Peak-Minimizing Online EV Charging

Shizhen Zhao and Xiaojun Lin
School of ECE,
Purdue University, West Lafayette, IN, USA
Email: {zhao147, linx}@purdue.edu

Minghua Chen
Department of Information Engineering,
The Chinese University of Hong Kong
Email: minghua@ie.cuhk.edu.hk

***Abstract*—In this paper, we consider an aggregator that manages a large number of Electrical Vehicle (EV) charging jobs, each of which requests a certain amount of energy that needs to be charged before a deadline. The goal of the aggregator is to minimize the peak consumption at any time by planning the charging schedules in order. A key challenge that the aggregator faces in the planning is that there exists significant uncertainty in future arrivals of EV charging jobs. In contrast to existing approaches that either require precise knowledge of future arrivals or do not make use of any future information at all, we consider a more practical scenario where the aggregator can obtain a limited amount of future knowledge. Specifically, we consider a model where a fraction of the users reserve EV charging jobs (with possible reservation uncertainty) in advance and we are interested in understanding how much limited future knowledge can improve the performance of the online algorithms. We provide a general and systematic framework for determining the optimal competitive ratios for an arbitrary set of reservation parameters, and develop simple online algorithms that attain these optimal competitive ratios. Our numerical results indicate that reservation can indeed significantly improve the competitive ratio and reduce the peak consumption.**

I. INTRODUCTION

Electrification of transportation is a major national priority due to its environmental and societal benefits. Converting fossil-fueled vehicles to EVs can increase the penetration of cleaner energy sources, improve energy efficiency, decrease the reliance on fossil fuels, and thus be more sustainable [1]. However, large-scale transportation electrification comes with both challenges and opportunities. In the US, transportation consumes 29% of total energy, while electricity consumes 40%. Thus, once a significant portion of transportation is electrified, if left uncontrolled they will significantly stress the capacity of the electrical grid. On the other hand, EV charging is a typical example of a deferrable load, and there is often considerable flexibility in the charging schedule, which may be exploited for the purpose of demand response to improve the overall system stability and efficiency [2].

In this paper, we are interested in developing intelligent EV charging algorithms under the scenario of an EV aggregator serving potentially a large number of EVs. Such an EV aggregator can represent a parking lot for an apartment complex and/or an office building that manages the EV charging of their customers. The EVs arrive with charging requests, each of which has a deadline for the charging request to be completed. This scenario was studied in [10][11] with the goal of minimizing the total energy cost of the aggregator subject to time-of-day pricing. In contrast, in this paper we focus on a different optimization objective, where the EV aggregator attempts to minimize the peak energy consumption at any given time during a billing period. Such a peak-minimizing objective is relevant due to the following reasons. First, meeting a higher peak demand requires a larger generation capacity, which is usually more expensive and "dirtier". Further, a large peak demand closer to the system capacity can potentially be a source of grid instability. Hence, from the utility provider's point of view, it is beneficial if the peak energy consumption can be reduced. In this regard, having the aggregator to reduce the peak consumption of a set of EVs can be taken as a first step towards reducing the overall peak consumption of the grid. Second, in light of the importance of controlling the peak consumption, some utility providers have introduced some forms of *peak-based pricing*. In this tyoe of pricing schemes, the customers are charged based on both the total usage in a billing period and the maximum (peak) usage at any time in the billing period. Specifically, if a customer's energy consumption is given as a sequence $(E_1, E_2, ..., E_n)$, then the total bill is of the form $c_1 \sum_i E_i + c_2 \max_i\{E_i\}$ [9]. In typical schemes (e.g., the Fort Collins Utility [4]), the unit charge for peak usage c_2 (between 4.75\$/kWh and 5.44\$/kWh) is over 100 times more than the unit charge for total usage c_1 (between 0.0245\$/kWh and 0.0367\$/kWh). Under this type of pricing schemes, when the aggregator defers EV charging jobs, the total energy consumption does not change. It is the peak demand that is changed. Hence, minimizing the EV aggregator's operating cost is also equivalent to minimizing the peak consumption.

A main challenge for designing peak-minimizing EV charging algorithms is the uncertainty of future arrivals and departures of EV charging requests. If all future EV charging jobs are known in advance, one can then readily compute the optimal charging schedule that minimizes the peak [5][6]. Unfortunately, knowing the entire future demand is usually infeasible in practice. On the other hand, if the statistics of the future demand are known, one can potentially formulate a stochastic control problem, e.g., as a Markov Decision Program (MDP) [13]. However, estimating the future statistics may not be easy either. If the statistics are incorrect, the performance guarantee from the MDP solution will also become unreliable. (Further, MDP typically suffers from the "curse of dimensionality" when the problem size is large.) At the other extreme, one may choose not to obtain or use any future information at all. For example, a myopic policy may compute the optimal charging schedule based only on the jobs that have arrived before time t and that remain to be served, as if it is an offline problem. Then, the myopic policy can use the corresponding optimal decision at time t [6]. However, we will show later that such a heuristic algorithm can have extremely poor performance under non-stationary demand. The Lyapunov optimization approach in [10][11] also does not require any prior knowledge of future information. However, it is commonly used to derive algorithms that approach the optimal *average*-performance (e.g., the operating cost) as queues [10] or other key system resources (e.g., battery size [11])) go to infinity. As such, it is unclear how it can be applied to obtain solutions for achieving optimal *peak*-performance with finite deadline or resource constraints. Another approach to deal with future uncertainty is to develop competitive online algorithms. [12] has studied EV charging using such a framework. However, the goal there is not to reduce the peak either. In constrast, our problem is most similar to the speed-scaling problem in the CPU scheduling literature [5][7]. An online algorithm called BKP is shown to achieve a competitive ratio of e. In other words, no matter what the future workload patterns look like, the peak of the BKP algorithm is at most e times the peak of the optimal offline algorithm. Further, the ratio e is shown to be the optimal competitive ratio [7], in the sense that there exists worst-case workload patterns such that no online algorithms can perform better. This algorithm can also be used in the setting of EV charging with the same performance bound, when there is no future information at all.

In reality, a competitive ratio of e is still quite large. (Having to pay e times more on the peak charge seems to be a costly proposition.) How can we design online algorithms that can achieve even better competitive ratios? We believe that a promising avenue is to study the settings that are in-between the two extremes described above, with either full future information or no future information. In practice, we usually have some degree of limited future information, which intuitively should help us design more efficient EV-charging algorithms. There are a number of interesting questions. First, how much benefit can we obtain by leveraging such limited future information, as compared to utilizing no future information at all? Second, how can we design online algorithms that optimially realize such benefits.

Specifically, in this work we study a practical setting with reservation. Customers can make reservations in advance for future EV-charging jobs. Note that such reservations naturally "reveal" future information to the aggregator, without the need for expensive forecasting[1]. In practice, the aggregator may offer price incentives to encourage reservations. However, not all customers will reserve in advance, and hence there will still be uncertainty due to "walk-in" jobs[2]. Suppose that the reserved jobs account for at least p fraction of the total EV-demand (the value of p will likely increase as the price incentives become more attractive), we can then study the two questions outlined earlier. To the best of our knowledge, competitive online algorithms under reservation have not been studied in the literature. A key contribution of our work is to develop a general framework that can quantify the best competitive ratio under an arbitrary set of parameters. Specifically, this general framework not only gives a lower bound on the optimal competitive ratio under each set of parameters, but also gives the corresponding online algorithm with a competitive ratio that attains this lower bound. Using these results, we can then quantify the gain in the optimal competitive ratio and the reduction of the peak energy consumption as key reservation parameters change (e.g., the reservation time L and the ratio p of the reserved demand). For example, when 60% of the jobs are reserved $\frac{1}{4}$ of the total time horizon ahead of their arrival times, the optimal competitive ratio is reduced to 1.39. Our numerical results indicate that our proposed online algorithm is effective in reducing the peak consumption.

