the BSS or one of the batteries under charge at the BSS achieves the target charging level, an algorithm is triggered to select up to F batteries at the BSS, whose charge is postponed by a period t_r, with $t_r \leq T_{max}$, as long as the following conditions hold:

$$c^G(t+t_r) = \min_{\forall i \in (0, T_{max}]} c^G(t+i) \tag{28}$$

$$c^G(t) > c^G(t+t_r) \tag{29}$$

where $t+t_r$ corresponds to the time between t and $t+T_{max}$ at which the battery charge, once suspended at time t, must be resumed to observe the minimum value of the cost for the energy drawn from the grid that is required to recharge the considered battery to the desired level, based on the available charging rate, that in turn depends on the current SOC and SOH of the battery. The value of $c^G(t)$ depends on the initial charge level of the battery, on the RE that is produced during the period in which the battery remains under charge, and on the time-varying electricity prices.

E. Exact policy solutions

Exact policy solutions represent an additional benchmark against which the performance of the proposed DP and RL approaches can be compared. These solutions yield the best performance in terms of either cost saving or capability of accomplishing the EV demand. The M-socket BSS problem can be represented as a *Discounted cost problem*, a particular subset of infinite horizon Dynamic Programming problems in which the cost-per-stage is bounded [46], even in case the number of stages tends to infinite. The discounted cost model achieves this aim by rapidly decreasing the contribution of costs in future stages.

The M-socket BSS problem can be exactly solved for the cases in which $\beta = 0$ or $\beta = 1$. Indeed, under $\beta = 0$, i.e., when priority is given to energy cost minimization, only the electrical cost is considered and accumulated in the cost function. Hence, according to [46], the discounted cost can be computed as:

$$J(x_k)_{\beta=0} = \lim_{N \to \infty} \sum_{k=0}^{N-1} \alpha^k \cdot a_k \cdot \left[\sum_i \mathbf{u}'^{(i)}_{\mathbf{k}} - \mathbb{E}\left[\omega_k^{PV}\right] \right]^{+} \tag{30}$$

where $\alpha \in (0,1)$ denotes a discount factor such that, as k increases, the cost contribution corresponding to stage k is progressively decreased. Notice that the cost corresponding to the last stage, $k = N$, does not appear in the equation, since the system is assumed to never reach a final stage, and the cost-per-stage decays exponentially with a factor α. In order to ensure the convergence of this equation, a sufficient condition is to have a bounded cost-per-stage function $g_k(\cdot) < M$, so that the discounted cost is bounded by the decreasing geometric progression $\left\{\alpha^k M\right\}$ [46]. The sequence inside the sum is always positive, and can be minimized if $\mathbf{u}'^{*}_{\mathbf{k}} = 0$, leading to a system in which batteries would never be charged. No other algorithm can yield a better performance in terms of

electricity consumption than $\mathbf{u'_k}^* = 0$, that we name the *Always OFF Algorithm*.

Conversely, considering $\beta = 1$, i.e., giving priority to minimizing EV service losses, the sum becomes:

$$J(x_k)_{\beta=1} = \lim_{N\to\infty} \sum_{k=0}^{N-1} \alpha^k \cdot \mathbb{I}\left\{ f\left(\left[\mathbf{x'_k} - \mathbf{u'_k} \right]^+ \right) > E_0 \right\}^T \cdot \mathbf{p}_\mathbf{k}^\lambda \quad (31)$$

In this case the sequence inside the sum can be minimized if the condition $f\left(\left[\mathbf{x'_k} - \mathbf{u'_k} \right]^+ \right) \leq E_0$ is satisfied. To satisfy this condition for as many time slots as possible, the maximum possible charge rate, leading to $\mathbf{u'_k}^* = \mathbf{u_{max}}$, must be applied. In an ON/OFF control framework, this means that all the batteries would always be charging. In this case, the system evolves according to Equation (6), but under $\mathbf{u'_k}^* = \mathbf{u_{max}}$ only the cost for the following state can be reduced. Hence, no other algorithm performs better in terms of EV service losses than the one based on $\mathbf{u'_k}^* = \mathbf{u_{max}}$, that we call the *Always ON Algorithm*.

VI. Key Performance Indicators

The following metrics are considered to evaluate the performance of the charging scheduling algorithms and their impact on the battery health:

- *Mean grid energy cost* - c_E: mean daily cost for the energy drawn from the electric grid to charge EV batteries at the BSS.
- *Mean missed service probability* - p_G: mean daily probability that an EV cannot be served upon arrival at the BSS due to the lack of batteries with a sufficient charge level to replace the empty EV battery.
- *Normalized grid energy cost* - $\hat{c_E}$: the mean grid energy cost is normalized as:
$$\hat{c_E} = \frac{c_E - c_E^{OFF}}{c_E^{ON} - c_E^{OFF}}$$
where c_E^{ON} and c_E^{OFF} correspond to the mean grid energy cost under the Always-ON and Always-OFF algorithms, respectively.
- *Normalized missed service probability* - $\hat{p_G}$: the mean service loss probability is normalized as:
$$\hat{p_G} = \frac{p_G - p_G^{ON}}{p_G^{OFF} - p_G^{ON}}$$
where p_G^{ON} and p_G^{OFF} represent the values of p_G under the Always-ON and Always-OFF algorithms, respectively. From this metric, we derive a Quality of Service indicator, denoted by Q_G, that is defined as $1 - \hat{p_G}$.
- *Grid energy consumption* - g_E: mean daily energy amount drawn from the grid by the BSS to recharge all the batteries plugged to the BSS sockets.
- *Mean battery stay time* - t_S: mean time spent by a battery in the BSS.
- *Mean switching occurrences* - n_S: mean number of times that the charging of a battery is suspended and resumed during the period in which the battery remains plugged to a BSS socket until complete recharge.
- *Renewable energy reward* - r: monetary reward received by the BSS operator for the locally produced extra renewable energy that is injected and sold to the SG during the observation period, assuming that it is bought by the SG operator at half the electricity selling price, as proposed in the literature [47].
- *Cost per service* - c_S: mean cost required to recharge a battery plugged to a BSS socket, computed as $c_S = \frac{c_E}{v}$, where v is the number of EVs arriving at the BSS and successfully receiving a fully recharged battery during the observation period.
- *Overall Yearly Cost* - C^Y: it is the system cost per year incurred by the BSS service provider. This cost includes both the operational cost (OPEX) due to the energy bought from the power grid during the BSS operation and the OPEX due to the management cost for replacing the batteries at the end of their lifetime, and it takes into account the monetary revenues received from the amounts of renewable energy that are sold back to the Smart Grid. Note that the cost for battery replacement is included in the computation of C^Y, since the considered scenario is based on the paradigm of *Battery-as-a-Service*. Indeed, the BSS system operator is the owner of the batteries, and the management cost for the replacement of a battery unit at the end of its lifetime is paid by the BSS operator. This cost is computed as follows:
$$C^Y = \left(\frac{C^B \cdot C^N}{T^B} \right) \cdot N^B + C^O - r \quad (32)$$
where C^B is the cost per 1 kWh of battery capacity, T^B corresponds to the expected lifetime of each battery, N^B is the average number of battery units -corresponding to a subset of the pool of batteries owned by the service provider- that are served daily by each BSS belonging to the system operator, whereas C^O is the yearly operational cost for the energy demand from the grid, and r is the yearly renewable energy reward.

