Online EV Scheduling Algorithms for Adaptive Charging Networks with Global Peak Constraints

<table>
<thead>
<tr>
<th>Journal:</th>
<th>Transactions on Sustainable Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID</td>
<td>TSUSC-2018-12-0140.R1</td>
</tr>
<tr>
<td>Manuscript Types:</td>
<td>Special Issue on Intersection of Computing and Communication Technologies with Energy Systems</td>
</tr>
<tr>
<td>Keywords:</td>
<td>Electric vehicles, online scheduling, approximation algorithm, competitive analysis</td>
</tr>
</tbody>
</table>
Online EV Scheduling Algorithms for Adaptive Charging Networks with Global Peak Constraints

Bahram Alinia, Mohammad H. Hajiesmaili, Zachary J. Lee, Noel Crespi, and Enrique Mallada

Abstract—This paper tackles online scheduling of electric vehicles (EVs) in an adaptive charging network (ACN) with local and global peak constraints. Given the aggregate charging demand of the EVs and the peak constraints of the ACN, it might be infeasible to fully charge all the EVs according to their charging demand. Two alternatives in such resource-limited scenarios are to maximize the social welfare by partially charging the EVs (fractional model) or selecting a subset of EVs and fully charge them (integral model). The critical challenge is the need for online solution design since in practical scenarios the scheduler has no information about future arrivals of EVs in a time-coupled underlying problem. For the fractional model, we devise both offline and online algorithms. We prove that the offline algorithm is optimal. Using competitive ratio as the performance measure, we prove the online algorithm achieves a competitive ratio of 2. The integral model, however, is more challenging since the underlying problem is strongly NP-hard due to 0/1 selection criteria of EVs. Hence, efficient solution design is challenging even in offline setting. We devise a low-complexity primal-dual scheduling algorithm that achieves a bounded approximation ratio. Built upon the offline approximate algorithm, we propose an online algorithm and analyze its competitive ratio in special cases. Extensive trace-driven experimental results show that the performance of the proposed online algorithms is close to the offline optimum, and outperform the existing solutions.

Index Terms—Electric Vehicle, Online Scheduling, Approximation Algorithm, Competitive Analysis

I. INTRODUCTION

In order to promote quick adoption of renewable energy sources, electrification of vehicles is a trend that has been globally advocated in recent years. According to a Bloomberg report, EVs will account for more than half of the new car sales by 2040 [1]. Consequently, it is expected that demand from EV charging will constitute a considerable portion of total demand. Currently, transportation consumes 29% of total energy in the US, while electricity production consumes 40% [2].

EV charging demand is a typical example of a deferrable load, and there is often considerable flexibility in the charging schedule. This property makes the problem of EV charging scheduling important and there is an extensive research along this direction in the literature (see the recent survey [3], and references therein). In most of the existing works, the EV charging demand is treated as a constraint to the problem in a low-load regime [4]. Motivated by rapid proliferation of EVs, this work tackles the EV scheduling in high-load regime, where given the aggregate charging demand of the EVs and the peak constraints of the charging network, it is not feasible to fully charge all the EVs according to their charging demand.

More specifically, this paper studies resource-constrained EV charging scheduling in an adaptive charging network (ACN) governed by a single operator in a campus-scale location such as a university, a headquarters, etc. [5]. A notable example is the Caltech ACN [6] where individual charging ports are organized into several charging stations (CSs) which are dispersed in a charging network with the capability of adaptive charging of the EVs. The problem is different from EV charging scheduling with capacity constraint in single station scenarios [2], [4], [7]–[12] (we refer to Section II for detailed discussions on related work), because of the essential need to respect the aggregate peak demand of the ACN. Note that, the ACN operator might limit the total power drawn from EVs to control costs [13], [14], reserve the capacity for other loads, and/or participate in demand-response programs.

We consider a scenario with multiple EVs where each EV has different charging profile in terms of availability, charging demand, charging rate limit, and valuation of getting charged (for details see Section III-A2). We formulate an online EV charging scheduling problem with the goal of selecting and scheduling a subset of EVs such that: (1) the charging demand of the selected EVs are (fully or partially) satisfied; (2) the charging rate limit of EV batteries are respected, (3) the global peak constraint of the ACN is satisfied [6]; (4) the local peak constraint of each CS is respected; and finally, (5) the total revenue obtained by the valuation of the EVs is maximized.

There are two main challenges in the design and implementation of scheduling algorithms for EVs satisfying the goals mentioned above. Firstly, the problem calls for online scheduling design. In practice, EVs arrive to the CS in online fashion and the scheduler has no information about the arrival and demand of the future EVs. Secondly, the underlying optimization problem in integral model is strongly NP-hard even in offline case (see Section V). This is because the problem is a mixed integer linear problem and a “time-expanded” extension of knapsack problem which is known as a classic NP-hard problem. In this paper, we tackle the challenge of online design by following competitive algorithm design [15] and the challenge of NP-hardness by pursuing approximation algorithm design [16] and make the following contributions:

▷ We first consider a fractional model (where EVs can be charged partially and the revenue is proportional accord-
ingly) and design an optimal offline scheduling algorithm. We then develop an efficient online algorithm in which no exact or stochastic information about the future EV arrivals is given. Despite its simplicity, the algorithm is proved to be \(2\)-competitive with optimal offline solution, i.e., the revenue of the proposed online algorithm is at least \(1/2\) of the offline optimum, regardless of input sequence. Even though there are competitive algorithms in the literature for similar problems, to the best of our knowledge, our algorithm is the first \(2\)-competitive algorithm which considers the charging rate limits.

\(\triangleright\) We next study the more challenging scenario of the integral model, where EVs must receive all their demand to make revenue. We first propose a polynomial-time primal-dual offline approximate algorithm. We analyze the approximation ratio of the algorithm and by strengthening the linear relaxed version of the mixed integer problem [17], we obtain an approximation ratio of \(\alpha = 1 + \sum_{j=1}^{m} \frac{p_j}{q_j} \cdot \frac{1}{s_1}\), where \(p_j\) is local peak constraint in station \(j\), \(q_j\) is the maximum charging rate of the EVs in station \(j\) and \(s\) is a slackness parameter. We highlight that when \(p_j \gg q_j\) and \(s\) is large enough, then \(\alpha \approx m + 1\), where \(m\) is the number of stations in the ACN. Built on the basis of the offline algorithm, we devise an online algorithm, and discuss its competitive ratio in special cases.

\(\triangleright\) We conduct a set of simulations to evaluate the performance of our proposed algorithms. The results of online algorithms for both integral and fractional settings are close to the optimum (within 90% and 94% for integral and fractional models in a representative scenario). In addition, our algorithm outperforms the existing scheduling algorithm in Caltech ACN [6] by 35% for integral revenue model.

This paper represents follow-up work to our previous study [40], where we address a simplified version of the problems studied in this paper in fractional revenue model. The competitive analysis in [40] is done under the assumption that all EVs have the same maximum charging rate. Also, [40] does not study the integral revenue model. In this paper, we extend the results and propose a \(2\)-competitive online algorithm with no assumption on the input parameters. Also, the current study investigates both fractional and integral revenue models while considering global peak constraint in the formulation and addressing multiple charging station scenario. Last, we consider more realistic trace-driven experiments in this paper and compare the results to an existing real-world scheduling algorithm.

II. RELATED WORK

There is an extensive work in the general topic of EV demand management and scheduling [3] by considering different scenarios such as optimal operations with EV coordination and congestion management [18], [46], [47], EV scheduling with incorporation of renewable energy, and energy storage systems [19], [20], and pricing and bidding [45], [48]. In this section, we focus on the literature related to peak-constrained EV charging scheduling in single and multiple stations.

1) Peak-Constrained EV Charging Scheduling: There is an extensive literature on EV charging scheduling problem focusing on single CS [4], [7], while the local and global peak constraints are omitted or only the local peak is considered. As we discuss in Section III-B, the global optimal solution cannot be obtained by separately solving the single station problems. Hence, those solutions cannot be directly applied to the multiple CS scenario with global peak constraints.

Studies in [21]–[26] tackled charging scheduling problem in multiple CSs. The authors in [21] studied a global cost minimization EV charging scheduling problem. However, there is no limit on the maximum peak demand that the system can tolerate. Consequently, the peak can be arbitrary high and beyond the physical limit of the transformers in the charging station [6] leading to huge electricity bills [14]. Besides, the charger devices installed in the CSs have limitation on the maximum power that they can transfer in each time unit [27]. [24] considered a multi-microgrid system with global peak constraint where each microgrid has a CS and the goal is to manage electricity exchange between microgrids and main grid to minimize the operating costs. The authors assume that required information about individual EVs is available by forecast which may not represent a real scenario. A similar assumption is made in [25], [26] where the objective is to maximize total utility of EVs and aggregators in a distribution network. The authors in [22] use a similar model as in this paper, where both local and global peak constraints are considered. However, the authors solve the single-slot problem, which fails to provide a general solution taking into account EVs’ arrival and departure times as considered in our study. Finally, as an alternative approach to control the peak, some studies directly target minimizing the peak [2], [28]. In [2], an online algorithm is developed for EV charging to minimize the peak and [28] proposed a valley filling method by leveraging V2G in peak hours. Although the peak is minimized in above works, it cannot guarantee that the minimized peak is tolerable by the underlying charging network.