The rest of the paper is organized as follows. In Section II, we present the system model. We discuss the necessity for the design of better online peak-minimizing algorithms in Section III. In Section IV, we develop a general framework that can quantify the best competitive ratio under an arbitrary set of parameters, and propose an online algorithm, called EPS, that attains the optimal competitive ratio. The optimal competitive ratio involves solving a linear programming problem. We propose an effective way of reducing the complexity of the linear program, and then study the impact of reservation on the optimal competitive ratio in Section V. Finally, we conduct simulations to demonstrate the effectiveness and robustness of our proposed EPS algorithm in Section VI.

II. System Model

We consider an aggregator managing the EV-charging jobs[3] of its customers. We assume that time is slotted. Let T be the total number of time-slots in a billing period, which can be one day or one month depending on the billing policy. We use $t \in \mathbb{T}$ to represent a typical time-slot, where $\mathbb{T} = \{1, 2, ..., T\}$. The goal of the aggregator is to reduce the peak consumption across all time-slots in the billing period. Consider a sequence J of EV-charging jobs. Each job $k \in J$ can be represented by a 4-tuple (s_k, d_k, e_k, v_k), which indicates that this EV arrives at the beginning of time slot $s_k \in \mathbb{T}$, departs at the end of time slot $d_k \in \mathbb{T}$, and requires e_k amount of energy to finish its request (we also refer to e_k as the demand). The 4-th term v_k is the reservation time for the job k. If this EV charging job k is reserved in advance, we will have $v_k < s_k$. Otherwise, $v_k = s_k$, and we refer to the job as a "walk-in" job. In practice, we expect that the aggregator will offer price incentives to

[1] In the literature, another way to capture limited future information is to use the look ahead window [3]. We note the the case with precise look-ahead window can be viewed as a special case of our general model. See the detailed discussion in Section II.

[2] We use the term "walk-in" since it is analogous to patients visiting a doctor's office without appointments.

[3] In this paper, we will use the terms "EVs", "EV charging jobs", or "jobs" interchangeably.

encourage its customers to make reservations in advance. We assume that each *reserved* job k must be reserved L time slots in advance, i.e., $v_k \leq s_k - L$. In other words, only jobs reserved "truly" in advance can qualify for price incentives. Later on, we will study the benefit of reservation as the parameter L varies. Here, we allow v_k to be non-positive, i.e., $v_k \leq 0$, in which case this EV-charging job is known at the beginning of the billing period[4].

With suitable price incentives, we would expect that at least a certain fraction of the users will reserve their EV-charging jobs in advance. This assumption is modeled as follows. Given a sequence of EV arrivals J, let $r_{i,j}^J$ be the total *reserved* demand with arrival time i and departure time j, and let $R_{i,j}^J = \sum_{j'=i}^{j} r_{i,j'}^J$ be the total reserved demand with arrival time i and departure time no greater than j. Similarly, let $a_{i,j}^J$ be the total *walk-in* demand with arrival time i and departure time j, and let $A_{i,j}^J = \sum_{j'=i}^{j} a_{i,j'}^J$ be the total walk-in demand with arrival time i and departure time no greater than j. According to our reservation model, all $r_{i,j}^J$'s are known at least L time-slots ahead of time i, while $a_{i,j}^J$'s can only be known at time i. In order to model the relationship between the reserved demand and the walk-in demand, we assume that the following inequality holds for all i, j,

$$p_l(R_{i,j}^J + A_{i,j}^J) \leq R_{i,j}^J \leq p_u(R_{i,j}^J + A_{i,j}^J), \tag{1}$$

where p_l and p_u are two positive constants that bound the fraction of reserved demand over the total demand. Note that in practice, even if a customer makes reservations, he may not be able to honor the reservation 100% of the time. He may predict his arrival time, deadline, or even demand imprecisely, or he may cancel the reservation altogether. Our model in (1) is sufficiently general to incorporate the case where the reservations are not 100% certain. Specifically, we can view $R_{i,j}^J$ as the mean of the reserved demand, and use $A_{i,j}^J$ to represent both the walk-in demand and the uncertainty from the reservation demand itself.

Note that the above model captures limited future information in two ways. First, each reservation naturally "reveals" to the aggregator about future demand patterns, without the need for expensive forecasting. This revelation property can be particularly useful when the demand patterns exhibits daily changes. Second, the parameters p_l and p_u can be extracted from historical data on consumer behavior, which also represent limited knowledge of the future. Our goal in this paper is thus to study how the aggregator can exploit such limited future information to improve its decisions.

In the literature, a related way to model limited future information is through a look-ahead window, i.e., at time t, future arrivals for the time interval $[t, t+L]$ are known precisely [3]. Note that this precise look-ahead model can be taken as a special case of our model by setting $p_l = p_u = 1$. However, in practice look-ahead information may not be precise either. Our model allows such uncertainty to be captured. Further, in practice, some EV charging jobs may be reserved more than L time-slots ahead, in which case we will obtain some future information beyond L time slots. Thus, our model with limited future information is more general and practical.

Given a sequence J of EV charging jobs, the aggregator needs to determine the amount of energy E_t^J drawn from the power grid at each time slot $t \in \mathbb{T}$. We use $E_J = \{E_1^J, E_2^J, ..., E_T^J\}$ to denote the service profile of the aggregator. We are interested in minimizing the peak consumption, i.e., $\max_t\{E_t^J\}$, subject to the constraint that all jobs are completed before their deadlines.

If all the charging jobs are known in advance, the problem can be written as follows and solved by an offline algorithm like the one in [5].

$$\min_{\text{All jobs are completed before their deadlines}} \max_t\{E_t^J\}. \tag{2}$$

Let $E_{J,\text{off}}^*$ be the optimal offline solution to (2). However, in practice, such perfect future knowledge is hard to obtain. An algorithm π is called an online algorithm if this algorithm computes $E_t^J(\pi)$ based only on the EV jobs arrived or reserved before or at time t. This online algorithm π is called feasible if all jobs are completed before their deadlines. Let $E_J^*(\pi) = \max\{E_t^J(\pi)\}$ be the peak energy drawn from the grid using a feasible online algorithm π. We study the performance of the online algorithm π using its competitive ratio (CR) $\eta(\pi)$, which is defined as the maximum ratio between $E_J^*(\pi)$ and $E_{J,\text{off}}^*$ under all possible job sequences J, i.e.,

$$\eta(\pi) = \max_J \left\{ \frac{E_J^*(\pi)}{E_{J,\text{off}}^*} \right\}.$$

An feasible online algorithm π is called optimal, if it attains the smallest competitive ratio. Our goal in this paper is to find such optimal online algorithms, and reveal how limited future information (e.g., reservation) improves the optimal competitive ratio.

III. Motivation for Better Online Algorithm

Unfortunately, developing competitive online algorithms is not an easy task, either with or without reservation. In this section, we will show that a myopic online algorithm (possibly a very natural one) could perform very poorly. Therefore, it is important to find better algorithms for online EV-charging.

To start with, we briefly review the offline optimal algorithm (called the YDS algorithm) proposed in [5].

A. Review of the Offline-Optimal YDS Algorithm

Let J be a sequence of EV-charging jobs. Define the *intensity* on an interval $I = [i, j]$ with respect to the job sequence J as

$$g_J(I) = \frac{\sum_{i'=i}^{j}(R_{i',j}^J + A_{i',j}^J)}{j - i + 1}. \tag{3}$$

[4]In this paper, we have assumed that the EV-charing jobs are the only jobs that the aggregator needs to control as far as the peak consumption is concerned. If there are other uncontrollable background load that contributes to the peak consumption, it is also possible to extend our model to incorporate background load. For example, assuming that the background load can be estimated in advance, we can treat the background load at time t as a reserved demand known at time $v_k = 0$ with $s_k = d_k = t$. The rest of the model will then apply.

Then, the YDS algorithm [5] is re-stated in Algorithm 1.