VII. Performance analysis

For the performance analysis we consider a BSS with $M = 20$ sockets, aiming at a system dimensioning suitable to satisfy the swap demand during peak periods, according to [13]. We assume a local RE supply consisting of a set of photovoltaic panels with capacity 500 kWp, based on [13]. 100 possible states are assumed for x_k, i.e., the discrete variable that represents the energy required to fully recharge each battery based on the current charge level, with x_{100} corresponding to the full battery capacity C. Batteries can be released from the BSS and made available for an arriving EV as soon as their charge level achieves $B_{th} \cdot C$. Considering that short-medium range routes in a urban scenario may not necessarily require a fully recharged battery, B_{th} is set to 0.9, representing a reasonable threshold to limit the periods of slower battery charging observed under SOC higher than 80%, hence reducing the probability of service unavailability during peak demand [34]. Furthermore, D_{max} is reasonably set to 0.8 to preserve battery lifetime [45]. EVs are assumed to arrive at the BSS with a random battery charge level L uniformly distributed as $\mathcal{U}(0.2, 0.4)$. Furthermore, the age of an EV battery upon arrival at the BSS is extracted randomly from a uniform distribution, $a_B = \mathcal{U}(0, a_{B_{max}})$, with $a_{B_{max}}$.

We set $a_{B_{max}}$ to 680 days to account for a relatively recent pool of batteries, in accordance to [36], that we adopt to derive f_{SOH}. The battery temperature is either T=25°C or T=40°C with equal probability, i.e., $p = 0.5$. We assume a battery cost, C^B, of 98.5 €/kWh [48], and a conservative value of battery lifetime, T^B, of 8 years [49]. Finally, N^B is set equal to the average daily number of EVs arriving at the BSS for battery replacement, i.e., 140, under the assumption that a battery undergoes a swapping operation no more than once a day. This value represents an intentionally conservative setting for N^B, that might lead to overestimate the battery replacement cost. The time step duration is set to $\Delta t = 5$ minutes. Our results are obtained simulating the BSS operation during the first two weeks of the year, unless differently specified.

A. Dynamic Programming control tables

We first analyze the output of the Finite-Horizon DP Algorithm, that is obtained based on the input information about the EV arrival rate, the expected solar energy production, and the price for the energy drawn from the grid. The output consists in the control table and the cost-to-go table. The first table indicates the optimal control to apply $u_k^*(x_k)$ at a given time and at a given state (k, x_k), whereas the latter indicates, for the same state variable, the cost-to-go function value $J_k^*(x_k)$.

Figure 3a shows the control table derived for the fifth most charged battery in the BSS, under $\alpha = 0.9$ and $\beta = 0.9$. Based on this table, at each time step k and depending on the depletion level of the considered battery, i.e. depending on x_k, the decision of activating (ON, represented by either red squares or orange squares, to indicate whether the activation decision entails a higher or lower charging rate, respectively, depending on the battery SOC) or deactivating (OFF, gray squares) the charging process during the current time step is taken. At the beginning of the day, the charging process is rarely activated, since other fully charged batteries are likely available at the BSS and the EV battery swapping demand is low. For k between 50 and 100, the separation boundary between the areas representing the ON and OFF states, respectively, shows a less steeper profile under values of x_k higher than 75, corresponding to lower SOC of the battery. This behavior is clearly related to the lower charging rate observed when SOC is lower than 25% and to the higher energy demand entailed by high values of x_k, hence making less convenient the activation of the charging process when the RE production is still low, and slowing down the reaching of states x_k corresponding to higher levels of battery charge. A rather complex behavior of the scheduling can be evinced from the irregular shapes of the gray, orange and red areas in the control table. The profile of the EV arrival rate in Fig. 2 shows that the highest values of EV arrival rate are observed during three main periods of the day. These peaks determine corresponding raises of the cost-to-go function values, especially for low levels of charge, as can be observed from Fig. 3b that reports the cost-to-go for all the possible combinations of (k, x_k). Consistently, from the control table it can be observed that a battery at low charge level starts its charging process at around $k = 50$ in order to be ready for the first arrival rate peak at $k = 100$.