2) Scheduling Under Demand Uncertainty: A main challenge in EV scheduling problem is to cope with demand uncertainty. Many studies including [2], [4], [8]–[12] addressed online scheduling problem with different objectives. [8], [11] studied the problem of maximizing social welfare considering the benefit for both users and service provider. [4] and [9] developed algorithms to minimize the charging price for the CS, where the proposed algorithm in [4] is 2.39-competitive. In [12], an online 2-competitive algorithm is proposed for a single CS, assuming that EVs are capable of being discharged in a negligible amount of time. The assumption, however, is not realistic for EVs.

Our problem in this paper is unique from above works in many respects. First, we study the problem in an ACN where several CSs exist, while none of the above studies solve the problem under this setting. Second, the previous algorithms do not work for both integral and fractional charging models. In addition, [9], [10] put no limit on the charging rate of EVs which makes their solution impractical in real scenarios. Also, [2], [4], [8]–[10] do not consider the peak limit of the CS. Note that some studies [29] tackle the scenarios of satisfying the large demands (possibly EV demands) by participating in electricity market. This study assumes that the electricity price is given and fixed, and leaves the case of real-time pricing as the future direction. In addition, there are studies that
incorporate prediction in online EV charging scheduling [30], [31]. The prediction-based approaches achieve satisfactory performance for the scenarios that follow prediction. Deviation from prediction models, however, degrades their performance. Our approach, on the other hand, has no assumptions on modeling/prediction, and in this way, is robust against any uncertainty in the instances to the problem. Finally, in [32], we considered a simplified EV charging scheduling in fractional model without global peak constraints and devised heuristic algorithms (without competitive and approximation analysis) with on-arrival commitment for EVs to notify the amount that they can receive by their departure.

3) Worst-case Analysis in Similar Scheduling Problems: Similar underlying scheduling problems with slightly different settings have been studied in the literature in both offline and online settings. In offline setting, the problem is more interesting under integral revenue model where the problem becomes a combinatorial optimization problem and approximation algorithms have been used to find effective solutions. The performance of an approximation algorithm is determined by its approximation ratio for offline algorithms. On the other hand, competitive online algorithms are used in the online setting and competitive ratio is the performance metric which compares the algorithm’s result to the offline optimal solution.

Offline integral model: Under integral revenue model, [38] proposes an offline algorithm for scheduling of batch jobs in cloud computing which is similar to EV scheduling problem. It is assumed that all jobs are available to be processed at time 0. The authors propose a \( \frac{C}{s-1} - 1 \)-approximation algorithm where \( C \) is the cloud capacity and \( s \) is the “slackness parameter” (See section III-A1 for the definition). Similarly, [39] tackles the offline integral problem and proposes a convex relaxation method to find a near-optimal solution. No theoretical bound is provided for the algorithm and the performance is examined by simulation results. In this paper, we propose ICs for the EV scheduling problem with slightly different settings in the constraint sets, and tackle the problem in online setting in which the full knowledge of the online parameters is not available a priori, and provide approximation ratio under specific scenarios.

Offline fractional model: In case of fractional revenue model (offline case), the scheduling problem is more straightforward to tackle since the underlying problem is linear. In this paper, we propose FCS as an optimal solution for offline fractional case with low time complexity.

Online fractional model: In the online scenario, the scheduling problem is fundamentally more challenging. In our recent study [40], we proposed two online algorithms referred to as WFAIR and WRAND for EV scheduling problem. The proposed algorithms provide a competitive ratio of \( 2 - 1/U \) where \( U \) is “scarcity level”. However, the result only holds when all EVs have the same maximum charging rate. Also, [40] does not study the integral revenue model. In this paper, we propose FOCS as a 2-competitive online algorithm with no assumption on the input parameters. Moreover, the current study investigates both fractional and integral revenue model while considering total peak constraint in the formulation and addressing multiple charging station scenario. In another work [34], two simple and natural online algorithms called FIRSTFit and ENDFit were developed and are proved to be 2-competitive. FOCS is extension of these algorithms by taking into account the maximum charging rate of the EVs and multiple charging station scenario (see “Remarks” at Section IV). As recent studies, [41], [43] provide online algorithms for EV scheduling problem, however, they do not take into account global peak constraint in the underlying problem.

Online integral model: Another direction is to tackle online scenario under integral revenue model. In this setting, the scheduling problem becomes fundamentally more challenging. We note that the integral scheduling problem is strongly NP-Hard even in the offline case [42]. Due to combined online and combinatorial challenges, there are very limited studies in this category. [44] provides an online algorithm without considering the maximum processing rate of the jobs where its competitive ratio can be arbitrary bad depending on “slackness” parameter. Our work in Section V-B extends this study for multiple station scenario and considering maximum charging rate for the EVs.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We consider a time-slotted system model in which the time horizon is divided to \( T \) equal length slots \( t = \{1, 2, \ldots, T\} \) (e.g., \( T = 24 \) with time slots of 1 hour length).

1) Charging Network: Our charging network model is inspired by the Caltech ACN [6] as illustrated in Fig. 1a. In the Caltech EV charging network (located in a parking garage), electricity is distributed through a two-level transformer architecture from a main switch panel to multiple EV switch panels (2 panels in the current Caltech ACN). Each EV switch panel then is connected to several chargers (≈25 chargers per panel in the Caltech ACN). The total power drawn from the main switch panel by the charging network has a power limit of \( p^{\text{total}} \), that is determined by the facility operator to control the costs, reserve the capacity for other loads, and/or participate in demand-response programs. In other words, \( p^{\text{total}} \), which we refer to it as the global peak, hereafter, limits the maximum aggregate EV charging load at each time slot.

We assume that there are \( m \) EV switch panels that represent \( m \) CSs. In addition to the global peak constraint, each CS \( j \) has a capacity constraint on its total power drawn, indicated by \( p_j \), and referred to it as the local peak constraint. The value of \( p_j \) is determined by the output power limit of the transformers installed between the main switch panel and EV switch panels and could be different for each EV switch panel. It is often observed that the charging demand of different CSs (EV switch panels in Fig. 1a) are well below the local peak constraints. To increase the flexibility due to heterogeneous charging demand of CSs, in the Caltech ACN, the global peak constraint of the main switch can be over-provisioned, i.e., \( p^{\text{total}} \) is less than the aggregate local peaks, \( \sum_j p_j \geq p^{\text{total}} \). While this increases flexibility, it also couples the problem of EV charging scheduling across different CSs. Our solutions in this paper will be centralized ones which can be obtained through communication between the CSs and a central server as illustrated in Fig. 1b.
3) Revenue Models: We consider two revenue models: (i) Fractional revenue model: In this model, the fractional charging is allowed, i.e., the revenue from each EV is proportional to the fraction of the demand that is fulfilled [11] (see Eq. (2)). We tackle this model in Section IV. (ii) Integral revenue model: In this model, EV $i$ pays $v_i$ if it is fully charged and zero otherwise, i.e., there is no partial revenue for partial charging. This is the model that is considered in [8], [9]. We tackle this model in Section V.

B. Problem Formulation

We formulate an optimization problem to schedule the charging of the EVs with the objective of maximizing total revenue obtained from charged EVs while respecting local and global peak constraints. Note that each revenue model makes the underlying optimization problem fundamentally different. More specifically, the fractional charging model is a linear problem. Integral revenue model, however, turns the underlying charging scheduling problem to a Mixed Integer Linear Program (MILP). The integer nature originates from the 0/1-selection decision on EVs. We formulate a Scheduling Problem for Adaptive charging Network (SPAN) under fractional revenue model as follows:

$$\text{SPAN : max} \quad \sum_{i=1}^{n} \frac{v_i}{D_i} \sum_{t=a_i}^{d_i} y_i^t$$

s.t.  \begin{align}
\sum_{t=a_i}^{d_i} y_i^t & \leq D_i, \quad \forall i, \quad (1a) \\
\sum_{i=1}^{n} y_i^t & \leq p_{t}^{\text{total}} \quad \forall t, \quad (1b) \\
\sum_{i:h(i)=j} y_i^t & \leq p_j \quad \forall t, j, \quad (1c)
\end{align}

$$y_i^t \leq \frac{k_i}{D_i} \sum_{i'=a_i}^{d_i} y_i^{t'}, \quad \forall i, t, \quad (1d)$$

vars.  \quad y_i^t \geq 0, \quad \forall i, t,$

where $y_i^t$ is the amount that EV $i$ is charged at slot $t$. Constraint (1a) ensures that the aggregate amount received by EV $i$ is at most the demand $D_i$. The global and local peak constraints are represented by constraints (1b) and (1c), respectively.

The constraint (1d) enforces the maximum charging rate of EVs. The straightforward way to express this constraint is to simply state that at each time slot $t$, the charging rate of EV $i$ should be less than or equal to its maximum charging rate, i.e., $y_i^t \leq k_i, \forall i, t$. However, for the sake of effective algorithm design for integral model and reducing the integrality gap of the relaxed linear problem, this constraint is strengthened in the form of Eq. (1d). Note that in case that the aggregated charging of EV $i$ during its availability window is equal to its demand, i.e., $\sum_{i'=a_i}^{d_i} y_i^{t'} = D_i$, Eq. (1d) reduces to the simple form of $y_i^t \leq k_i$. This is a natural way in approximation algorithm design to improve the performance of the algorithms under linear-relaxation based design [17].