1 Repeat steps 2-4 until the set J is empty.
2 Let $I^* = [i, j]$ be the time interval with the maximum intensity, i.e., $g_J(I^*) = \max_I\{g_J(I)\}$.
3 Let the service profile during interval I be $E_t^J = g_J(I^*), t \in I$, and serve all the jobs within the interval I^*, i.e., all jobs satisfying $i \le s_k \le d_k \le j$, by the *earliest deadline* policy.
4 Modify the job sequence J as if the time interval I^* does not exist. More precisely, first delete from J all the jobs within the interval I^*. Second, all deadlines $d_k \ge i$ are reduced to $\max\{i-1, d_k - (j-i+1)\}$, and all arrival times $s_k \ge i$ are reduced to $\max\{i, s_k - (j-i+1)\}$.

Algorithm 1: Offline-optimal YDS algorithm

Note that we do not update the reservation times in step 4 of the YDS algorithm. This is because that the reservation times do not matter in the offline optimal algorithm, when all future jobs are known in advance. Furthermore, it is easy to see that the intensity of the maximum-intensity interval decreases as the YDS algorithm proceeds. Therefore, the optimal offline value $E^*_{J,\text{off}}$ of the peak consumption is given by the maximum intensity at the first run of step 2, i.e.,

$$E^*_{J,\text{off}} = \max_I\{g_J(I)\}. \tag{4}$$

B. A Myopic Online Algorithm

The YDS algorithm cannot be used online when future EV-charging jobs are not known in advance. The following myopic algorithm represents a natural online algorithm. At each time slot t, the myopic online algorithm uses the YDS algorithm to compute the optimal serving rate based only on the remaining workload and the future reserved workload known at time t. It then uses this rate to serve its known workload by the *earliest deadline* policy. A similar idea has been proposed in [6]. However, we will show that this myopic algorithm could have an arbitrarily poor competitive ratio (CR).

Lemma 1. *If there is no reservation, the competitive ratio η^* of the myopic algorithm can be arbitrarily large as $T \to \infty$, i.e., for any constant $M > 0$, there exists $T > 0$ and an arrival pattern, such that the peak rate under the myopic algorithm is at least M times the optimal peak rate under the optimal offline algorithm.*

Proof: See Section VII-A. ■

One would expect that reservation may improve the performance of the myopic algorithm. Unfortunately, the following lemma states that no matter how large is the fraction of the reserved demand, the myopic online algorithm still has an arbitrarily large CR.

Lemma 2. *Under our reservation model (see Section II), for any L and $p_l < p_u = 1$, the competitive ratio η^* of the myopic algorithm can be arbitrarily large as $T \to \infty$.*

Proof: See Section VII-B. ■

The above two lemmas indicate that, if EV charing is not scheduled properly, the aggregator may potentially face a huge peak rate. Hence, it is important to design better (even optimal) online algorithms.

In fact, if there is no reservation, an online algorithm called BKP is proposed in [7] and shown to achieve a CR of e. Further, this CR e is shown to be the optimal. However, in practice e is still a large number. In this work, we are interested in how limited future knowledge (through reservation) may help us to significantly improve the competitive ratio. Unfortunately, the techniques for proving the competitive ratio and its optimality in [7] are very specific and seems difficult to handle reservation. In the next section, we will develop a very general framework that can lead to optimal online algorithms under an arbitrary set of reservation parameters.

IV. Optimal Peak-Minimizing Online EV Charing

In this section, we propose a general framework for designing optimal online EV-charging algorithms with reservations. For ease of exposition, we will focus on the case where p_u is 1 in constraint (1). In other words, the reserved demand and the walk-in demand now satisfy the following simplified constraint:

$$p(R^J_{i,j} + A^J_{i,j}) \le R^J_{i,j} \le R^J_{i,j} + A^J_{i,j}. \tag{5}$$

We note that there is no loss of generality in this simplification. If $p_u \ne 1$, we know that there will be at least $(\frac{1}{p_u} - 1)R^J_{i,j}$ future walk-in demand. Thus, we can view this part of walk-in demand as some pseudo "reserved demand". Specifically, let $\tilde{R}^J_{i,j} = R^J_{i,j} + (\frac{1}{p_u} - 1)R^J_{i,j} = \frac{R^J_{i,j}}{p_u}$, and $\tilde{A}^J_{i,j} = A^J_{i,j} - (\frac{1}{p_u} - 1)R^J_{i,j}$, then constraint (1) can be equivalently expressed as

$$\frac{p_l}{p_u}(\tilde{R}^J_{i,j} + \tilde{A}^J_{i,j}) \le \tilde{R}^J_{i,j} \le \tilde{R}^J_{i,j} + \tilde{A}^J_{i,j}.$$

Let $p = \frac{p_l}{p_u}$. The constraint (1) is then converted to the form in (5).

In addition, if we let $C = \frac{1-p}{p}$, constraint (1)can be further simplified as

$$0 \le A^J_{i,j} \le CR^J_{i,j}. \tag{6}$$

The following analysis will be based on constraint (6).

A. Lower Bound on the Competitive Ratio

We first present a lower bound on the competitive ratio (CR) of an arbitrary online algorithm. As readers will see, the lower bound can be obtained by considering the following sequence of job arrivals.

Fix $n \in \mathbb{T}$. Consider a job sequence J_n with the following form. The arrival time of each job $k \in J_n$ satisfies $1 \le s_k \le n$. All jobs have the same deadline n. Further, for all reserved jobs with arrival time i, they are reserved exactly L time-slots ahead, i.e., at time $i - L$. The reserved demand and the walk-in demand satisfy constraint (6). Let $\mathcal{J}_n$ be the set of all J_n's with such form.

Consider an arbitrary feasible online algorithm π_n with CR η_n. We apply this algorithm to an EV-arrival sequence $J_n \in \mathcal{J}_n$. Then, we have the following lemma.

Lemma 3. *Given an online algorithm π_n with CR η_n, its service profile $E_{J_n}(\pi_n) = \{E_1^{J_n}(\pi_n), E_2^{J_n}(\pi_n), ..., E_n^{J_n}(\pi_n)\}$ under an EV-arrival sequence $J_n \in \mathcal{J}_n$ must satisfy*

$$E_t^{J_n}(\pi_n) \leq \eta_n E_{pe}^{J_n}(t), t = 1, 2, ..., n$$

where

$$E_{pe}^{J_n}(t) = \max_{j=1,...,h_n(t)} \left\{ \frac{\sum_{i=j}^{t} A_{i,n}^{J_n} + \sum_{i=j}^{h_n(t)} R_{i,n}^{J_n}}{n-j+1} \right\}, \quad (7)$$

and $h_n(t) = \min\{t+L, n\}$. (In (7), the subscript "pe" stands for "peak estimation".)

Proof: See Section VII-C. ■

The intuition of Lemma 3 is as follows. At time t, the aggregator knows all the walk-in demand with arrival time no greater than t and all the reserved demand with arrival time no greater than $h_n(t) = \min\{t+L, n\}$ (since all the reserved jobs are reserved exactly L time-slots ahead). Based on such known demand, we can take $E_{\text{pe}}^{J_n}(t)$ as the estimate of the peak consumption at time t. In fact, if there were no more new jobs after time t, $E_{\text{pe}}^{J_n}(t)$ would have been the offline-optimal peak service rate. If $E_t^{J_n}(\pi_n) > \eta_n E_{\text{pe}}^{J_n}(t)$, then in the case where there is no demand after time t, π_n will violate the assumption that its CR is η_n.

With Lemma 3 in mind, we study another constraint on π_n. The feasibility of π_n implies that all jobs can be finished before the end of the time slot n (recall that all jobs have the same deadline n). Therefore, we must have

$$\sum_{t=1}^{n} E_t^{J_n}(\pi_n) \geq \sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right). \quad (8)$$

Combining Eqn. (8) with Lemma 3, we then obtain

$$\eta_n \geq \frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)}.$$

Define the following optimization problem:

$$\begin{aligned} \sup_{J_n} \quad & \frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)} \\ \text{subject to} \quad & (6), (7) \end{aligned} \quad (9)$$

Let η_n^* be the optimal solution to the optimization problem (9). Let $\eta^* = \max_{n \in \mathbb{T}}\{\eta_n^*\}$. Then, the following theorem shows that η^* gives a lower bound on the optimal CR, i.e.,

Theorem 4. *For any feasible online algorithm π, its CR must be greater than or equal to η^*.*

Proof: We prove by contradiction. Suppose that there exists a feasible online algorithm $\tilde{\pi}$ with CR $\eta(\tilde{\pi}) < \eta^*$. Let $\epsilon = \eta^* - \eta(\tilde{\pi}) > 0$.