B. Effects of parameter configuration

The hyper-parameter α, as shown in Equation (30), regulates the '*visibility*' of the Finite-Horizon DP algorithm. As α gets closer to 1, the step cost-function values for future time steps are weighted more heavily by the algorithm. Fig. 4 depicts the values of the cost-to-go (the color palette from yellow to purple is used to represent higher to lower values) for different combinations of k (x-axis) and x_k (y-axis), setting $\beta = 0.9$, and under $\alpha = 0.95$ (Fig. 4a) and $\alpha = 0.50$ (Fig. 4b). The cost-to-go function tends to raise around the traffic peaks. However, under higher setting of α, the raise of the cost-to-go function starts earlier, i.e., at lower values of time step k, with respect to the case of smaller α. Furthermore, the period by which the raise of the cost-to-go function is anticipated under higher α tends to progressively become longer as the value of x_k increases, hence leading to a more scattered distribution of gradually varying cost-to-go values around the peak periods. Conversely, under smaller α, the visibility of the algorithm rapidly decreases and the cost-to-go function tends to assume more concentrated values. In this work, α is set to 0.9. Assuming that once $\alpha^k < 0.1$ can be neglected, this setting of α entails a visibility for the algorithm of about:

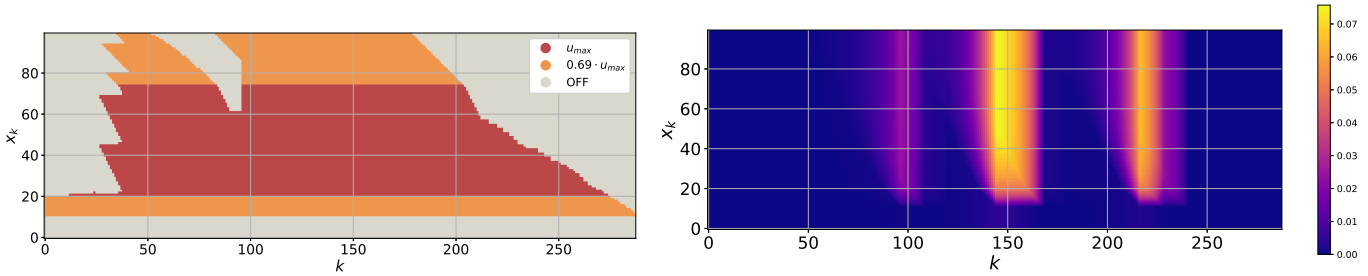
$$k_H = \log_\alpha (0.1) \approx 22 \tag{33}$$

Considering a time step of $\Delta t = 5$ min, the algorithm features a visibility of about $\Delta t \cdot k_H = 109$ min, amounting up to nearly 2 hours of visibility, which corresponds to the recharging time of an empty battery.

The hyper-parameter β regulates the trade-off between the need for purchasing the electrical energy from the grid and the likelihood of failing to accomplish the EV battery replacement demand. Fig. 5 reports the control tables obtained through Finite-Horizon DP Algorithm for different values of β, i.e., $\beta = 0.9$ (Fig. 5a) and $\beta = 0.2$ (Fig. 5b). As β gets closer to 1, the probability of not being able to serve an EV becomes larger in the cost function, hence the algorithm tends to more frequently activate the battery charge process. Almost at any time step k, we observe at least a state x_k that triggers the activation of the battery charge process. Conversely, as β reduces, the algorithm prioritizes not buying energy from the grid, so that the socket is activated to charge the battery only when solar energy is available.

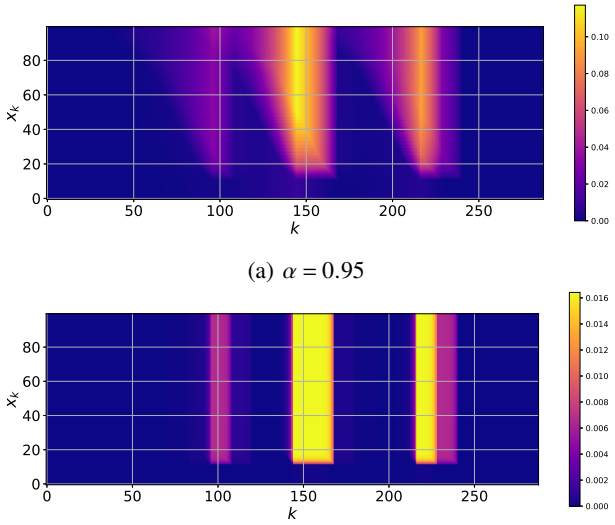
C. Problems with Finite-Horizon DP

The DP Algorithm solves the optimization problem starting at $k = N$, that in our case represents the end of the day. In order to run the algorithm, the starting condition, $g_N(x_k)$, must be known for each possible final state x_k. In our system, this condition is set to $g_N(x_k) = 0$, since no particular cost is assigned to the state of the battery by the end of the day, as it can be observed from the cost-to-go table depicted in Fig. 6a. However, under this settings, as the end of the day approaches the charge of some batteries is deactivated, depending on the value of x_k, since the charging rate would be not sufficient to fully satisfy the demand E_0 during the remaining time slots until the end of the day. This can be clearly evinced from Fig. 6b that reports the control table obtained for one of the BSS sockets assuming $\beta = 1$, that corresponds to the Always



(a) (b)

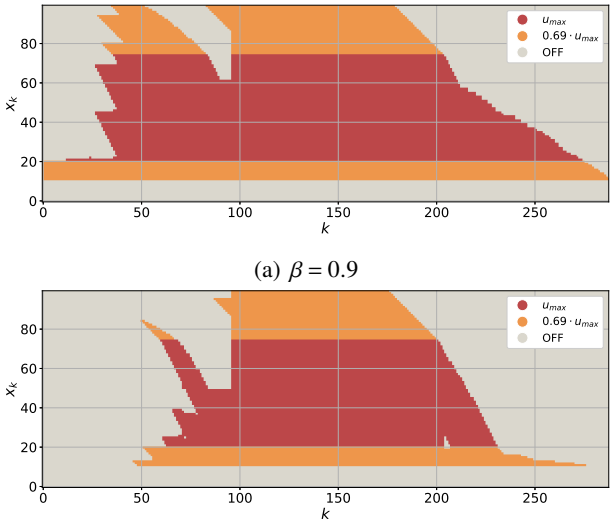
Fig. 3: Control table (a) and Cost-to-go values table (b) obtained under Finite-Horizon DP Algorithm for the fifth most charged battery ($\alpha = 0.9$, $\beta = 0.9$).