The SPAN is an extension of formulated problem in [33] where a job scheduling problem in cloud applications is studied. It turns out that the resource allocation problem in cloud systems and the EV charging scheduling problem in a single station share similar structure. Indeed, each charging profile in our scheduling problem can be seen as a job in cloud system with a deadline, value, and CPU demand. The
TABLE II: Proposed algorithms and their properties.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Revenue model</th>
<th>Type</th>
<th>Optimality</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>fCS</td>
<td>fractional</td>
<td>offline</td>
<td>Optimal</td>
<td>(O(n^2T + nT^2))</td>
</tr>
<tr>
<td>oCS</td>
<td>integral</td>
<td>offline</td>
<td>((1 + \sum_{j=1}^{m} \frac{p_j}{y_q - y_{q-1}}))-approximate</td>
<td>(O(nT \log T + n^2T))</td>
</tr>
<tr>
<td>fOCs</td>
<td>fractional</td>
<td>online</td>
<td>(2)-competitive</td>
<td>(O(n^T))</td>
</tr>
<tr>
<td>iOCs</td>
<td>integral</td>
<td>online</td>
<td>(6(1 + \frac{D}{p_{max}} T + \epsilon))-competitive, (m = 1)</td>
<td>(O(n^2T))</td>
</tr>
</tbody>
</table>

SPAN, however, comes with an additional constraint (1b), which makes it different from the problem in [33], such that the existing solution will not work in the new setting. More importantly, this paper studies both offline and online solutions for the problem in both fractional and integral settings, while [33] tackles only the offline integral model.

IV. FRACTIONAL REVENUE MODEL

In the fractional model, the revenue of the CS from EV \(i\) is directly proportional to the amount of resource that the EV is received, i.e.,

\[
v_i = \frac{y_i}{D_i} v_i,
\]

where \(v_i\) is the gain, if the entire demand \(D_i\) is fulfilled and \(v_i\) is the fractional gain.

We propose a simple algorithm, called fCS, with low computational complexity of \(O(n^2T + nT^2)\) for the fractional model that finds the optimal solution in offline setting. Note that even though linear programs can be solved in polynomial time in general, the complexity of our proposed algorithm is much lower than the general linear program algorithms. Moreover, our proposed offline algorithm applies a valley-filling strategy to reduce the peak. A summary of proposed algorithms with their complexity and assumptions is given in Table II.

A. OPTIMAL OFFLINE DESIGN

In this section, we propose an optimal algorithm for offline fractional model. As a natural solution, one may think of a greedy algorithm as follows: at each time slot, check the EVs with highest unit-value and process others if only the remaining resource cannot be allocated to the selected EV(s) (e.g., due to maximum charging rate constraint or fulfillment of the EVs’ demand). This approach is a popular, yet straightforward method in the scheduling. However, it turns out that this approach cannot provide an optimal solution to the problem even in fractional revenue model. In fact, this solution is only 2-approximation (i.e., the profit obtained by the algorithm would be 1/2 of the optimum). As an intuition of the proof think of two EVs in a single charging station with power capacity \(p\) and \(\frac{D_1}{T_1} - \frac{D_2}{T_2} = \epsilon\) where \(\epsilon > 0\) is an arbitrary small number, \(a_1 = a_2 = 1, d_1 = 2, d_2 = 1\) and \(D_1 = D_2 = k_1 = k_2 = p\). Then, applying the simple greedy, EV 1 will be selected at time slot 1 and EV 2 will have no chance to get charged due to its deadline constraint while EV 1 could wait until time slot 2 and still receive all its demand.

We refer the proposed algorithm in this section as the fCS and summarize it as Algorithm 1. The fCS works in two phases. In the first phase (Section IV-A1), the algorithm decides on the amount of resource to be allocated to each EV within its availability window and reserves resources accordingly. In this phase, the details of allocation is not known. The actual resource allocation is done in the second phase (Section IV-A2) by setting variables \(v_i\).

Before discussing the details of the algorithm, we give a definition and introduce some notations to facilitate our algorithm design.

**Definition 1.** Time interval \([\delta, \delta']\) is a “super interval” for interval \([t, t']\) if \(1 \leq \delta \leq t AND \ t' \leq \delta' \leq T\). Moreover, \(I_{t, t'}\) is the set of all super intervals of interval \(I_{t, t'}\) i.e., \(I_{t, t'} = \{[\delta, \delta']: 1 \leq \delta \leq t AND \ t' \leq \delta' \leq T\}\).

The number of super intervals of an interval is at most \(T^2\) and at least one (for interval \([1, T]\)).

Let \(R_i\) be the amount of resource that is reserved for EV \(i\) by the fCS and \(I_{t, t'}\) as time interval \([t, t']\). Then, assuming that charging demands are sorted in non-increasing order of their unit values, \(A_j^1(t, t')\) is the aggregate residual resource in interval \(I_{t, t'}\) at station \(j\) assuming that the reservation for EVs 1 to \(i\) is accomplished. We now explain in detail each phase of the algorithm.

1) Phase I-Reservation: In Line 1, the EVs are sorted in a non-increasing order of their unit values. In Line 5, the fCS processes demand of EV \(i\), picked from the ordered list, and sets \(R_i\) as the amount to be reserved for EV \(i\) which will be allocated in Phase II. In Line 7, the residual resource of all intervals in set \(I_{a_i, d_i}\) decreases by \(R_i\) and EV \(i\) is added to the set of selected EVs.

**Lemma 1.** Provided that for EV \(i\) we have

\[
R_i \leq \min_{t, t'} \left\{ D_i, \min_{t, t'} A_j^{i-1}(t, t'), \forall t, t' : I_{t, t'} \in I_{a_i, d_i} \right\}
\]

with a dummy \(A_j^0(t, t')\) defined as \(A_j^0(t, t') = (t' - t + 1) \times \min\{p_{total}, p_j\}\), then there is a feasible allocation to allocate \(R_i\) to EV \(i\) in its availability window \([a_i, d_i]\).

In Lemma 1, \(A_j^0(t, t')\) indicates the available resource when no charging request is processed in \(I_{t, t'}\). In Eq. (3), the second term, i.e., \(\min_{t, t'} A_j^{i-1}(t, t')\), indicates the minimum remaining resource in all super intervals of interval \(I_{a_i, d_i}\). For \(i \geq 1\), \(A_j^i(a_i, d_i)\) is defined as follows:

\[
A_j^i(a_i, d_i) = \begin{cases} A_j^{i-1}(a_i, d_i) - R_i & j = h(i), \\ A_j^{i-1}(a_i, d_i) & j \neq h(i). \end{cases}
\]

The optimal value of \(R_i, i = 1, \ldots, n\) is set according to the following lemma:

**Lemma 2.** Given \(n\) EVs sorted in a non-increasing order of the unit values, \(v_1/D_1 \geq v_2/D_2 \geq \cdots \geq v_n/D_n\, and the
Algorithm 1: FCS

Input: $n$ EVs with their profile, local and global peak constraints $p_j, j = 1, \ldots, m$ and $p_{\text{total}}$

Output: Optimal scheduling under fractional model

1. Sort charging requests in non-increasing order of their unit values, i.e., \( \frac{v_i}{D_i^t} \geq \frac{v_j}{D_j^t} \geq \cdots \geq \frac{v_n}{D_n^t} \)
2. \( \mathcal{L} \leftarrow \emptyset \)
3. //Phase I
4. for \( i = 1, \ldots, n \) do
5. \( R_i \leftarrow \min \{ D_i, \min_{t,t'} A_{h(i)}^{i-1}(t,t'), \forall t, t': I_{a_i,d_i} \} \)
6. if \( R_i > 0 \) then
7. \( A_{h(i)}^{i-1}(t,t') \leftarrow A_{h(i)}^{i-1}(t,t') - R_i, \forall t, t': [t,t'] \in I_{a_i,d_i} \)
8. \( \mathcal{L} \leftarrow \mathcal{L} \cup i \)
9. //Phase II
10. Sort EVs in \( \mathcal{L} \) in increasing order of their charging flexibility i.e., \( \frac{(d_i - a_i + 1)k_i}{d_i^t}, i \in \mathcal{L} \).
11. for \( i = 1, \ldots, |\mathcal{L}| \) do
12. Pick EV \( i \) from the sorted list \( \mathcal{L} \).
13. feasible \( \leftarrow \{ \sum_{t=0}^{d_i^t} \min\{k_i, A_{h(i)}^{i-1}(t)\} \} - R_i \)
14. if feasible \( < 0 \) then
15. Re-allocate previously allocated EVs such that \( \text{feasible} \geq 0 \)
16. Arbitrarily allocate \( R_i \) to EV \( i \) in its availability window

The following theorem characterizes the complexity of FCS.

Theorem 2. The time complexity of FCS algorithm is \( O(n^2T + nT^2) \) where \( n \) is the number of EVs and \( T \) is number of time slots.

B. Online Scenario

In this section, we devise an algorithm for the scenario that EVs arrive in online fashion. The scheduling decisions at each time slot are made given the information of available EVs and neither exact values nor stochastic modeling of future arrivals is available. Our goal is to obtain a competitive ratio for the online algorithm. A scheduling algorithm \( A \) is \( c \)-competitive for \( c \geq 1 \) if the revenue obtained by the optimal offline algorithm is at most \( c \) times the algorithm \( A \)'s revenue for any input sequence [15].