According to the definition of η^*, there must exist $n \in \mathbb{T}$, and a job sequence $J_n \in \mathcal{J}_n$, such that

$$\frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)} > \eta^* - \epsilon = \eta(\tilde{\pi}).$$

Apply the algorithm $\tilde{\pi}$ to the job sequence J_n. According to Lemma 3, we must have $E_t^{J_n}(\tilde{\pi}) \leq \eta(\tilde{\pi}) E_{\text{pe}}^{J_n}(t)$. Then,

$$\begin{aligned} \sum_{t=1}^{n} E_t^{J_n}(\tilde{\pi}) \quad & \leq \quad \eta(\tilde{\pi}) \sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t) \\ & < \quad \sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right) \end{aligned}$$

Thus, some job with deadline $d \leq n$ cannot be finished before its deadline, which contradicts to the assumption that $\tilde{\pi}$ is feasible. ■

In general, the optimization problem (9) can be easily converted into a linear programming problem and solved using standard solvers. We will provide more details in Section V.

Remark 1. *Our formulation of the CR in (9) shares some similarity to the results in [14]. However, [14] does not consider reservation, and there is substantial difficulty in extending the techniques in [14] to the case with reservation. Specifically, a key step in [14] is to show that the problem with variable deadlines has the same CR as the problem with a single deadline (see Theorem 4.26 in [14]). However, for our reservation model, there is another degree of freedom, i.e., the time when the job is reserved. The formulation in (9) suggests that we may focus on the case when the jobs are reserved least in advance (i.e., exactly L time-slots ahead). However, it is unclear that how to generalize the techniques of [14] to show that the problem when reservation can be made at least L time-slots ahead of arrival time also has the same CR as the problem when all reservations are made exactly L time-slots ahead of arrival time. In this paper, we use a different strategy: in Theorem 4, we only show that (9) provides a lower bound on the CR. In the following, we then provide an online algorithm that attains this lower bound, thus avoiding the above difficulty. This technique may also be of independent interest for other problem settings.*

B. Optimal Online Algorithms

Interestingly, the optimization problem (9) not only gives a lower bound on the competitive ratio, but also leads to an online algorithm that can attain the lower bound as we will demonstrate below. Next, we propose the Estimated Peak Scaling (EPS) algorithm, and show that the competitive ratio of this online algorithm achieves the lower bound η^*.

Given a sequence J of EV-charging jobs (jobs in J could have different deadlines), let $J(t) \subseteq J$ be the set of jobs known before or at time t, which includes all the walk-in jobs with arrival time no greater than t, and all the reserved jobs with reservation time no greater than t. Then, the EPS algorithm is formally stated as follows.

The following theorem states that the EPS algorithm is a feasible online algorithm with competitive ratio η^*. Thus, the

Input: Job sequence J, time slot t

1. Assume that there is no new jobs after time t, use the YDS algorithm on the known jobs $J(t)$ to compute the optimal peak, i.e., $E^*_{J(t),\text{off}}$ as if it is an offline problem. Let $E^J_{\text{pe}}(t) = E^*_{J(t),\text{off}}$.
2. Set $E^J_t = \eta^* E^J_{\text{pe}}(t)$.
3. Serve jobs by the *earliest deadline* policy. Specifically, we sort all unfinished EV jobs with arrival time no greater than t according to their deadlines in an ascending order, i.e., $d_{k_1} \le d_{k_2} \le ...$ Then, we use E^J_t amount of energy to charge the EV k_1, and then $k_2, k_3, ...$ until all these EV jobs are completed or the amount of energy E^J_t is exhausted.

Algorithm 2: EPS algorithm

EPS algorithm is an optimal online algorithm.

Theorem 5. *Given any job sequence J, the EPS algorithm satisfies the following two requirements:*

1) *(η^* optimality) at each time slot t, the service rate E^J_t satisfies $E^J_t \le \eta^* E^*_{J,\text{off}}$;*
2) *(feasibility) all jobs can be completed before their deadlines.*

The first part of Theorem 5 is easy. Note that since $J(t) \subseteq J$, we must have $E^*_{J(t),\text{off}} \le E^*_{J,\text{off}}$. Then,

$$E^J_t = \eta^* E^J_{\text{pe}}(t) = \eta^* E^*_{J(t),\text{off}} \le \eta^* E^*_{J,\text{off}}.$$

Now, we focus on the second part. The proof of the feasibility of the EPS algorithm is based on the following lemma.

Lemma 6. *A sufficient and necessary condition for a service profile $E_J = \{E^J_1, E^J_2, ..., E^J_T\}$ to be feasible, i.e., all jobs can be completed before their deadlines, is that for all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$, the following inequality holds,*

$$\sum_{t=t_1}^{t_2} (A^J_{t,t_2} + R^J_{t,t_2}) \le \sum_{t=t_1}^{t_2} E^J_t.$$

Proof: See Section VII-D. ■

Now, we are ready to present the proof of Theorem 5.

Proof: Based on Lemma 6 and the above discussion, we only need to show that for all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$,

$$\sum_{t=t_1}^{t_2} (A^J_{t,t_2} + R^J_{t,t_2}) \le \sum_{t=t_1}^{t_2} \eta^* E^*_{J(t),\text{off}}.$$

Equivalently, we need to show that

$$\eta^* \ge \frac{\sum_{t=t_1}^{t_2} (A^J_{t,t_2} + R^J_{t,t_2})}{\sum_{t=t_1}^{t_2} E^*_{J(t),\text{off}}}. \tag{10}$$

To show inequality (10), we need to draw a connection between the right hand side (R.H.S.) of (10) and the optimization problem (9). We first simplify (9) by substituting $A^{J_n}_{t,n}$ by a_t, $R^{J_n}_{t,n}$ by r_t, and $E^{J_n}_{\text{pe}}(t)$ by b_t. Then, (9) can be transformed to the following equivalent optimization problem:

$$\begin{aligned} \max_{a_t, r_t \ge 0} \quad & \frac{\sum_{t=1}^{n} (a_t + r_t)}{\sum_{t=1}^{n} b_t} \\ \text{subject to} \quad & b_t = \max_{j=1,...,h_n(t)} \left\{ \frac{\sum_{i=j}^{t} a_t + \sum_{i=j}^{h_n(t)} r_t}{n-j+1} \right\} \\ & 0 \le a_t \le C r_t \end{aligned} \tag{11}$$

For $n = t_2 - t_1 + 1$, the optimal solution of the optimization problem (11) is then $\eta^*_{t_2-t_1+1}$.

We now consider (10). Since the job sequence J satisfies (6), we must have $0 \le A^J_{t,t_2} \le C R^J_{t,t_2}$ for all $t = t_1, ..., t_2$. Suppose that the following inequality holds,

$$E^*_{J(t),\text{off}} \ge \max_{j=t_1,...,h'(t)} \left\{ \frac{\sum_{i=j}^{t} A^J_{i,t_2} + \sum_{i=j}^{h'(t)} R^J_{i,t_2}}{t_2 - j + 1} \right\}, \tag{12}$$

where $h'(t) = \min\{t + L, t_2\}$. Then, if we substitute A^J_{t,t_2} by a'_{t-t_1+1}, R^J_{t,t_2} by r'_{t-t_1+1}, and $E^*_{J(t),\text{off}}$ by b'_{t-t_1+1} for all $t = t_1, ..., t_2$, we must have that the R.H.S. of (10) is no greater than the optimal value of the following optimization problem.

$$\begin{aligned} \max_{a'_t, r'_t \ge 0} \quad & \frac{\sum_{t=1}^{t_2-t_1+1} (a'_t + r'_t)}{\sum_{t=1}^{t_2-t_1+1} b'_t} \\ \text{subject to} \quad & 0 \le a'_t \le C r'_t \\ & b'_t \ge \max_{j=1,...,h_{t_2-t_1+1}(t)} \left\{ \frac{\sum_{i=j}^{t} a'_t + \sum_{i=j}^{h_{t_2-t_1+1}(t)} r'_t}{t_2 - t_1 + 1 - j + 1} \right\} \end{aligned} \tag{13}$$

It is easy to see that the optimal value of (13) is smaller than or equal to the optimal value of (11) with n replaced by $t_2 - t_1 + 1$. Therefore,

$$\text{R.H.S. of } (10) \le \eta^*_{t_2-t_1+1} \le \eta^*,$$

where the second inequality comes from the fact that $\eta^* = \max_{n \in \mathbb{T}} \{\eta^*_n\}$.