(a) $\alpha = 0.95$

(b) $\alpha = 0.50$

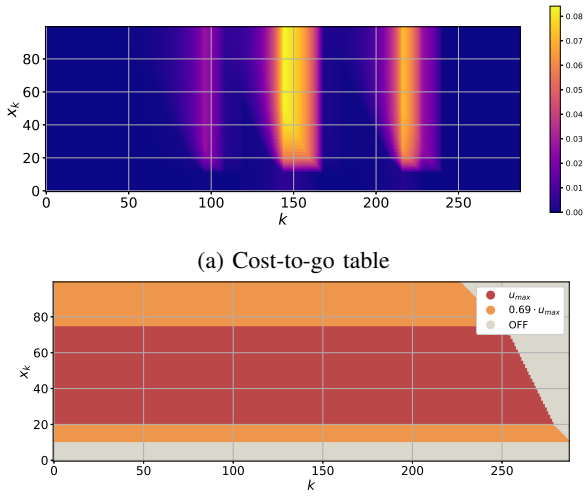
Fig. 4: Cost-to-go tables obtained through Finite-Horizon DP Algorithm for different values of α.



(a) $\beta = 0.9$

(b) $\beta = 0.2$

Fig. 5: Control tables obtained through Finite-Horizon DP Algorithm for different values of β.



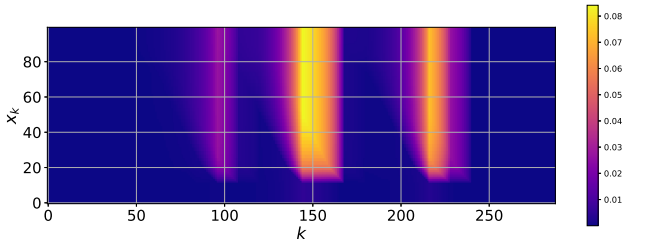
(a) Cost-to-go table

(b) Control table

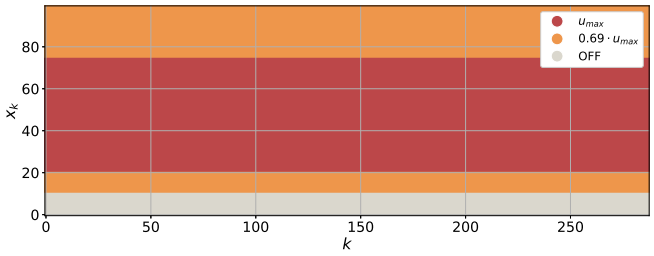
Fig. 6: Cost-to-go and control table obtained through Finite-Horizon DP Algorithm under $\alpha = 0.9$ and $\beta = 1$.

ON algorithm. Similarly to Fig. 3a and Fig. 5, we observe that the activation of the charging process starts to become less convenient much earlier if the SOC is lower than 25%, due to the slower battery charging rate.

To overcome the boundary conditions effect of Finite-Horizon DP, a value iteration based approach is considered to implement an Infinite-Horizon DP algorithm. Indeed, a finite horizon limited to one day is not reasonable, considering that the BSS system is under operation all through the year. We adopt the value iteration approach, assuming that input curves are interpreted as stationary at the beginning of each day. The cost-to-go table reported for this case in Fig. 7 shows that cost-to-go values are all non-zero, unlike the Finite-Horizon DP case depicted in Fig. 6a in which the cost-to-go is set to zero for $k = N$, i.e., at the end of the day. Furthermore, cost-to-go values at $k = 0$ are the same observed at $k = N$. Finally, the control table shown in Fig. 7b does not present the boundary effect highlighted in Fig. 6 for Finite-Horizon DP, resulting in the same control policy both at the beginning of the day and at the end of the day.



(a) Cost-to-go table



(b) Control function

Fig. 7: Cost-to-go and control table obtained through Infinite-Horizon DP Algorithm (through value iteration) for different values for $\alpha = 0.9$ and $\beta = 1$.

VIII. Parameter tuning to trade off cost and QoS

The tuning of hyper-parameter β allows to trade off cost and Quality of Service, either assigning a higher weight to the cost for the energy drawn from the grid or to the missed service probability, i.e., the probability that the empty battery of an EV arriving at the BSS cannot be replaced with a fully charged battery. Fig. 8a shows the normalized values of the mean grid energy cost, $\hat{c_E}$ (blue curve), and of the Quality of Service, in terms of Q_G (orange curve), for increasing values of β under ADP algorithm. Under $\beta = 0$, only the electricity cost is taken into account, corresponding to the strategy based on the Always OFF algorithm, that minimizes the grid energy cost at the price of a huge missed service probability, that results maximized. Conversely, setting $\beta = 1$, an Always ON algorithm is applied maximizing grid costs and QoS, entailing a service loss probability of 0.013. By only slightly decreasing β to 0.99, a sharp reduction of the cost of about 11% is observed, at the price of a limited impairment of QoS, of just 3% (corresponding to $p_G = 0.042$). As β is further reduced, the cost tends to reduce more and more gradually, in parallel with a progressively slower degradation of the QoS. This trend leaves margin to accurately select the value of β corresponding to the proper working point, depending on the desired performance targets.

To better compare the system performance under different scheduling algorithms, Fig. 8b reports the Quality of Service, in terms of Q_G, versus the normalized grid energy cost, $\hat{c_E}$, under several algorithms (including the benchmark algorithms), considering different values of β (varying between 0 and 1), that are represented by the different points within a given algorithm (with increasing values of β from the left to the right side of the plot).