The proposed online algorithm for the fractional model, referred to as FOCS, is listed as Algorithm 2. The FOCS is a simple yet efficient algorithm that always selects EVs with the highest unit value to allocate as follows. First, the algorithm sorts the available EVs at each time slot \( t \) based on their unit value (Line 2). In this step, the algorithm breaks ties based on EVs’ deadline i.e., if two EVs have the same unit value, then the one with earliest deadline comes first in the sorted list. Next, the FOCS selects one EV at a time from the sorted list to allocate with maximum charging rate considering the EV’s remaining demand, maximum charging rate, and peak constraints (Lines 3-4). The allocation is continued until all resources are allocated or there is no more EV that could be allocated. The time complexity of the FOCS is \( O(n^2T) \), determined by cost of its “for” loop multiplied by total number of times that the algorithm needs to be run.

Algorithm 2: FOCS: \( \forall t \in \{1, 2, \ldots, T\} \)

1. \( M^t \leftarrow \) The set of available EVs that did not receive their entire demand by \( t \)
2. Sort EVs in set \( M^t \) indexed by \( i = 1, \ldots, |M^t| \):
3. for \( i = 1, \ldots, |M^t| \) do
4. \( y_i^t \leftarrow \min\{k_i, D_i - \sum_{\tau=t-a_i}^{t} y_{\tau}^t, p_h(i) - \sum_{\tau'=a_i}^{t} p_h(\tau') \}, y_{\tau}^t \) for \( \tau = a_i, \ldots, t \) starting from time slot \( t = a_i \) until \( R_i \) units is allocated.

Despite the simplicity of FOCS which makes it easy to implement, its performance is sound and within a constant factor of the offline optimum. We now proceed to analyze the performance of the FOCS by first giving some preliminaries.

Fix an optimal scheduling and let \( S_{\text{FOCS},t} \) and \( S_{\text{OPT},t} \) be the sets of EVs selected by the FOCS and optimal solution at time slot \( t \), respectively. Let \( y_i^t \) and \( z_i^t \) be the charging rate of EV \( i \) set by FOCS and OPT, respectively. We define \( \Delta_i^t \) as follows:

\[
\Delta_i^t = \begin{cases} 
\min\{z_i^t - y_i^t, r_{i,t} \} & \text{if } r_{i,t} > y_i^t, \\
0 & \text{otherwise},
\end{cases}
\]

where \( r_{i,t} \) is the remaining demand of EV \( i \) by the end of time slot \( t \). \( \Delta_i^t > 0 \) indicates that the optimal algorithm allocated \( \Delta_i^t \) units more resources to EV \( i \) than the FOCS by time slot.
that could be feasibly allocated by FOCS to the EV $i$. If for any EV $i \in S_{OPT,t}$ and time slot $t \in T$ we have $y_i^t = z_i^t$, i.e., $\Delta_i^t = 0$, then the FOCS is obviously optimal because it gains whatever the optimal solution gains. We define loss of the FOCS imposed by EV $i$ as follows:

$$l_i,t = \Delta_i^t \frac{v_i}{D_i^t}.$$  

(5)

Note that the loss always takes a non-negative value as $\Delta_i^t \geq 0$. When FOCS sets charging rate of an EV less than its rate in the optimal solution, it gains $\Delta_i^t \frac{v_i}{D_i^t} < \Delta_i$ less than the optimal solution from that EV. An upper bound for the distance between the optimal objective value (denote by OPT) and the revenue of the FOCS (denote by ALG) is summation of the losses over all time slots and all EVs, i.e.,

$$OPT - ALG \leq \sum_{t=1}^{T} \sum_{i \in S_{OPT,t}} l_i,t.$$  

(6)

In addition to the amount of loss, the gain of the algorithm from charging alternative EVs should also be taken into account in comparison between OPT and ALG.

Let $i \in S_{OPT,t}$, $i \notin S_{ALG,t}$ and $g_i,t$ be the gain that the FOCS obtains from charging another EV instead of $i$ at time slot $t$. We are going to show that in the FOCS, for each EV $i$ with $l_i,t > 0$ there must be another EV (denote by $i'$) where $y_i'^t \geq \Delta_i'^t$ and $\frac{D_i'}{D_i} \geq \frac{D_i'}{D_i}$ which means in resource allocation phase for EV $i$, the FOCS allocated the difference $\Delta_i' \geq 0$ to another EV with the same or higher unit value. This can be proved by considering the fact that (i) the selected EVs have higher unit values than the unselected EVs, and (ii) the charging rate of the selected EVs are set to the maximum feasible value. Moreover, the FOCS is “work-conserving” i.e., it does not let any resource remain unused if there are some EVs that can use it.

Let $g_i,t$ denote the gain that FOCS obtains from allocating the same amount of resource that optimal algorithm allocated to EV $i$ (with size $z_i^t$) to another EV(s). If $\Delta_i^t = 0$, the loss is zero. If $\Delta_i^t > 0$, then by the charging strategy that the FOCS uses we can conclude that (i) $i$ is not finished by the FOCS, and (ii) no more resources from station $p_h(i)$ can be feasibly allocated, otherwise, the FOCS could allocate more resources to EV $i$. Therefore, the $\Delta_i^t$ units of the resource are allocated to one or multiple other EVs (denote them by set $J_i^t$) by the FOCS. Moreover, it must hold that all the EVs in set $J_i^t$ have a unit value equal to or higher than $v_i/D_i$ which yields $g_i,t \geq l_i,t$, otherwise, the FOCS should not prefer the EVs in $J_i^t$ to $i$. Since the result holds for any arbitrarily EV $i$, we can get the following:

$$\sum_{i=1}^{n} \sum_{t=1}^{T} l_i,t \leq \sum_{i=1}^{n} \sum_{t=1}^{T} g_i,t.$$  

(7)

Moreover, the total gain of FOCS, i.e., ALG, is equal to sum of its gains from each single EV:

$$ALG = \sum_{i=1}^{n} \sum_{t=1}^{T} g_i,t.$$  

(8)

With the above discussion and using Eqs. (6), (7) and (8) we are able to derive a competitive ratio of 2 for the FOCS.

**Theorem 3.** The FOCS is 2-competitive.

**Remarks:** When there is only one CS and EVs have no limit on their charging rate, the FOCS is identical to the FIRSTFIT algorithm [34] which is known to be 2-competitive for classic job scheduling problem. However, the charging rate limitation is crucial for EV charging scenario. Moreover, [34] uses a “charging argument” to prove the competitive ratio of the proposed algorithm which cannot be directly applied to our problem. Thus, the FOCS extends the FIRSTFIT and makes it practical for the EV charging scenario. Moreover, the proof technique used for the competitive analysis of the FOCS is fundamentally different from the one used in [34].

**V. INTEGRAL REVENUE MODEL**

The MILP form of the SPAN in integral model is a generalized form of the 0/1-knapsack problem which is a well-known NP-hard problem. To give an intuition, consider the scheduling problem in a single time slot (i.e., $T = 1$). Then, allocating power resources to the EVs is equivalent to allocating the capacity of knapsack to the items. In Section V-A, we propose a fast polynomial time offline approximation algorithm for the integral problem. In Section V-B, we extend the result and propose an online algorithm for the integral model.

**A. Offline Scenario**

We design our offline scheduling algorithm under integral model referred to as iCS, to solve the SPAN approximately. Since the performance analysis of the proposed algorithm relies on a dual fitting method and utilizes weak duality property, we first need to construct the dual problem of SPAN. Toward this, we introduce variables $\alpha, \beta, \gamma$ and $\pi$. Generally, in primal-dual approximation algorithm design, each constraint (resp. variable) in primal (resp. dual) problem is associated with a variable (resp. constraint) in dual (resp. primal) problem. In our case, constraints (1a) (1b), (1c) and (1d) in the SPAN are respectively associated with dual variables $\alpha, \beta, \gamma$ and $\pi$ (for more details see [16]). The dual problem is formulated as follows:

$$\min \sum_{i=1}^{n} D_i \alpha_i + \sum_{j=1}^{m} \sum_{t=1}^{T} p_j \beta(t) + \sum_{t=1}^{T} p^{total} \gamma(t)$$

s.t. $\alpha_i + \beta(t) + \gamma_i + \pi(t) - \frac{k_i}{D_i} \sum_{t'=a_i}^{d_i} \pi(t') \geq \frac{v_i}{D_i}$,

$$\forall i, t \in [a_i, d_i],$$

(9a)

vars. $\alpha_i, \beta_i, \gamma_i, \pi_i(t) \geq 0, \quad \forall i, t.$

1) **Explanation of the Main Algorithm:** Our algorithm design is inspired by the basic algorithm proposed in [33]. The algorithm in [33], however, works for a single CS where arrival time of all EVs are the same and there is no global peak constraint. The iCS algorithm (listed as Algorithm 3) works in two phases. In the first phase it sorts the charging requests based on their unit values in a non-increasing order. Then, it selects most valuable demand. If the remaining resource is enough to cover the entire demand of the EV, it is admitted to receive the demand (Lines 6-7).

**Scheduling of the Selected EV:** If the feasibility check passed (Line 6 in Algorithm 3), iCS calls sub-procedure
SMARTALLOCATE to allocate required resources in interval $[a_i, d_i]$. Then, $\alpha_i$ is set to $v_i/D_i$ in order to cover dual constraint in Eq. (9a) (Lines 7-8).