Based on the above discussion, it only remains to prove Eqn. (12). Recall that $E^*_{J(t),\text{off}}$ is equal to the maximum intensity over all possible intervals (see Section III-A). Consider only a subset of intervals as follows.

$$\mathcal{I} = \{[t_1, t_2], [t_1 + 1, t_2], ..., [h'(t), t_2]\}.$$

We must have

$$E^*_{J(t),\text{off}} = \max_{I} \{g_{J(t)(I)}\} \ge \max_{I \in \mathcal{I}} \{g_{J(t)(I)}\}. \tag{14}$$

For each interval $I = [j, t_2] \in \mathcal{I}$, the intensity with respect to $J(t)$ is given by (3), i.e.,

$$g_{J(t)}(I) = \frac{\sum_{i=j}^{t_2} (A^{J(t)}_{i,t_2} + R^{J(t)}_{i,t_2})}{t_2 - j + 1}. \tag{15}$$

Note that at time $t = t_1, ..., t_2$, for any walk-in job k that contributes to the term $\sum_{i=j}^{t} A^J_{i,t_2}$ (i.e., it arrives no later than t), it must belong to the set of walk-in jobs in $J(t)$. Thus, it must also contribute to the term $\sum_{i=j}^{t_2} A^{J(t)}_{i,t_2}$. Similarly, for any reserved job k that contributes to the term $\sum_{i=j}^{h'(t)} R_{i,t_2}^J$ (i.e., it arrives no later than $h'(t) = \min\{t+L, t_2\}$), it must be reserved no later than $h'(t) - L \le t$. Hence, this job k must belong to the set of reserved jobs in $J(t)$, and thus also contributes to the term $\sum_{i=j}^{t_2} R_{i,t_2}^{J(t)}$. Therefore, we must have

$$\sum_{i=j}^{t_2} (A_{i,t_2}^{J(t)} + R_{i,t_2}^{J(t)}) \ge \sum_{i=j}^{t} A_{i,t_2}^J + \sum_{i=j}^{h'(t)} R_{i,t_2}^J. \tag{16}$$

Combining Eqn. (15), (14) and (16), we immediately obtain

$$E_{J(t),\text{off}}^* \ge \max_{j=t_1,\dots,h'(t)} \left\{ \frac{\sum_{i=j}^{t} A_{i,t_2}^J + \sum_{i=j}^{h'(t)} R_{i,t_2}^J}{t_2 - j + 1} \right\}.$$

Therefore, Eqn. (12) holds, and thus Eqn. (10) follows. We then conclude that the EPS algorithm is a feasible online algorithm with CR η^*. ∎

Remark 2. *The above results can be viewed as a superset of the results in [7][14]. Specifically, when there is no reservation ($p = 0$ or $C = \infty$), the above algorithm reduces to one that is similar to the BKP algorithm [7]. The competitive ratio is also close to e. (It is not exactly e because the time horizon is finite [14].) However, with reservation, the competitive ratio will improve as can be seen soon in Section V.*

V. Computation and Discussion of the Optimal CR

In the previous section, we propose a peak-minimizing online EV-charing algorithm with the optimal competitive ratio (CR). Note that the proposed EPS algorithm needs the value of the optimal CR η^* in its operation (see Algorithm 2). In this section, we propose an effective way of calculating η^*. We also discuss the impact of the demand reservation on the optimal CR η^*.

A. Computation of η^*

Recall that $\eta^* = \max_{n\in\mathbb{T}}\{\eta_n^*\}$. Therefore, to obtain η^*, we need to find an effective way of computing η_n^*, which involves solving the optimization problem (9).

1) Variable Reduction: We first show that when solving (9), we can simply focus on the case where $A_{i,n}^{J_n} = CR_{i,n}^{J_n}$ for all $i = 1, 2, \dots, n$.

Consider an arbitrary J_n satisfying (6). We pick any $t_0 = 1, 2, \dots, n$, and construct J_n' that satisfies the following two constraints:

1) for $i \ne t_0$, $A_{i,n}^{J_n'} = A_{i,n}^{J_n}$, $R_{i,n}^{J_n'} = R_{i,n}^{J_n}$;
2) for $i = t_0$, $A_{i,n}^{J_n'}$ and $R_{i,n}^{J_n'}$ satisfy

$$A_{i,n}^{J_n'} = CR_{i,n}^{J_n'}, A_{i,n}^{J_n'} + R_{i,n}^{J_n'} = A_{i,n}^{J_n} + R_{i,n}^{J_n}. \tag{17}$$

Based on (6) and (17), it is easy to verify that $R_{i,n}^{J_n'} \le R_{i,n}^{J_n}$ for all $i = 1, 2, \dots, n$. Therefore, from Eqn. (7), we then have, for all t and $j = 1, 2, \dots, h(t)$,

$$\begin{aligned} &\frac{\sum_{i=j}^{t} A_{i,n}^{J_n} + \sum_{i=j}^{h(t)} R_{i,n}^{J_n}}{n-j+1} = \frac{\sum_{i=j}^{h(t)} (\mathbb{1}_{\{i\le t\}} A_{i,n}^{J_n} + R_{i,n}^{J_n})}{n-j+1} \\ \ge\ &\frac{\sum_{i=j}^{h(t)} (\mathbb{1}_{\{i\le t\}} A_{i,n}^{J_n'} + R_{i,n}^{J_n'})}{n-j+1} = \frac{\sum_{i=j}^{t} A_{i,n}^{J_n'} + \sum_{i=j}^{h(t)} R_{i,n}^{J_n'}}{n-j+1}, \end{aligned}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function. Thus, we have $E_{\text{pe}}^{J_n} \ge E_{\text{pe}}^{J_n'}$. Then,

$$\frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)} \le \frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n'} + R_{t,n}^{J_n'}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n'}(t)}.$$

We can apply the above procedure for $t_0 = 1, 2, \dots, n$ sequentially. Let $J_n^{(n)}$ be the EV-demand sequence obtained after n iterations. Then, we have $A_{i,n}^{J_n^{(n)}} = CR_{i,n}^{J_n^{(n)}}$, and

$$\frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n} + R_{t,n}^{J_n}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)} \le \frac{\sum_{t=1}^{n} \left(A_{t,n}^{J_n^{(n)}} + R_{t,n}^{J_n^{(n)}}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n^{(n)}}(t)}.$$

Thus, only considering those J_n's satisfying $A_{i,n}^{J_n} = CR_{i,n}^{J_n}$ is sufficient for obtaining the optimal solution of (9).