Although β can be set to any value from 0 to 1, the most interesting working zone is actually close to the Always ON performance (β=1). Indeed, the BSS operator is likely interested in limiting cost savings to favor the satisfaction of the customer demand, rather than highly reducing costs at the price of dissatisfied customers, hence selecting $\beta \sim 1$. Fig. 8c shows a zoomed version of Fig. 8b, focusing on the working zone of interest, i.e., close to the Always ON performance. The best fitting line passing through $(1, 1)$ is traced for the Heuristic algorithm, whose performance is considered as baseline reference. An equally clear linear trend cannot be observed under RL and DP, likely reflecting a deeper influence yielded by the non-linear behavior of the BSS system on the overall performance of these algorithms. Results show that both RL and DP significantly outperform Heuristics in providing better QoS at a lower cost, under any setting of β. In general, RL features the best performance in terms of both energy cost and QoS, hence guaranteeing the lowest missed service probability under any budget constraint. However, under high values of β, DP performs slightly better than RL, although the difference is almost negligible.

TABLE III: Battery statistics

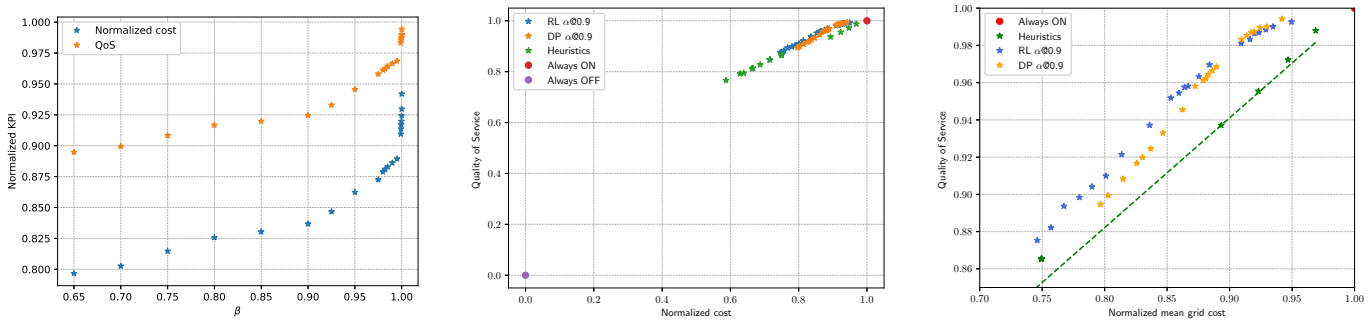
Algorithm	$\mathbb{E}(t_S)$ [min]	$\max(t_S)$ [min]	$\mathbb{E}(n_S)$	$\max(n_S)$
Heuristics	209	1145	1.40	3
DP	137	916	1.48	7
RL	135	845	1.89	8

IX. Beyond system performance indicators

We now investigate how the proposed scheduling algorithms affect some metrics that may result relevant for the battery health. Two case studies are presented to evaluate the RL performance over a long period of time covering one year of BSS operation.

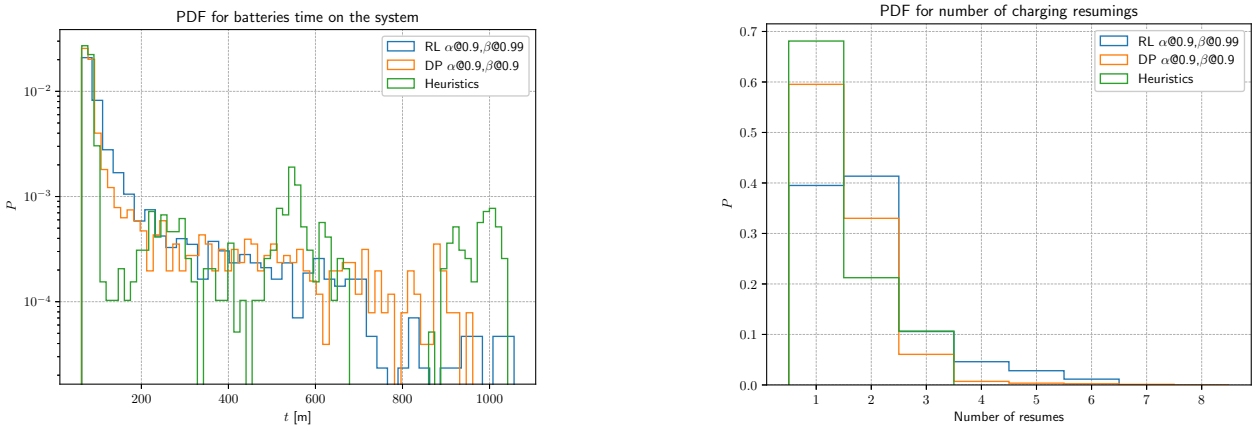
a) Battery health: We compare how the different algorithms affect the average time spent by a battery in the BSS before completing its charge, t_S, and the average number of times that the charging process of a battery that is under charge at a socket of the BSS is deactivated and resumed, n_S.

Since ADP and RL algorithms operate sorting batteries according to the state of charge to take charging scheduling decisions, some batteries might remain in the BSS for extremely long times. Fig. 9a shows the Probability Density Function (PDF) of the time spent in the system by a battery for each algorithm. DP Algorithm features the shortest tail, followed by the Heuristics and the DP Algorithm. As confirmed by the results shown in Table III, that reports the average and maximum values of t_S and n_S obtained from simulations under the various algorithms, while both DP and RL guarantee an average time spent in the system that is more than 35% lower than under the Heuristic, RL provides the best performance even in the worst case, yielding the shortest period of time spent in the BSS, that amounts to about 14 hours. Conversely, up to more than 15 and 19 hours are required to fully recharge a battery under DP and the Heuristic algorithm, respectively. The low values of average and maximum time spent in the system by a battery guaranteed by RL represent a significant



(a) (b) (c)

Fig. 8: Performance in terms of cost and Quality of Service for varying β, under Infinite Horizon DP (a) and under different algorithms (b-c), with (c) representing a zoomed area of (b).