In SMARTALLOCATE, let us define $W(t, h(i)) = \sum_{i', h'(i')=h(i)} y_{i'}^{t}$ as total workload at time slot $t$ in CS $h(i)$ and $W(t, h(i))$ as total available load to allocate at time slot $t$ for CS $h(i)$. We always have $W(t, h(i)) + W(t, h(i)) = p(h(i), \forall t, i)$. For scheduling, SMARTALLOCATE applies two main policies: 1) valley-filling, and 2) right-to-left allocation. With valley-filling, the slots with more available resources are preferred which helps to reduce the peak of the system. Right-to-left allocation is used when two or more time slots are equal in terms of their remaining resources. A ranking based approach is used to apply the aforementioned policies. To charge EV $i$, SMARTALLOCATE ranks time slots in interval $[a_i, d_i]$. Then, charging is done by allocating resources from the higher ranked time slot to lowest one. The rank of a time slot $t$ is calculated based on remaining resources in the time slot (valley-filling) and value of $t$ (right-to-left allocation).

Algorithm 3: iCS

1. initialize: $y \leftarrow 0, \alpha \leftarrow 0, \beta \leftarrow 0, \gamma \leftarrow 0, \pi \leftarrow 0$
2. Sort charging requests in non-decreasing order of their unit values: $v_1/D_1 \geq v_2/D_2 \geq \cdots \geq v_n/D_n$
3. for $(i=1 \ldots n)$ do
   4. for $t = a_i \ldots d_i$ do
      5. $\sigma_t \leftarrow \min \{ \rho_{h(i)} - \sum_{i'=1, h'(i')=h(i)} y_{i'}^{t}, \pi \}$
      6. if $D_i \leq d_{i}$ then SMARTALLOCATE($i, D_i$)
      7. $\alpha_i \leftarrow v_i/D_i$
      8. else if $(\beta(d_i) = 0)$ then BETCOVER(i)
      9. end
   10. end
11. end
12. for $(i=1 \ldots n)$ do
13. if EV $i$ is not selected then RECONSIDER(i);
14. end

Dual Feasibility of the Non-selected EVs: If remaining power resource is not enough to fully charge EV $i$, i.e., $D_i > \sum_{t=a_i}^{d_i} \sigma_t$, the EV cannot be selected. However, we still need to satisfy constraint (9a) in dual problem which is done by calling BETCOVER($i$). To cover the constraint (9a) for EV $i$, sum of dual variables for all $t \in [a_i, d_i]$ should be greater than or equal to $v_i/D_i$. BETCOVER($i$) sets $\beta(t)$ to $v_i/D_i$ for all time slots $t$ in interval $[t_{cov}, R(d_i)]$ (Lines 3-4 of Algorithm 5). Observe that $\beta(t') \geq v_i/D_i$, $t' < t_{cov}$ (with $t_{cov} > 1$) considering that the demands are sorted in a non-increasing order of the unit-values and $\beta(t')$ is already set to $v_i/D_i$ when processing the earlier charging demand of EV $i'$ which is not selected. Hence, $v_i/D_i \geq v_i/D_i$, thereby $\beta(t) \geq v_i/D_i$, and the dual constraint in (9a) is satisfied. Lines 1-4 of BETCOVER is enough to cover the dual feasible solution for the primal problem (with primal value $\Gamma$) and a feasible solution for the dual problem (with dual value $\Lambda$). Then, assuming that the primal problem is a maximization problem, to prove that the algorithm is $c$-approximation (for $c \geq 1$), the important part is to show that $\Lambda \leq c \Gamma$. Then, based on weak duality theorem we have $\Lambda \leq OPT$, and it is concluded that $\Gamma \geq \frac{1}{c} \times OPT$ where OPT is the optimal value. Based

Algorithm 4: SMARTALLOCATE($i, D_i$)

1. Rank time slots in interval $[a_i, d_i]$ such that for any $t_1$ and $t_2$:
   2. rank($t_1$) > rank($t_2$) iff $W(t_1, h(i)) > W(t_2, h(i))$ OR $W(t_2, h(i)) <= W(t_2, h(i)) \land t_1 > t_2$
3. while $\sum_{t'=a_i} y_{i'}^{t'} \neq D_i$ do
   4. Select time slot $t$ with highest rank which is not selected before
   5. Allocate $\min \{ k_i, W(t, h(i)), D_i - \sum_{t=a_i} y_{i'}^{t}, p_{total} - \sum_{t'} y_{i'}^{t'} \}$ to EV $i$ at $t$

Algorithm 5: BETCOVER($i$)

1. $t_{cov} \leftarrow \min \{ t : \beta(t) = 0 \}$
2. $R(d_i) = \max \{ t \geq d_i : \forall t' \in (d_i, t], \bar{W}(t') < q_{b(i)} \}$
3. for $(t = t_{cov} \ldots R(d_i))$
4. $\beta(t) \leftarrow v_i/D_i$
5. for $(i' = 1 \ldots R(d_i))$
6. if $(i' = 1 \ldots n)$
   7. if $y_{i'}^{t} > 0 \land \Phi_i(t) = 0$ then
    8. $\Phi_i(t) \leftarrow \frac{y_i^{t} - \sum_{t'=a_i} y_{i'}^{t'} - p_{total} - \sum_{t'} y_{i'}^{t'}}{D_i}$
   end
end

Algorithm 6: RECONSIDER($i$)

1. $L \leftarrow \emptyset$
2. $v_{inc} \leftarrow v_i$
3. $\sigma_t \leftarrow 0, t = a_i, \ldots, d_i$
4. for $(i = 1 \ldots n)$ do
5. if $EV i'$ is selected $\land (h(i') = h(i)) \land (v_{inc} - v_{i'} > 0$ then
   6. Add EV $i'$ to list $L$
   7. $v_{inc} \leftarrow v_{inc} - v_{i'}$
   8. for $(t = a_i, \ldots, d_i)$ do
      9. $\sigma_t \leftarrow \sigma_t + \min \{ k_i, y_{i'}^{t} \}$
   end
10. if $\sum_{t=a_i}^{d_i} \sigma_t \geq D_i$ then
   11. Remove EVs in list $L$ from charging schedule
   12. SMARTALLOCATE($i, D_i$)

Improving the Gain: In the second phase, the iCS tries to increase total value of selected EVs by calling RECONSIDER($i$) on every unselected EV $i$ (Lines 12-14). The RECONSIDER($i$) (listed as Algorithm 6) checks whether the total revenue can be increased by replacing some selected EVs with EV $i$ or not (Lines 4-9). If such EVs are found, the algorithm stops their charging and allocates EV $i$ using SMARTALLOCATE (Lines 10–12).

2) Analysis: In a primal-dual algorithm, the goal is to design an algorithm in a way that it produces a good solution for primal problem (with primal value $\Gamma$) and a feasible solution for the dual problem (with dual value $\Lambda$). Then, assuming that the primal problem is a maximization problem, to prove that the algorithm is $c$-approximation (for $c \geq 1$), the important part is to show that $\Lambda \leq c \Gamma$. Then, based on weak duality theorem we have $\Lambda \leq OPT$, and it is concluded that $\Gamma \geq \frac{1}{c} \times OPT$ where OPT is the optimal value. Based
on the above understanding, the following theorem scrutinizes the approximation ratio of the iCS assuming that arrival times are the same for all EVs.

**Theorem 4.** iCS algorithm is a \(1 + \sum_{j=1}^{m} \frac{p_i}{p_j - q_j}, \frac{s}{s-1}\)-approximation when EVs have same arrival time.

Note that in the case that the system is flexible enough, i.e., \(s \gg 1\), and the maximum charging rates of stations are much bigger than those of EVs, i.e., \(p_j \gg q_j, \forall j\), the approximation ratio approaches \(m+1\). And in the case that there is one single station, the approximation ratio is 2.

### B. Online Scenario

Due to the binary selection variable, the online solution design under integrals model is more challenging than the one for fractional model. We propose iCS that is built upon the offline iCS. In particular, the iCS calls the iCS at each time slot for the set of available EVs, however, any algorithm that is designed for offline integrals model can be used alternatively.

Hereinafter, A refers to the iCS or a similar algorithm.

**Algorithm 7:** iCS: \(\forall t \in \{1, 2, \ldots, T\}\)

1. Let \(A\) be an algorithm that solves the problem with \(a_1 = \cdots = a_n\)
2. \(R^t \leftarrow \) set of EVs arrived at time slot \(t\)
3. \(M^t \leftarrow R^t \cup \{i : i \in T, \text{ and } \sum_j q_j \leq D_i\}\)
4. Based on the residual in interval \([t, T]\), use algorithm \(A\) to allocate EVs in set \(R^t\) assuming no further arrivals
5. \(S_{iCS,t} \leftarrow \{i : i \in M^t \text{ and } i \text{ is admitted at } t\}\)
6. \(\hat{\Gamma}_{A,R^t} \leftarrow \sum_{i \in S_{iCS,t}} v_i\)
7. Assume all reserved resources are freed at time slots \(t, t+1, \ldots, T\)
8. \(r_{i,t} \leftarrow \text{remaining demand of } i \text{ at } t, \forall i\)
9. \(D_i^t \leftarrow r_{i,t}, v_i^t \leftarrow \frac{r_{i,t}}{D_i^t}, v_i^t \in M^t\)
10. Run \(A\) on \(M^t\) using \(D_i^t\) and \(v_i^t\) and reconstruct \(S_{iCS,t}\)
11. if \(\Gamma_{A,M^t} \leftarrow \Gamma_{A,R^t}\) then
   12. Use original values
13. else
   14. Use the second schedule
15. Use the first schedule

The iCS is summarized as Algorithm 7. At slot \(t\), the iCS compares two scheduling results returned by \(A\) and chooses among them. In the first scheduling, the iCS keeps all reserved resources in interval \([t, T]\) intact. Then, for utilizing the remaining resources, the algorithm runs \(A\) over arrived EVs at time slot \(t\). In this case, the total revenue obtained by the active EVs (i.e., EVs that are available but not received their entire demand yet) is denoted by \(\hat{\Gamma}_{A,R^t}\) (Line 6 of the algorithm).