Based on the above discussion, we can simplify the expression of $E_{\text{pe}}^{J_n}(t)$ as

$$E_{\text{pe}}^{J_n}(t) = \max_{j=1,\dots,h(t)} \left\{ \frac{\sum_{i=j}^{h(t)} (1 + C\mathbb{1}_{\{i\le t\}}) R_{i,n}^{J_n}}{n-j+1} \right\}, \tag{18}$$

and simplify (9) as

$$\begin{aligned} \sup_{J_n} \quad & \frac{(1+C)\sum_{t=1}^{n} R_{t,n}^{J_n}}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)} \\ \text{subject to} \quad & (18). \end{aligned} \tag{19}$$

2) Converting (19) to a Linear Programming (LP) Problem: Eqn. (18) can be converted to a set of linear constraints, i.e.,

$$E_{\text{pe}}^{J_n}(t) \ge \frac{\sum_{i=j}^{h(t)} (1 + C\mathbb{1}_{\{i\le t\}}) R_{i,n}^{J_n}}{n-j+1}, j = 1, \dots, h(t). \tag{20}$$

Define the following fractional LP problem, i.e.,

$$\begin{aligned} \sup_{J_n} \quad & \frac{(1+C)\sum_{t=1}^{n} R_{t,n}^{J_n}}{\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t)} \\ \text{subject to} \quad & (20). \end{aligned} \tag{21}$$

Note that in the optimal solution of (21), Eqn. (18) must hold for all t. Otherwise, we can decrease $E_{\text{pe}}^{J_n}(t)$ to get a better solution of (21). Hence, problem (21) has the same optimal solution as (19).

Finally, note that if all $R_{t,n}^{J_n}$'s and $E_{\text{pe}}^{J_n}(t)$'s are scaled by a constant, both the objective function and the constraint (20) remain the same. Let

$$\sum_{t=1}^{n} E_{\text{pe}}^{J_n}(t) = 1. \tag{22}$$

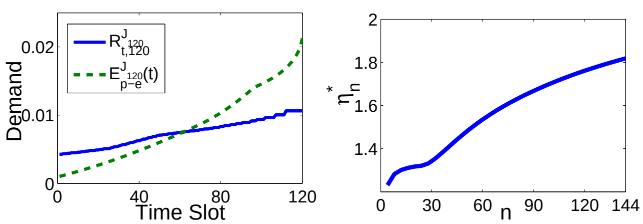
Then, the fractional LP problem (21) can be converted to the following equivalent LP problem, i.e.,

$$\begin{aligned} \sup_{J_n} \quad & (1+C)\sum_{t=1}^{n} R_{t,n}^{J_n} \\ \text{subject to} \quad & (20), (22). \end{aligned} \tag{23}$$

Next, we will solve η_n^* based on (23).

3) An Example: In this section, we use an example to illustrate the shape of the demand in the optimal solution of (23), and how η_n^* varies with respect to n. In this example, we assume that the billing period is a day, and the duration of each time slot is 10 minutes. Therefore, $T = 144$, and $\eta^* = \max_{n=1}^{144}\{\eta_n^*\}$. We assume that the reserved EV charging jobs must be reserved at least 4 hours ahead. Thus, $L = 24$. Further, we assume that $C = 1$, which indicates that at least half of the total demand is reserved demand.

First, we compute η_n^* for a specific value of $n = 120$. We use the MATLAB CVX package [15] to numerically solve (23). The result is $\eta_{120}^* = 1.7614$, and the corresponding $R_{t,120}^{J_{120}}$'s and $E_{\text{pe}}^{J_{120}}(t)$'s are plotted in Fig. 1.(a) ($A_{t,120}^{J_{120}}$ is not shown in this figure because we know $A_{t,120}^{J_{120}} = CR_{t,120}^{J_{120}}$).



(a) Optimal solution of (23) in the case $C = 1, L = 24, n = 120$.

(b) η_n^* vs. n.

Fig. 1. Example.

Fig. 1.(a) suggests that $R_{t,120}^{J_{120}}$ is increasing in t. This observation is consistent with the intuition that, the more uncertainty future demands have, the more difficult it is for online algorithms to make decisions.

Next, we compute η_n^* for different n's ranging from 1 to 144. Fig. 1.(b) shows how η_n^* varies with respect to n. Based on the values of η_n^*'s, we finally obtain $\eta^* = 1.8185$.

B. Impact of Reservation on η^*

In this section, we show how the optimal competitive ratio η^* varies as the key reservation parameters change. Again, we set $T = 144$. There are two parameters related to the reservation, p (or C) and L. We will vary p and L, and characterize their impact on η^*. Such results will help us understand how reservation improve the performance of the online algorithm.

For $L = 0, 36, 72, 108, 144$, we compute η^* for different p's. From Fig. 2, we can see that when $L = 0$, η^* remains at the highest value[5] of 2.39 regardless of the value of p. The reason is that in the case of $L = 0$, the reserved jobs are allowed to reserve upon its arrival, and thus the worst case CR would be the same as if there is no reservation. As this L increases, we know more advance information about the future. Therefore, as L increases, η^* will decrease. As for p, it is the fraction of reserved demand over the total demand. As p increases, the total demand uncertainty will decrease, and thus the CR η^* will decrease. For example, when $L = 72$ and $p = 0.6$ (i.e., 60% of the total demand is from the jobs that are reserved $\frac{1}{2}$ of the time horizon ahead of their arrivals times), the optimal competitive ratio is reduced to 1.39. In the extreme case where $L = 144$ and $p = 1$, i.e., all the future knowledge are known exactly at the beginning, the CR becomes $\eta^* = 1$. An interesting observation is that, for a moderate value of L, e.g., $L = 36$, we already attain most part of the benefits from reservation. Hence, in practice, the aggregator can focus on price incentives for comparable time-intervals of advanced reservation.

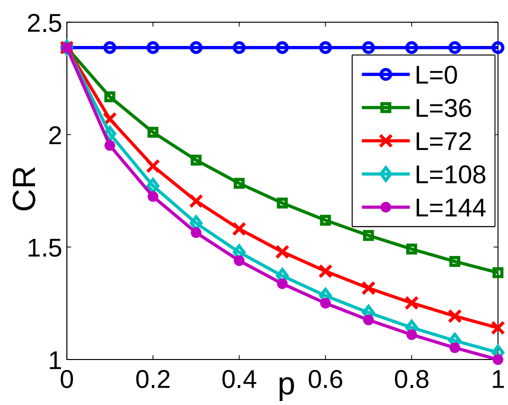


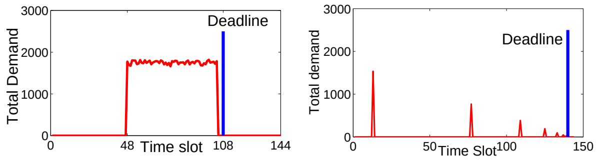
Fig. 2. Impact of Reservation on CR η^*.

VI. Simulation

In this section, we compare the performance of our EPS algorithm with two online algorithms for EV charging. The first algorithm does not coordinate among different EV's. For each EV k, the aggregator simply charge this EV k at a constant rate of $\frac{e_k}{d_k - s_k + 1}$. We call this algorithm the uncoordinated algorithm, which can be view as the scenario when each user manages its own EV charging. The second algorithm is the myopic algorithm we discussed in Section III-B.

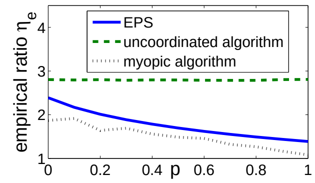
We generate the arrival pattern in the following way. We assume that there are EV charging jobs arriving continuously from time t_1, but all of these jobs must leave before time t_2. Specifically, we simulate 200000 jobs. The arrival time of each job is uniformly distributed in $[t_1, t_2 - \sigma]$, the deadline of each job is t_2, and the EV-charging demand is a random variable uniformly distributed in $[0, 1]$. Such an arrival pattern may arise for a parking lot near an office complex, where most offices open at 8am, and close at 6pm. Thus, we set $t_1 = 48$, and $t_2 = 108$. An example of this arrival pattern is plotted in Fig. 3. (a).

[5]Note that here we have $\eta^* < e$ because the time horizon $T = 144$ is finite. If $T \to \infty$, we will have $\eta^* \to e$ [14].

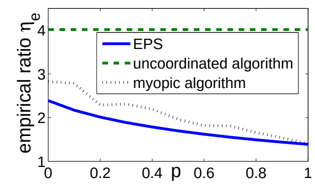


(a) Total demand vs. time in the first arrival pattern ($t_1 = 48, t_2 = 108, \sigma = 3$). (b) Total demand vs. time in the second arrival pattern.

Fig. 3. Two arrival patterns.