(a) t_S (b) n_S

Fig. 9: Probability Density Function (PDF) of the mean time spent by a battery in the system, t_S (a), and PDF of the number of times that the charging is suspended/resumed during the time in which the battery is in the BSS, n_S (b).

advantage for the BSS operator, since fully recharged batteries are more frequently available to satisfy EV demand.

Frequently interrupting and resuming the battery charging process may significantly affect battery deterioration, impairing its lifetime [50]. Furthermore, the quality of power delivered to grid-connected customers may be reduced by frequent switching, especially in case of wide-spread implementation [50]. To investigate this aspect, we hence evaluate the average number of times that charging is suspended and resumed during the permanence of a battery in the BSS, $\hat{nS}$, whose PDF is reported in Fig. 9b. The Heuristics algorithm represents the most conservative case, featuring lower frequency of battery charge resuming, as confirmed by the statistics reported in Table III for the various algorithms. Conversely, RL yields the highest switching frequency, with a charging process that can be interrupted and resumed up to 8 times during the battery permanence at the BSS, which is almost three-fold higher than the maximum value provided by the Heuristic algorithm. The higher reactivity of RL algorithm is coupled with the shorter time spent by the battery under charge in a BSS, as shown in Table III. Overall, although RL may lead to relatively faster battery degradation, this downside is compensated by the advantages offered by a dynamic scheduling algorithm in terms of cost and Quality of Service.

Clearly, given its complexity, RL is characterized by a huge computational time, amounting up to almost 190 ms, far larger than for the DP. Nevertheless, the RL execution time is by significantly lower than $\Delta t = 5$ min, that represents the discrete time period within which real time charging scheduling decisions are taken. Hence, despite being the most complex algorithm, RL can be feasibly adopted in a real time scenario.

b) Case studies: Finally, in order to better compare the performance improvement introduced by the Reinforcement Learning algorithm with respect to the Heuristics one, we report two case studies in which a one year-long simulation is performed.

1) Comparing algorithms under similar grid energy demand: In the first case study, for the Heuristics algorithm, F_{max} is set to 17 and $T_{max} = 40$ hours. Under RL, hyperparameters are tuned to $\alpha = 0.9$ and $\beta = 0.99$. The setting of $\beta = 0.99$ allows to match the same level of electricity consumption from the grid observed under the Heuristic algorithm. Table IV highlights the main performance metrics obtained under the two tested algorithms. All the metrics are improved under RL. Although under both the Heuristic and

TABLE IV: Performance metrics under different algorithms, with configuration settings yielding the same energy demand.

Metric	Control algorithm		Improvement
	Heuristic	RL	
p_G	0.084	0.048	-42.9%
c_S [€]	0.249	0.226	-9.34%
g_E [Wh]	$7.77 \cdot 10^5$	$7.75 \cdot 10^5$	-0.25%
r [k€]	5.778	6.096	+5.50%
C^Y [k€]	42.910	41.417	-3.48%

RL algorithms the missed service probability is below 0.09, more than 40% additional EVs can be successfully served under RL. Furthermore, despite a negligible reduction of the energy drawn from the grid, RL provides more than 9% cost per service decrease with respect to the Heuristic approach, showing that the RL algorithm allows to utilize the electrical energy bought from the SG in a smarter and more efficient way and to slightly increase the reward for the amount of energy that can be sold to the SG. Finally, we highlight that even including the battery management expenditures, the yearly cost incurred by the service provider, C^Y, is reduced by about 3.5%.

2) Comparing algorithms under the same QoS target: In the latter case study, we change the configuration of the Heuristic algorithm to achieve the same QoS of RL, under the constraint of $p_G \leq 0.05$. F_{max} is set to 17 and T_{max} = 36.8 hours. The corresponding performance metrics reported in Table V show that RL reduces the energy demand from the grid by almost 8%, hence entailing a relevant reduction of the carbon footprint due to the lower consumption of non renewable energy. Furthermore, on the one hand RL reduces the cost per service by more than 18% (about twice the reduction obtained in the first case study), still providing the same QoS level. On the other hand, the revenues derived from the energy sold to the Smart Grid are only slightly reduced under RL, and the yearly cost incurred by the service provider, C^Y, which includes the battery management cost, is decreased by almost 7%.

The presented case studies clearly show that RL significantly outperforms Heuristic algorithm in reducing operational cost, also due to timely and more effective scheduling decisions that take advantage of the periods of lower electricity prices, and in providing a greener operation of the BSS system, still guaranteeing the satisfaction of the desired QoS requirements.

X. Conclusion

In this paper we address the EV battery charging scheduling problem in a renewable powered BSS, designing two adaptive algorithms based on Approximate DP and RL, that aim at modulating and dynamically adapting the scheduling of the battery charging process to the stochastic nature of the system. Our results show that both approaches are effective and significantly improve Quality of Service at a lower cost with respect to benchmark approaches. However, the RL based approach achieves the best level of Quality of Service under any budget constraint, allowing to decrease the probability of not satisfying the EV demand by up to more than 40% with respect to

TABLE V: Performance metrics under different algorithms, under the same QoS target.

Metric	Control algorithm		Improvement
	Heuristic	RL	
p_G	0.048	0.048	0%
c_S [€]	0.277	0.226	-18.45%
g_E [Wh]	$8.38 \cdot 10^5$	$7.75 \cdot 10^5$	-7.56%
r [k€]	5.877	6.096	+3.73%
C^Y [k€]	44.243	41.417	-6.82%

Heuristic approaches, and to yield a significant cost reduction of almost 20%. and a fine tuning of the scheduling algorithm hyper-parameters is fundamental to properly trade off cost and Quality of Service requirements according to business needs, and to provide a greener operation of the BSS system. Clearly, the optimal deployment of BSS charging scheduling techniques cannot overlook their effects on the battery health, that may accelerate the process of battery degradation and impair the BSS management cost. Future work is required to integrate additional performance and sustainability goals in the deployment of complex multi-objective charging scheduling techniques, that trade off possible conflicting business needs, SG operator requirements and feasibility constraints, so as to facilitate a wider penetration of BSS technology for urban e-mobility.