In the second scheduling, the iCS considers the case that it can sacrifice the previously admitted EVs by canceling their reservations and allocating the freed resources to more valuable demands. For this purpose, the algorithm modifies the demand and valuation of the previously admitted EVs such that each demand is replaced by the EVs remaining demand, and the valuation of the EV is proportionally calculated based on the remaining demand (Line 9 of the algorithm) so that the unit values of EVs do not change. Then, the iCS runs \(A\) on set of active EVs \(M^t\) where the corresponding gain is denoted by \(\hat{\Gamma}_{A,M^t}\). If \(\Gamma_{A,M^t} \leftarrow \hat{\Gamma}_{A,R^t}\), the iCS forgets the previously admitted EVs and follows the second scheduling.

### VI. Simulation Results

**Simulation Setup and Overview:** We consider charging scheduling of EVs during a period of 12 time slots of length 1 hour (e.g., from 08:00 to 20:00). We gathered information of 12 popular EV models in the market to use in the simulation. Each EV model is characterized by its battery capacity as shown in Table III and maximum charging rate. Tesla models can get charged with up to 100 kW using Tesla super chargers. Also, all other models have rapid DC charging capability (up to 50 kW) with CHAdeMO method. This setting for charging rates is in accordance with future DCFC systems that can have several fast DC chargers installed in the charging station. The peak intervals include 08:00-10:00, 12:00-14:00, and 18:00-20:00 according to NHTS survey [4], [35]. The probability of an EV’s arrival during the peak hours is two times higher than the off-peak periods. Demands are uniform random values from \(\left[\frac{1}{2}U_i, U_i\right]\) where \(U_i\) is the battery capacity for EV \(i\) in Table III. The deadline of each EV is set according to slackness parameter \(s\) (with default value of 1.2), its demand and maximum charging rate of the battery as \(d_i = a_i + \left[\frac{D_i}{\hat{s}}\right] - 1\). EVs are assigned to different CSs randomly and given as input to the algorithms. The willingness to pay by each user for one kWh of power is a random uniform number from interval \(\left[\frac{1}{2}z, \frac{3}{2}z\right]\) where \(z = 0.11\) is national average price of electricity in the US [36]. In the simulation, the results are plotted with 95% confidence level and each point represents average result of 50 random scenarios. Table IV explains the comparison algorithms including current implemented algorithm in Caltech ACN referred to as iOLP [6] and its adapted version for the fractional revenue model FoLP. The iOLP and FoLP algorithms runs as follows: at each time slot assume that there will be no further arrivals and solve the optimization problem (e.g., by commercial solvers) for the current set of EVs and their remaining demands. In Table IV, the letters “I”, “F” and “O” in front of the algorithms’ name refer to integral, fractional and online types, respectively.

### VI. Simulation Results

**Table III:** EV models and their battery capacity.

<table>
<thead>
<tr>
<th>Model</th>
<th>Battery</th>
<th>Model</th>
<th>Battery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mitsubishi i-MiEV</td>
<td>16 kWh</td>
<td>Citroen C-Zero</td>
<td>14 kWh</td>
</tr>
<tr>
<td>Peugeot iOn</td>
<td>16 kWh</td>
<td>Honda</td>
<td>25.5 kWh</td>
</tr>
<tr>
<td>Hyundai Kona</td>
<td>64 kWh</td>
<td>Nissan LEAF</td>
<td>40 kWh</td>
</tr>
<tr>
<td>Hyundai Ioniq</td>
<td>28 kWh</td>
<td>BMW i3</td>
<td>22/33 kWh</td>
</tr>
<tr>
<td>Tesla Model S/X</td>
<td>60/100 kWh</td>
<td>Kia Soul EV</td>
<td>27 kWh</td>
</tr>
</tbody>
</table>

The following theorem characterizes the competitive ratio of the iCS in a special case.

**Theorem 5.** Let \(A\) be iCS in iCS algorithm and \(m = 1\). Assuming that EVs are released in \(b\) distinct groups where arrival time of EVs in each group are the same, the iCS is \(b(1 + \frac{p-q}{p-q})\)-competitive with optimal offline solution, where \(p\) is the station peak and \(q = \max(k_i, i = 1, \ldots, n)\).
The proposed algorithms are compared to the optimal offline solution of integral model (IOPT) and IOCS. Note that comparison with offline optimal could be considered as a baseline comparison with the category of approximate offline algorithms in integral model [39].

Recall that fCS is optimal offline solution. The notable observations are as follows: (i) the general trend in scenario of Fig. 2 is that by increasing the number of EVs, total revenue increases. This is because with more number of EVs, the scheduler has more freedom to choose more valuable EVs. (ii) as explained in Section IV, under fractional charging model, better results are expected due to increased scheduling flexibility in CSs. According to the simulation data that we extracted from Fig. 2, the gain obtained by fCS and FOCS in Fig. 2 are respectively 12% and 10% more than the gain of the fCS and FOCS. (iii) In Fig. 2a and Fig. 2b, sum of the local peaks is less than or equal to the global peak while in Fig. 2c this sum is two times greater than the global peak. Consequently, the total revenue significantly improved when the number of CSs is increased from 2 to 4 while there is a slight improvement from 4 to 8 CSs as the global peak constraint prevents the algorithms from charging more EVs. (iv) in integral revenue model, the proposed IOCS algorithm acts significantly better than IOCS. In particular, IOCS improves IOLP by 38%, 36%, and 32% for m = 2, m = 4 and m = 8, respectively. In fractional revenue model, however, there is a very slight difference between FOCS and FOCS. (v) IOCS approximates IOPT by 97%, 94%, and 95% for m = 2, m = 4, and m = 8, respectively. On average, IOCS is 90% of IOPT and, FOCS is 94% of its optimal offline solution, FCS. (vi) finally, the results depict that IOCS achieves much better results in practice as compared to the theoretical approximation ratio that characterizes the performance in worst-case scenario.

**Comparison Based on Actual Peak:** The constraint set in the SPAN assures that any feasible solution respects the local and global peak constraints. An efficient scheduling algorithm may take a further step by not only satisfying the peak constraints but to further reduce the peak as much as possible. The proposed offline algorithms (i.e., IOPT and FOCS) apply valley-filling policy to reduce the peak. The online algorithms (i.e., IOCS and IOPT) do not apply the same policy as they do not have future knowledge to be able to balance allocated resources. To investigate the effect of employed valley-filling strategy, we conducted a set of simulations by varying local peak constraints. The results of IOCS is compared to IOPT offline solution of integral model (IOPT) and IOCS. Note that comparison with offline optimal could be considered as a baseline comparison with the category of approximate offline algorithms in integral model [39].

Recall that fCS is optimal offline solution. The notable observations are as follows: (i) the general trend in scenario of Fig. 2 is that by increasing the number of EVs, total revenue increases. This is because with more number of EVs, the scheduler has more freedom to choose more valuable EVs. (ii) as explained in Section IV, under fractional charging model, better results are expected due to increased scheduling flexibility in CSs. According to the simulation data that we extracted from Fig. 2, the gain obtained by fCS and FOCS in Fig. 2 are respectively 12% and 10% more than the gain of the fCS and FOCS. (iii) In Fig. 2a and Fig. 2b, sum of the local peaks is less than or equal to the global peak while in Fig. 2c this sum is two times greater than the global peak. Consequently, the total revenue significantly improved when the number of CSs is increased from 2 to 4 while there is a slight improvement from 4 to 8 CSs as the global peak constraint prevents the algorithms from charging more EVs. (iv) in integral revenue model, the proposed IOCS algorithm acts significantly better than IOCS. In particular, IOCS improves IOLP by 38%, 36%, and 32% for m = 2, m = 4 and m = 8, respectively. In fractional revenue model, however, there is a very slight difference between FOCS and FOCS. (v) IOCS approximates IOPT by 97%, 94%, and 95% for m = 2, m = 4, and m = 8, respectively. On average, IOCS is 90% of IOPT and, FOCS is 94% of its optimal offline solution, FCS. (vi) finally, the results depict that IOCS achieves much better results in practice as compared to the theoretical approximation ratio that characterizes the performance in worst-case scenario.