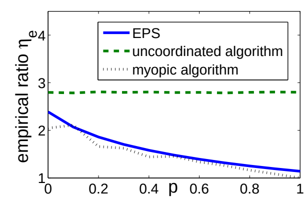


(a) η_e vs. p for the first arrival pattern.

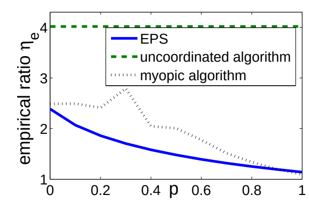


(b) η_e vs. p for the second arrival pattern.

Fig. 4. $L = 36$.

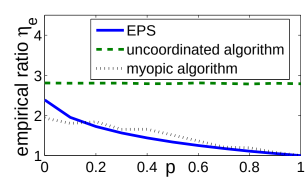


(a) η_e vs. p for the first arrival pattern.

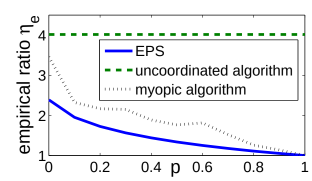


(b) η_e vs. p for the second arrival pattern.

Fig. 5. $L = 72$.



(a) η_e vs. p for the first arrival pattern.



(b) η_e vs. p for the second arrival pattern.

Fig. 6. $L = 144$.

We would like to see the benefit of reservation on reducing the peak-load. We assume that the reserved demand $r_{i,j}$ is at lease p fraction of the total demand $r_{i,j} + a_{i,j}$ (Fig. 3.(a)). Specifically, we assume that $\frac{r_{i,j}}{r_{i,j}+a_{i,j}}$ is a random variable uniformly distributed in $[p, 1]$. Further, we assume that the reserved EV-charging jobs are reserved exactly L time slots ahead. Given an online algorithm, we define its empirical ratio η_e as the ratio between the peak consumption under this algorithm and the optimal peak consumption under the offline optimal algorithm. Then, we vary L, and compute the empirical ratio η_e, for different p's under all the three algorithms. From Fig. 4-6.(a), we can see that both the EPS algorithm and the myopic algorithm perform much better than the uncoordinated algorithm, while the EPS algorithm and the myopic algorithm have comparable performance. Therefore, by coordinating among different EV's, we can significantly reduce the peak consumption.

To better understand the performance of the EPS algorithm and the myopic algorithm, we generate another arrival pattern (Fig. 3.(b)). The second arrival pattern is similar to the arrival pattern we studied in Section III-B. In Section III-B, we have shown that if we have infinite batches of jobs, the myopic algorithm does not have a finite CR. However, the second arrival pattern here only has finite batches of jobs. Thus, we would expect that the gap between the empirical ratio of the myopic algorithm and the optimal CR will not be as dramatic. Nevertheless, from Fig. 4-6.(b), we can see that the EPS algorithm performs much better than the myopic algorithm. Further, we note that the empirical rate η_e's are the same for the two arrival patterns under the EPS algorithm, while the empirical rate η_e's are dramatically different across the two arrival patterns under the myopic algorithm. Such an observation indicates that the EPS algorithm is more robust in reducing the peak than the myopic algorithm. One may argue that the arrival pattern in Fig. 3.(b) may occur rarely in practice. However, from the grid stability point of view, it is indeed rare events that lead to costly failures [16]. Hence, the ability of the EPS algorithm to gracefully handle the peak even in the worst case is highly desirable in practice.

VII. Proof

A. Proof of Lemma 1

Proof: Consider the job arrival pattern depicted in Fig. 7. All jobs have the same deadline T. Without of loss of generality, assume that $T = 2^n$. The first batch of jobs arrives at time 0, and has a total demand of T. The second batch of jobs arrives at time $\frac{T}{2}$, and has a total demand of $\frac{T}{2}$. The n-th batch of jobs arrives at time $T - \frac{T}{2^{n-1}}$, and has a total demand of $\frac{T}{2^{n-1}}$. It is easy to see that the peak rate in the

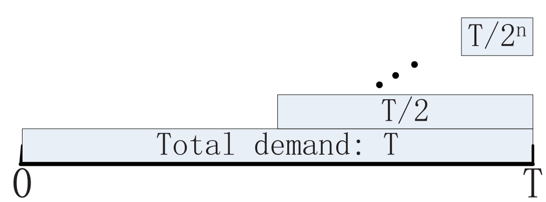
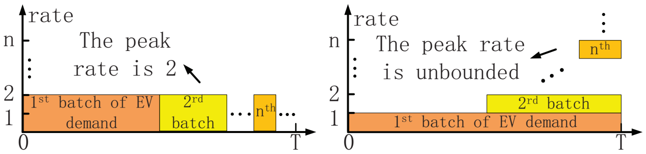


Fig. 7. EV-demand arrival pattern.

optimal offline YDS solution is 2 (Fig. 8.(a)). With a serving rate of 2, every batch of jobs can be finished right before the arrival of the next batch of jobs. However, the myopic online



(a) The offline optimal algorithm. (b) The myopic online algorithm.

Fig. 8. The service profiles of two algorithms.

algorithm will behave quite differently (Fig. 8.(b)). In the time period $[0, \frac{T}{2}]$, the myopic algorithm only knows the first batch of jobs. Hence, the serving rate is 1. At time $\frac{T}{2}$, only half of the demand of batch 1 is served. Then, the second batch of jobs arrives, which adds to the remaining half from the first batch of demand. The total outstanding demand is T, and it needs to be served in the interval $[\frac{T}{2}, T]$. Hence, the service rate of the myopic algorithm increases to 2 in the interval $[\frac{T}{2}, \frac{3T}{4}]$. In a similar manner, we can see that in the interval $[T - \frac{T}{2^{n-1}}, T - \frac{T}{2^n}]$, the service rate of the myopic algorithm will be n (Fig. 8.(b)). As n goes to infinity, the peak-serving rate of the myopic algorithm is unbounded. Thus, the CR of the myopic online algorithm can be arbitrarily large as $T \to \infty$. ∎

B. *Proof of Lemma 2*

Proof: Consider the same job arrival pattern shown in Fig. 7. We assume that for each batch of EV demand, exact p_l fraction of the demand is reserved at or before time 0 (the constraint $r_k \le s_k - L$ is thus met for any L), and the rest is walk-in demand. We use x_k to denote the serving rate during the time interval $[T - \frac{T}{2^{k-1}}, T - \frac{T}{2^k}]$ under the myopic online algorithm. In the time period $[0, \frac{T}{2}]$, the myopic algorithm knows the total reserved demand, which is $2p_l T$, and the demand of the first batch of walk-in jobs, which is $(1-p_l)T$. Then, based on (4), it is easy to check that

$$x_1 = \frac{2p_l T + (1-p_l)T}{T} = 1 + p_l. \tag{24}$$

Further, we can derive an induction formula for the sequence $\{x_k\}$. In the time interval $[T - \frac{T}{2^{n-1}}, T - \frac{T}{2^n}]$. The myopic algorithm knows the total reserved demand, which is $2p_l T$, and the demand of the first n batches of walk-in jobs, which is $(1-p_l)T\sum_{s=0}^{k-1} 2^{-s}$. Among these known demand, $T\sum_{s=1}^{k-1} 2^{-s}x_s$ amount of it has been served. Then, we can show that

$$x_k = \frac{2p_l T + (1-p_l)T(\sum_{s=0}^{k-1} 2^{-s}) - T\sum_{s=1}^{k-1} 2^{-s}x_s}{2^{-(k-1)}T}. \tag{25}$$

Solving the above recursive formula gives $x_n = 2p_l + n(1-p_l)$. Therefore, as long as $p_l < 1$, the peak-serving rate is unbounded as $n \to \infty$. Thus, the CR of the myopic online algorithm can be arbitrarily large as $T \to \infty$. ∎

C. *Proof of Lemma 3*

Proof: We prove by contradiction. Suppose that there exist t_0, such that $E_{t_0}^{J_n}(\pi_n) > \eta_n E_{\text{pe}}^{J_n}(t_0)$.