References

[1] IEA, "Key World Energy Statistics 2020," Tech. Rep., 08 2020.

[2] "Global EV Outlook 2023, IEA, Paris," 2023. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2023

[3] P. Chakraborty, R. N. Dizon-Paradis, and S. Bhunia, "Savior: A sustainable network of vehicles with near-perpetual mobility," *IEEE Internet of Things Magazine*, vol. 6, no. 2, pp. 108–114, 2023.

[4] M. Straka, P. De Falco, G. Ferruzzi, D. Proto, G. Van Der Poel, S. Khormali, and L. Buzna, "Predicting popularity of electric vehicle charging infrastructure in urban context," *IEEE Access*, vol. 8, pp. 11 315–11 327, 2020.

[5] "FOTW #1272 (January 9, 2023)." Vehicle Technologies Office (U.S. Department of Energy), 2023, www.energy.gov/eere/vehicles/articles/fotw-1272-january-9-2023-electric-vehicle- battery-pack-costs-2022-are-nearly [Accessed: 2023.06.13].

[6] J. Deng, C. Bae, A. Denlinger, and T. Miller, "Electric vehicles batteries: Requirements and challenges," *Joule*, vol. 4, no. 3, pp. 511–515, 2020.

[7] S. Rivera, S. M. Goetz, S. Kouro, P. W. Lehn, M. Pathmanathan, P. Bauer, and R. A. Mastromauro, "Charging infrastructure and grid integration for electromobility," *Proceedings of the IEEE*, vol. 111, no. 4, pp. 371–396, 2023.

[8] X. Chen, K. Xing, F. Ni, Y. Wu, and Y. Xia, "An electric vehicle battery-swapping system: Concept, architectures, and implementations," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 5, pp. 175–194, 2022.

[9] M. A. H. Rafi and J. Bauman, "A comprehensive review of dc fast-charging stations with energy storage: Architectures, power converters, and analysis," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 2, pp. 345–368, 2021.

[10] S. S. Sayed and A. M. Massoud, "Review on state-of-the-art unidirectional non-isolated power factor correction converters for short-/long-distance electric vehicles," *IEEE Access*, vol. 10, pp. 11 308–11 340, 2022.

[11] H. Wu, "A survey of battery swapping stations for electric vehicles: Operation modes and decision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 163–10 185, 2022.

[12] T. Zhang, X. Chen, Z. Yu, X. Zhu, and D. Shi, "A monte carlo simulation approach to evaluate service capacities of ev charging and battery swapping stations," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3914–3923, 2018.

[13] D. Renga, G. Centonze, and M. Meo, "Renewable powered battery swapping stations for sustainable urban mobility," in *2022 IEEE International Smart Cities Conference (ISC2)*, 2022, pp. 1–7.

[14] T. Zhang, X. Chen, B. Wu, M. Dedeoglu, J. Zhang, and L. Trajkovic, "Stochastic modeling and analysis of public electric vehicle fleet charging station operations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9252–9265, 2022.

[15] A. A. Shalaby, M. F. Shaaban, M. Mokhtar, H. H. Zeineldin, and E. F. El-Saadany, "A dynamic optimal battery swapping mechanism for electric vehicles using an lstm-based rolling horizon approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 218–15 232, 2022.

[16] S. Rajkumar, P. Nagaveni, A. Amudha, M. Siva Ramkumar, G. Emayavaramban, and T. Selvaganapathy, "Optimizing ev charging in battery swapping stations with cso-pso hybrid algorithm," in *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, 2023, pp. 1566–1571.

[17] J. Zheng, T. Xie, F. Liu, W. Wang, P. Du, and Y. Han, "Electric vehicle battery swapping station coordinated charging dispatch method based on cs algorithm," in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2017, pp. 150–154.

[18] J. Xue, C. Yan, D. Wang, J. Wang, J. Wu, and Z. Liao, "Adaptive dynamic programming method for optimal battery management of battery electric vehicle," in *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, 2020, pp. 65–68.

[19] J. Jin and Y. Xu, "Optimal policy characterization enhanced actor-critic approach for electric vehicle charging scheduling in a power distribution network," *IEEE Trans. on Smart Grid*, vol. 12, no. 2, pp. 1416–1428, 2021.

[20] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, 2019.

[21] F. L. D. Silva, C. E. H. Nishida, D. M. Roijers, and A. H. R. Costa, "Coordination of electric vehicle charging through multiagent reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2347–2356, 2020.

[22] Y. Chu, Z. Wei, X. Fang, S. Chen, and Y. Zhou, "A multiagent federated reinforcement learning approach for plug-in electric vehicle fleet charging coordination in a residential community," *IEEE Access*, vol. 10, pp. 98 535–98 548, 2022.

[23] H. Li, Z. Wan, and H. He, "Constrained ev charging scheduling based on safe deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2427–2439, 2020.

[24] L. Yan, X. Chen, J. Zhou, Y. Chen, and J. Wen, "Deep reinforcement learning for continuous electric vehicles charging control with dynamic user behaviors," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5124–5134, 2021.

[25] Q. Xing, Y. Xu, and Z. Chen, "A bilevel graph reinforcement learning method for electric vehicle fleet charging guidance," *IEEE Transactions on Smart Grid*, vol. 14, no. 4, pp. 3309–3312, 2023.

[26] Y. Zhang, M. Li, Y. Chen, Y.-Y. Chiang, and Y. Hua, "A constraint-based routing and charging methodology for battery electric vehicles with deep reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 14, no. 3, pp. 2446–2459, 2023.