**Comparison Based on Actual Peak:** The constraint set in the SPAN assures that any feasible solution respects the local and global peak constraints. An efficient scheduling algorithm may take a further step by not only satisfying the peak constraints but to further reduce the peak as much as possible. The proposed offline algorithms (i.e., IOPT and FOCS) apply valley-filling policy to reduce the peak. The online algorithms (i.e., IOCS and IOPT) do not apply the same policy as they do not have future knowledge to be able to balance allocated resources. To investigate the effect of employed valley-filling strategy, we conducted a set of simulations by varying local peak constraints. The results of IOCS is compared to IOPT offline solution of integral model (IOPT) and IOCS. Note that comparison with offline optimal could be considered as a baseline comparison with the category of approximate offline algorithms in integral model [39].

Recall that fCS is optimal offline solution. The notable observations are as follows: (i) the general trend in scenario of Fig. 2 is that by increasing the number of EVs, total revenue increases. This is because with more number of EVs, the scheduler has more freedom to choose more valuable EVs. (ii) as explained in Section IV, under fractional charging model, better results are expected due to increased scheduling flexibility in CSs. According to the simulation data that we extracted from Fig. 2, the gain obtained by fCS and FOCS in Fig. 2 are respectively 12% and 10% more than the gain of the fCS and FOCS. (iii) In Fig. 2a and Fig. 2b, sum of the local peaks is less than or equal to the global peak while in Fig. 2c this sum is two times greater than the global peak. Consequently, the total revenue significantly improved when the number of CSs is increased from 2 to 4 while there is a slight improvement from 4 to 8 CSs as the global peak constraint prevents the algorithms from charging more EVs. (iv) in integral revenue model, the proposed IOCS algorithm acts significantly better than IOCS. In particular, IOCS improves IOLP by 38%, 36%, and 32% for m = 2, m = 4 and m = 8, respectively. In fractional revenue model, however, there is a very slight difference between FOCS and FOCS. (v) IOCS approximates IOPT by 97%, 94%, and 95% for m = 2, m = 4, and m = 8, respectively. On average, IOCS is 90% of IOPT and, FOCS is 94% of its optimal offline solution, FCS. (vi) finally, the results depict that IOCS achieves much better results in practice as compared to the theoretical approximation ratio that characterizes the performance in worst-case scenario.
and GreedyRTL [33] where the latter is an approximation
scheduling for single station scenario with EVs having same
arrival time (see the explanations after formulating the SPAN
in Section III-B). The results are shown in Fig. 3. Along with
total revenue in Fig. 3a, we also report total actual peak in
Fig. 3b and percentage of fully charged EVs in Fig. 3c. In Fig.
3a, the results for iCS and GreedyRTL are almost identical
while iOCs is 90% of the other two algorithms, on average.
When $p_j \geq 400$, the total revenue for iCS and GreedyRTL
does not increase in Fig. 3a and percentage of fully charged
EVs is 100 for both algorithms according to Fig. 3c. From
this point and onward, the scheduling is not challenging for
offline methods to obtain optimal answer because of resource
sufficiency. However, it is still a challenge to control total
actual peak for the system. As a result of valley-filling policy,
the value of actual peak of iCS in Fig. 3b remains almost
unchanged for $p_j \geq 400$. iOCs and GreedyRTL, however,
continuously increase the peak demand, since they are not
using any peak shaving approach.

VII. CONCLUSION

This paper proposed offline and online algorithms for the
EV charging scheduling problem under fractional and integral
revenue models in an adaptive charging network (ACN). The
problem is different, and more challenging than the existing
single station EV charging scheduling problems since it
requires respecting the aggregate peak charging demand of
the ACN. As the notable contributions, our proposed online
algorithm for fractional revenue model achieves constant com-
petitive ratio of 2. Moreover, the offline integral algorithm
achieves a theoretical bound on the optimality gap and ap-
proximates the optimum by 92%, on average in experiments.
As a future work, we plan to study the problem under posted
pricing mechanism where the charging station publishes the
unit price of the power (which can be varied over time) and
the users can accept or reject the offer. Another interesting
future direction is to tackle the EV charging scheduling in
market-based scenarios where the prices change over time.

REFERENCES

are-about-to-boost-global-power-demand-300-fold,” 2017.
[2] S. Zhao, X. Lin, and M. Chen, “Peak-minimizing online ev charging:
Price-of-uncertainty and algorithm robustification,” in Proc. of IEEE
INFOCOM, 2015.
1553, 2015.
decision algorithm for electric vehicles without future information,”
IEEE Transactions on Smart Grid, vol. 6, no. 6, pp. 2810–2824, 2014.
for office buildings with workplace ev charging and renewable energy,”
IEEE Transactions on Transportation Electrification, vol. 3, no. 1,
electric vehicle charging selection algorithm in power systems,” IEEE
vehicle charging with electricity cost,” in Proc. of ACM e-energy,
low-complexity electric vehicle charging scheduling: optimality and
scalability,” IEEE Transactions on Power Systems, vol. 32, no. 2,
[10] S. Chen and L. Tong, “iEMS for large scale charging of electric vehicles:
Architecture and optimal online scheduling,” in Proc. of SmartGrid-
Connect, 2012.
An online auction framework for electric vehicle park-and-charge,” in
Jennings, “An online mechanism for multi-unit demand and its application
to plug-in hybrid electric vehicle charging,” Journal of Artificial
online economic dispatching for microgrids,” IEEE Transactions on
Media, 2013.
integrity gaps for capacitated network design and covering problems,” tech. rep., Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 1999.
[18] J. C. Mukherjee and A. Gupta, “Distributed charge scheduling of
plug-in electric vehicles using inter-aggregator collaboration,” IEEE
electric vehicles on the expansion planning of distribution systems
considering renewable energy, storage and charging stations,” IEEE
Transactions on Smart Grid, 2017.
“An innovative two-level model for electric vehicle parking lots in
distribution systems with renewable energy,” IEEE Transactions on
[21] Y. He, B. Venkatesh, and L. Guan, “Optimal scheduling for charging and
discharging of electric vehicles,” IEEE transactions on smart grid,
power proﬁle tracking for heterogeneous charging of electric vehicles,”
[23] M. Moradlou, M. P. Moghaddam, M. Haghiﬁam, and E. Alishahi,
“A multi-objective optimization problem for allocating parking lots in
a distribution network,” International Journal of Electrical Power &
exchange scheduling for multimicrogrid system with electric vehicles,”
[25] M. Zeng, S. Leng, Y. Zhang, and J. He, “Qoe-aware power manage-
ment in vehicle-to-grid networks: a matching-theoretic approach,”
pey charging/discharging coordination in smart distribution systems,”
[28] E. L. Karfopoulos and N. D. Hatziygiou, “Distributed coordination of
electric vehicles providing v2g services,” IEEE Transactions on Power
“Predictive scheduling framework for electric vehicles with uncertainties
of user behaviors,” IEEE Internet of Things Journal, vol. 4, no. 1,
and predictive deferrable load control,” in Proc. of IEEE CDC,
with on-arrival commitment,” IEEE Transactions on Intelligent Trans-
mechanisms for deadline-sensitive jobs in large computing clusters,”
[34] E.-C. Chang and C. Yap, “Competitive on-line scheduling with level of

https://mc.manuscriptcentral.com/tsusc-cs
A. Details of the Reconsider(i)

Before giving the details of Reconsider(i), we first explain the intuition behind this algorithm. Note that if in the scheduling problem we set \( T = 1, m = 1, k_i = p^{total}, a_i = d_i = 1 \) for all \( i \), then the problem is equivalent to the well-known 0-1 knapsack problem [16]. In the knapsack problem, a widely used greedy approach sorts items based on their unit values and selects items accordingly. It turns out that in this approach the approximation factor can be arbitrarily bad. For example, consider a knapsack problem with two items with \( v_1 = 2, v_2 = p^{total} \), \( D_1 = 1, and \ D_2 = p^{total} \). Given these values we have \( v_1/D_1 > v_2/D_2 \). To maximize total value of selected items, the optimal solution chooses item 2 while greedy algorithm selects item 1 which results in a worst-case approximation factor of \( c/\text{OPT in general where } c \) is a constant (in this example \( c = 2 \)). To resolve it, one approach is to re-consider unselected items after running greedy algorithm and replace some selected items in the knapsack with unselected ones and then check whether the result is improved or not. In a simple case, only the largest unselected item can be examined which makes a significant theoretical improvement by providing a worst case approximation factor of \( \text{OPT}/2 \).

1CS algorithm leverages the same idea but using a more intelligent replacing method called Reconsider(i). Reconsider(i) is called on every unselected EV \( i \). It tries to find some selected EVs that if they are replaced by EV \( i \), total revenue from the selected EVs increases.

B. Proof of Lemma 1

We prove the theorem by induction.

**Base case**: When \( n = 1 \), there is only one EV that can be easily scheduled in its interval since all charging profiles represent a feasible demand and all resources are free.