Consider the job sequence J'_n. J'_n only contains the same set of walk-in jobs in J_n arriving before or at time t_0, and the same set of reserved jobs reserved before or at time t_0. Thus, if we apply the same online algorithm π_n to J'_n, it must produce the same decisions at time $t \le t_0$, i.e.,

$$E_t^{J'_n}(\pi_n) = E_t^{J_n}(\pi_n), t = 1, 2, ..., t_0.$$

On the other hand, if the job sequence is indeed J'_n, we actually know all the job arrivals at time t_0. Therefore, we can compute the offline optimal peak of J'_n using the YDS algorithm. Note that all jobs in J'_n have the same deadline n. Therefore, when computing the offline optimal peak $E^*_{J'_n,\text{off}}$, we only need to focus on the intervals with right end-point being n, i.e., $I = [j, n], j = 1, 2, ..., n$. Furthermore, J'_n does not have any demand after time $h(t) = \min\{t + L, n\}$. Hence, we can further restrict j to be from 1 to $h(t)$. Based on the above discussion, we have

$$E^*_{J'_n,\text{off}} = \max_{j=1,...,h(t)} \{g_{J'_n}([j, n])\}.$$

Note that in J'_n, the total demand of jobs with arriving time $s \ge j$ and departure time $d \le n$ is equal to $\sum_{i=j}^{t} A_{i,n}^{J_n} + \sum_{i=j}^{h(t)} R_{i,n}^{J_n}$. Therefore,

$$\begin{aligned} E^*_{J'_n,\text{off}} &= \max_{j=1,...,h(t)} \left\{ \frac{\sum_{i=j}^{t} A_{i,n}^{J_n} + \sum_{i=j}^{h(t)} R_{i,n}^{J_n}}{n-j+1} \right\} \\ &= E_{\text{pe}}^{J_n}(t). \end{aligned}$$

Then,

$$E_{t_0}^{J'_n}(\pi_n) = E_{t_0}^{J_n}(\pi_n) > \eta_n E_{\text{pe}}^{J_n}(t_0) = \eta_n E^*_{J'_n,\text{off}},$$

which contradicts to the fact that π_n has CR η_n. ∎

D. *Proof of Lemma 6*

Proof: The necessity is obvious. We focus on the sufficiency in the following proof.

Suppose that $A_{t_1,t_2}^J + R_{t_1,t_2}^J \le \sum_{t=t_1}^{t_2} E_t^J$ for any $1 \le t_1 \le t_2 \le T$. We will show that E_J is feasible based on the earliest-deadline-first policy.

We prove by contradiction. If E_J is not feasible, then there must exist at least one job request k that misses its deadline. Without loss of generality, we assume that this job's deadline is at time slot d. We say a time slot $t < d$ is *good*, if and only if all the energy E_t is used to serve job requests with deadline no later than d. It is easy to see that time slot d is always good.

If all the time slots $t = 1, 2, ..., d-1$ are good, then there is no energy wasted during the first d time slots, and all of the energy is used to serve jobs with deadlines no later than d. Note that $A_{1,d}^J + R_{1,d}^J \le \sum_{t=1}^{d} E_t^J$. Then, job k must be completed before time d, which contradicts to our assumption.

If there exists some time slots $t < d$ that is not good, let $t_b = \max\{t < d | t \text{ is not good}\}$. Then, in time slots $t = t_b + 1, ..., d$, no energy is wasted, and only job requests with deadline smaller or equal to d are served. Furthermore, all jobs with arrival time no later than t_b and deadline no later than d must have been completed before or at time slot t_b. (Otherwise, t_b would have been good because the energy E_t^J could have been used to serve these jobs according to the earliest-deadline-first policy.) It also implies that job k cannot arrive before t_b (otherwise it would have been completed). Further, in time slots $t = t_b + 1, ..., d$, all the energy must be used to first serve requests with arrival time later than t_b and deadline smaller or equal to d. Note that $A^J_{t_b+1,d} + R^J_{t_b+1,d} \leq \sum_{t=t_b+1}^{d} E_t^J$. Then, job k must be finished before time d, which contradicts to our assumption. ■

VIII. Conclusion

We study online peak-minimizing algorithms for an aggregator, which manages a large set of EV charging jobs with deadlines. Existing algorithms either require precise future knowledge or do not make use of any future knowledge. In contrast, we focus on a more practical scenario where some limited future knowledge can be obtained. Specifically, we consider the scenario where such limited future knowledge is revealed by job reservation. We then propose a general and systematic approach to design competitive online algorithms. Our proposed algorithm, called EPS, can attain the optimal competitive ratio under an arbitrary set of reservation parameters. We also characterize the benefit of reservation in reducing the peak consumption. Compared to the previous online algorithms (e.g. BKP [7]) that do not make use of any future knowledge, the proposed EPS algorithm can significantly reduce the competitive ratio. Finally, Simulation results demonstrate that the EPS algorithm is indeed very robust and effective in reducing the peak.

References

[1] M. Valentine-Urbschat and W. Bernhart, "Powertrain 2020 - The future drives electric," Roland Berger Strategic Consultants Rep., Munich, Germany, pp. 1-95.

[2] InternatIonal energy agency, "Impact of Smart Grid Technologies on Peak Load to 2050", 2011.

[3] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic Provisioning in Next-Generation Data Centers with On-Site Power Production," *in Proceedings of the fourth International Conference on Future Energy Systems (ACM e-Energy)*, Berkeley, CA, US, May 22-24, 2013.

[4] http://www.fcgov.com/utilities/business/rates/electric/coincident-peak.

[5] F. Yao, A. Demers, and S. Shenker, "A Scheduling Model for Reduced CPU Energy," *Proc. of the IEEE syposium on Foundations of Computer Science* Los Alamitos, CA, Oct. 1995.

[6] L. Gan, U. Topcu, and S. Low, "Optimal Decentralized Protocol for Electric Vehicle Charging," *IEEE Trans. on Power Systems*, 2011.

[7] N. Bansal, T. Kimbrel, and K. Pruhs, "Speed scaling to manage energy and temperature," *Journal of the ACM*, vol. 54, no. 1, March 2007.

[8] Z. Liu, A. Wierman, Y. Chen, and B. Razon, "Data Center Demand Response: Avoiding the Coincident Peak via Workload Shifting and Local Generation," *Proc. of ACM Sigmetrics*, Pittsburgh, PA, Jun. 2013.

[9] A. Bar-Noy, M. P. Johnson, and O. Liu, "Peak shaving through resource buffering," *Approximation and Online Algorithms*, Springer Berlin Heidelberg, pp. 147-159, 2009.

[10] M. Neely, A. Tehrani, and A. Dimakis, "Efficient algorithms for renewable energy allocation to delay tolerant consumers," *Smart Grid Communications (SmartGridComm)*, 2010 First IEEE International Conference on, pp. 549-554, oct. 2010.

[11] S. Chen, N. B. Shroff and P. Sinha, "Heterogeneous Delay Tolerant Task Scheduling and Energy Management in the Smart Grid with Renewable Energy," *IEEE Journal on Selected Areas in Communications*, 2013.

[12] S. Chen, T. He, and L. Tong, "Optimal deadline scheduling with commitment," *2011 Allerton Conference on Communication, Control and Computing*, Oct. 2011.

[13] L. Chen, N. Li, L. Jiang and S. H. Low, "Optimal demand response: problem formulation and deterministic case," *Control and Optimization Theory for Electric Smart Grids*, Aranya Chakrabortty and Marija Ilic, Editors, Springer 2012

[14] B. Hunsaker, A. J. Kleywegt, M. W. P. Savelsbergh, and C. A. Tovey, "Optimal online algorithms for minimax resource scheduling," *SIAM Journal on Discrete Mathematics*, vol. 16, no. 4, pp. 555-590, 2003.

[15] http://cvxr.com/cvx/.

[16] P. Hines, K. Balasubramaniam, and E. C. Sanchez, "Cascading failures in power grids," *IEEE Potentials*, vol. 28, no. 5, pp. 24-30, 2009.