[27] K. Preusser and A. Schmeink, "Energy scheduling for a der and ev charging station connected microgrid with energy storage," *IEEE Access*, vol. 11, pp. 73 435–73 447, 2023.

[28] V. Murali, A. Banerjee, and V. G. Venkoparao, "Optimal battery swapping operations using reinforcement learning," in *2019 Fifteenth International Conference on Information Processing (ICINPRO)*, 2019, pp. 1–6.

[29] Y. Liang, Z. Ding, T. Zhao, and W.-J. Lee, "Real-time operation management for battery swapping-charging system via multi-agent deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 14, no. 1, pp. 559–571, 2023.

[30] J. Jin, S. Mao, and Y. Xu, "Optimal priority rule-enhanced deep reinforcement learning for charging scheduling in an electric vehicle battery swapping station," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4581–4593, 2023.

[31] X. Yu, F. Wang, and H. Wang, "Optimal battery swapping and charging strategy considering on-site solar generation," in *2023 IEEE/IAS Industrial and Commercial Power System Asia (ICPS Asia)*, 2023, pp. 1082–1087.

[32] M. Emre, A. Stevens, and D. Naberezhnykh, "Modelling range extension of electric vehicles using dynamic wireless power transfer," 04 2018.

[33] V. M. B. Pereira, J. O. De Sousa, G. C. Fonseca, and R. N. Santos, "An automated framework for lithium battery state of health (soh) analysis," in *2023 IEEE 8th Southern Power Electronics Conference and 17th Brazilian Power Electronics Conference (SPEC/COBEP)*, 2023, pp. 1–8.

[34] J. Mies, J. Helmus, and R. van den Hoed, "Estimating the charging profile of individual charge sessions of electric vehicles in the netherlands," *World Electric Vehicle Journal*, vol. 9, p. 17, 06 2018.

[35] H. Wu, G. K. H. Pang, and X. Li, "A realistic and non-linear charging process model for parking lot's decision on electric vehicles recharging schedule," in *2020 IEEE Transportation Electrification Conference Expo (ITEC)*, 2020, pp. 2–7.

[36] P. Keil, "Aging of lithium-ion batteries in electric vehicles," Ph.D. dissertation, Technische Universität München, 2017, https://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20170711-1355829-1-5 [Accessed: 2024.01.26].

[37] M. Dalmasso, M. Meo, and D. Renga, "Radio resource management for improving energy self-sufficiency of green mobile networks," in *Performance Evaluation Review*, vol. 44, no. 2, Sept 2016, pp. 82–87.

[38] A. P. Dobos, *PVWatts Version 5 Manual*, Sep 2014.

[39] "Gestore Mercati Energetici," https://www.mercatoelettrico.org/En/download/DatiStorici.aspx, [Online; accessed 22 September 2021].

[40] T. G. Alghamdi, D. Said, and H. T. Mouftah, "Profit maximization for evses-based renewable energy sources in smart cities with different arrival rate scenarios," *IEEE Access*, vol. 9, pp. 58 740–58 754, 2021.

[41] O. Hafez and K. Bhattacharya, "Modeling of pev charging load using queuing analysis and its impact on distribution system operation," in *2015 IEEE Power Energy Society General Meeting*, 2015, pp. 1–5.

[42] S. Kabir, A. Shufian, R. Islam, M. M. Islam, M. A. Islam, and M. S. R. Mahin, "Impact of grid-tied battery to grid (b2g) technology for electric vehicles battery swapping station," in *2023 10th IEEE International Conference on Power Systems (ICPS)*, 2023, pp. 1–6.

[43] Z. Wang, P. Jochem, and W. Fichtner, "A scenario-based stochastic optimization model for charging scheduling of electric vehicles under uncertainties of vehicle availability and charging demand," *Journal of Cleaner Production*, vol. 254, p. 119886, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652619347560

[44] S. Park, A. Pröbstl, W. Chang, A. Annaswamy, and S. Chakraborty, "Exploring planning and operations design space for ev charging stations," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, ser. SAC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 155–163.

[45] D. Renga and M. Meo, "Dimensioning renewable energy systems to power mobile networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 366–380, 2019.

[46] D. Bertsekas, *Dynamic Programming and Optimal Control*, ser. Athena Scientific optimization and computation series. Athena Scientific, 2005, no. v. 1. [Online]. Available: https://books.google.it/books?id=-E5qQgAACAAJ

[47] M. Ali, M. Meo, and D. Renga, *Cost Saving and Ancillary Service Provisioning in Green Mobile Networks*. Cham: Springer International Publishing, 2019, pp. 201–224.

[48] B. Venditti, "Visualized: What is the Cost of Electric Vehicle Batteries?" Elements - Visual Capitalist, https://elements.visualcapitalist.com/cost-of-electric-vehicle-batteries/, 2023, [Online; accessed 15 February 2024].

[49] J. Neubauer and A. Pesaran, "A Techno-Economic Analysis of BEV Service Providers Offering Battery Swapping Services," vol. 2, 04 2013.

[50] A. Malhotra, N. Erdogan, G. Binetti, I. D. Schizas, and A. Davoudi, "Impact of charging interruptions in coordinated electric vehicle charging," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016, pp. 901–905.

Daniela Renga is Assistant Professor in the Department of Electronics and Telecommunications at the Politecnico di Torino, Italy. She received the PhD degree in Electronics and Telecommunications Engineering from the same Politecnico in 2018.

Felipe Spoturno received the Electronics Engineering Bachelor Degree from Universidad ORT Uruguay in 2017 and the Master Degree in ICT for Smart Societies from the Politecnico di Torino in 2022. He is working as AI and IoT Solutions Architect at Sense Reply (Turin, Italy).

Michela Meo is Full Professor in the Department of Electronics and Telecommunications at the Politecnico di Torino, Italy. She received the Laurea degree in Electronic Engineering in 1993, and the Ph.D. degree in Electronic and Telecommunications Engineering in 1997, both from the Politecnico di Torino.