**Induction step**: Let \( k \in \mathbb{Z}^+ \) is given and the claim is true for \( n = k \) i.e., EVs \( 1, \ldots, k \) can be feasibly scheduled to receive their reserved resources. Now let \( n = k + 1 \). We claim that EV \( k + 1 \) can be feasibly scheduled in its interval. To prove, assume that this claim does not hold. Therefore, there should be at least one interval say \( I_{t,t'} \) such that \( A^k_{h(k+1)}(t,t') < 0 \) and \( A^k_{h(k+1)}(t,t') \geq 0 \). Then, one of the following cases holds: a) \( I_{t,t'} \notin \mathcal{I}_{a_{k+1},d_{k+1}} \) and b) \( I_{t,t'} \in \mathcal{I}_{a_{k+1},d_{k+1}} \). In case a, since \( I_{t,t'} \notin \mathcal{I}_{a_{k+1},d_{k+1}} \), resources in interval \( I_{a_{k+1},d_{k+1}} \) does not affect remaining resource in \( I_{t,t'} \). Therefore, we have \( A^k_{h(k+1)}(t,t') \geq 0 \). In case b, according to Eq. (3), we have \( R_{k+1} \leq A^k_{h(k+1)}(t,t') \). Also, \( A^{k+1}_{h(k+1)}(t,t') = A^k_{h(k+1)}(t,t') - R_{k+1} \) which gives \( A^k_{h(k+1)}(t,t') \geq 0 \). Consequently, in both cases \( A^k_{h(k+1)}(t,t') \geq 0 \) which is a contradiction. Therefore, the original claim holds.

C. Proof of Lemma 2

By induction.

**Base case**: When \( n = 1 \), the claim holds since \( R_i > \min \{ D_i, \min_{t,t'} A^i_{h(i)}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i} \} \) is not feasible and the gain is maximized with maximum value of \( R_i \) i.e., \( R_i = \min \{ D_i, \min_{t,t'} A^{i+1}_{h(i)}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i} \} \).

**Induction step**: Assume the claim holds for \( n = k; k > 1 \) i.e., \( R^*_i = \min \{ D_i, \min_{t,t'} A^{i+1}_{h(i)}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i} \} \) is the optimal value for \( R_i, i = 1, \ldots, k \). Now let \( n = k + 1 \). We claim that \( R^*_{k+1} = \Gamma \) where \( \Gamma = \min \{ D_{k+1}, \min_{t,t'} A^k_{h(i)}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_{k+1},d_{k+1}} \} \). To prove, assume \( \Gamma \) is not the optimal value of \( R_{k+1} \). Therefore, since according to the definition of \( R_{k+1} \) it always holds that \( R_{k+1} \leq \Gamma \) thus, \( R^*_{k+1} > \Gamma \) and the amount of resource to be reserved for EV \( i \) is decreased by \( \Gamma - R^*_i \). This amount, can be only assigned to EVs \( k + 2 \) to \( n \) since \( R_i \) is set to its maximum value for \( i = 1, \ldots, k \). However, having \( v_k/D_k \geq v_i/D_i \) for \( i = k + 2, \ldots, n \) the optimal total revenue can be increased by setting \( R_{k+1} \) which is a contradiction.

D. Proof of Theorem 3

We first prove the theorem for single station scenario and then extend it to multiple stations. From (6), (7), and (8) we obtain the following result

\[ \text{OPT} - \text{ALG} \leq n \sum_{i=1}^{n} g_{i,t} \leq \text{ALG}, \quad \text{(10)} \]

hence, \( \text{OPT} \leq 2\text{ALG} \).

In multi-station setting, the difference with the previous case is that when \( \Delta^t_i > 0, i \in \{1, \ldots, n\} \), then FOCS may allocate the difference \( \Delta^t_i \) to one or multiple EVs in any CS that might not be \( h(i) \). However, the inequality \( l_{i,t} \leq g_{i,t} \) is still valid as FOCS is centralized and uses a single sorted list for all EVs. Using similar deductions as in the single station setting, it is easy to verify that the competitive ratio of 2 is preserved.

E. Proof of Theorem 4

Without loss of generality we assume that \( a_i = 1, \forall i \), however, the proof holds for any other constant value for arrival time. We sum up all costs of covering dual constraints and then provide a bound for it.

Each EV is either selected or not selected. For the unselected EVs, \( \sum_{j=1}^{m} \sum_{i=1}^{n} p_j \beta(t) \) determines the cost. When Betacover(i) is running as a result of charging request disapproval of EV \( i \), for any previously accepted request \( i' \) the algorithm sets \( \Phi_{i'}(t) \) to a value proportional to \( y_{i'} \) for \( t \leq R(d_i) \) (Line 8 of Betacover(i) algorithm). The followings are proved in [33] for a single station \( h(t) = h(i) = j \):

\[ \sum_{i'=1}^{n} d_{i'} \Phi_{i'}(t) \leq \left[ \frac{p_j - q_j}{s - 1} \right] \sum_{i'=1}^{n} v_{i'} \quad \text{(11)} \]

\[ \sum_{t=1}^{T} p_j \beta(t) \leq \sum_{i=1}^{m} d_{i} \Phi_{i}(t) \quad \text{(12)} \]

For \( m \) CSs, we can obtain the following inequality based on (11) and (12),

\[ \sum_{j=1}^{m} \sum_{t=1}^{T} p_j \beta(t) \leq \sum_{j=1}^{m} \left( \frac{p_j - q_j}{s - 1} \right) \sum_{i=1}^{n} v_{i} \quad \text{(13)} \]
Now for notational convenience let’s define $A_j, B_j$ and $C$ as follows:

$$A_j = \frac{p_j}{p_j - q_j} \cdot \frac{s}{s - 1},$$

$$B_j = \sum_{i: h(i) = j} v_i,$$

$$C = \sum_{j=1}^{m} B_j = \sum_{j=1}^{m} v_i.$$

We can write the right hand side of Eq. (13) as follows:

$$\sum_{j=1}^{m} A_j B_j = \sum_{j=1}^{m} \left[ A_j \left( C - \sum_{i: h(i) \neq j} B_h(i) \right) \right] = \sum_{j=1}^{m} A_j \left( C - \sum_{i: h(i) \neq j} B_h(i) \right) = \sum_{j=1}^{m} A_j \left( C - B_j \right) \leq \sum_{j=1}^{m} A_j C - \sum_{j=1}^{m} A_j \left( C - \max_j B_j \right) = \max_j \{B_j\} \sum_j A_j,$$  

(14)

From (13) and (14) we get

$$\sum_{j=1}^{m} \sum_{t=1}^{T} p_j \beta(t) \leq \max_j \{B_j\} \sum_j p_j \frac{s}{s - 1}.$$  

(15)

For the selected EVs, the covering cost is determined by the term $\sum_i D_i \alpha_i$ in the dual objective which equals to $\sum_{i \in S} v_i$ where $S$ is the set of selected EVs. Therefore, the total cost of covering dual constraints equals to

$$\Lambda = \sum_{i \in S} v_i + \max_j \{B_j\} \sum_j p_j \frac{s}{s - 1} \leq \sum_{i \in S} v_i + \sum_{i \in S} \sum_{j=1}^{m} p_j \frac{s}{s - 1} = \left[ 1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} \cdot \frac{s}{s - 1} \right] \sum_{i \in S} v_i,$$  

(16)

Given that the primal value obtained from tCS is $\Gamma = \sum_{i \in S} v_i$, we get

$$\Lambda \leq \left[ 1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} \cdot \frac{s}{s - 1} \right] \Gamma.$$  

(17)

Finally, considering the fact that $\Lambda \geq \text{OPT}$, we conclude that

$tCS$ is $\left[ 1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} \cdot \frac{s}{s - 1} \right]$-approximation.

**F. Proof of Theorem 5**

In the worst case $b = T$ i.e., there are $T$ groups where at each time slot, a group of EVs arrive. Observe that

$$\text{OPT} \leq \rho \Gamma_{A,R^1} + \cdots + \rho \Gamma_{A,R^T}.$$  

Put it simply, the increase in optimal gain at each time slot is at most equal to maximum gain that can be obtained from arrived EVs at time slot $t$. Moreover, according to the $tCS$, the gain of the algorithm obtained from set of active jobs at each time slot $t$, $\Gamma^t_{tCS}$, is as follows:

$$\Gamma^t_{tCS} = \max\{\Gamma_{A,R^t}, \Gamma_{A,M^t}\}.$$  

Therefore,

$$\Gamma_{A,R^t} \leq \Gamma^t_{tCS}, t = 1, \ldots, T.$$  

Thus, $\text{OPT} \leq T \left( 1 + \frac{\rho}{p - q} \cdot \frac{s}{s - 1} \right)$. If $b < T$, we can obtain

$\text{OPT} \leq b \left( 1 + \frac{\rho}{p - q} \cdot \frac{s}{s - 1} \right)$ by the same analysis.

**G. Proof of Theorem 1**

By utilizing Lemma 2, $tCS$ sets $R_i$ to its optimal value for EV $i, i = 1, \ldots, n$. Based on Theorem 1, since it is feasible to allocate $R_i$ to EV $i, i = 1, \ldots, n$, the total gain by $tCS$ is optimal.

**H. Proof of Theorem 2**

The algorithm starts by sorting the charging profiles which costs $O(n \log n)$. Then, in the first “for” loop in Lines 4–8, the algorithm calculates $R_i$ for $i = 1, \ldots, n$. This requires us to check that for each EV $i$, there are enough available resources in all time intervals in the set $I_{a_i,d_i}$. By definition, number of these times slots is $(T - d_i + 1)\alpha_i$ which is $O(T^2)$ and their length varies from 1 to $T$. Therefore, the complexity of the first “for” is $O(nT^2)$. Finally, in the second “for” loop where the algorithm makes re-allocations, it should check all previously allocated EVs in their availability interval which can be done in $O(n^2T)$ and dominates the cost of sub procedure SMARTALLOCATE. Therefore, the total cost is $O(n^2T + nT^2)$.