Online Scheduling of Deferrable Jobs: Value of Partial Future Information for Peak Minimization

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*—We study the fundamental gain from exploiting partial future knowledge for scheduling deferrable jobs to minimize peak demand levels. Each deferrable job requests a certain amount of resources that need to be allocated before a deadline. Given an arrival sequence of deferrable jobs, the objective is to minimize the peak demand at any time by scheduling the service of the deferrable jobs. A key challenge of this problem is that there exists significant uncertainty in future arrivals of deferrable 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 a limited amount of partial future information is readily available. Based on the framework of competitive online algorithms, we propose to use the optimal competitive ratio as a metric for quantifying the value of partial future information. We develop a powerful computation framework for computing the optimal competitive ratio given various types of partial future information and for finding the corresponding optimal online scheduling algorithm. This computation framework can be used to quantify the value of different types of partial future information even before acquiring the information. According to our numerical results, we can achieve close-to-1 competitive ratios even with very coarse-grained partial future information. Specifically, the job-reservation information could reduce the achievable competitive ratio by more than 50%, and the minimum-job-duration information could further reduce the competitive ratio by 10%.**

I. Introduction

Peak demand is a key concern for many service systems that aim to provide stringent performance guarantees. For example, for power system, the demand and supply must be balanced at all times[2]. Thus, the peak demand determines the amount of generation capacity that the operator must procure in advance. Similarly, for data center and cloud computing, peak demand determines the network bandwidth that need to be provisioned to end users [3]. Due to this reason, in these systems it is common to price the service based on the peak demand. An example for this type of peak-based pricing for power systems is provided in [4]. Similar peak-based (e.g., 95-percentile peak) schemes have also been used by ISPs for pricing network-bandwidth usage [5].

Intelligently scheduling deferrable jobs can be a highly-effective way to reduce peak demand. Note that such deferrable jobs are prevalent in the above systems, e.g., electrical vehicle charging jobs for smart grid, large file transfer in data center, low priority jobs for cloud computing. The flexibility of these deferrable jobs can be used to shave the "peak" and fill the "valley", and thus reduce the peak demand. However, even though these jobs are deferrable, they may still have their own deadlines. Further, there may be significant uncertainty in the arrivals of future deferrable and non-deferrable load. If not scheduled properly, future peak may occur, which further stress the entire system (see Section III-B for examples). Hence, there is a pressing need to develop effective scheduling algorithms under deadline constraints and future uncertainty of deferrable jobs.

An earlier version of this work has been presented in 51st Annual Allerton Conference on Communication, Control, and Computing [1].

A. Electrical Vehicle (EV) Charging as a Use Case

In order to make our presentation more concrete, throughout the paper we will use EV charging as a concrete example for the deferrable-job scheduling problem. Nonetheless, our proposed method will also work for other types of deferrable jobs. Consider 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 [6][7] 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 type 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 $c_1 \sum_i E_i + c_2 \max_i\{E_i\}$ [4]. In typical schemes (e.g., the National Grid [4]), the average charge for peak usage c_2 (8.32$/kW-month) is over 100 times more than the unit charge for total usage c_1 (0.07$/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.

B. Existing approaches and challenges

A main challenge for peak-minimizing EV charging is the uncertainty of future arrivals and departures of EV charging requests. (Again, the same challenge also exists for other types of deferrable jobs under uncertainty, e.g., data center [8].) Clearly, if all future EV charging jobs are known in advance, one can then readily compute the optimal charging schedule that perfectly minimizes the peak [9][10]. Unfortunately, knowing the entire future demand is usually unrealistic. The other extreme is where no information about future EV charging requests is available to the aggregator. Here, because the aggregator cannot go back in time (i.e., the aggregator cannot adjust a "sub-optimal" charging decision that was made in the past), the performance of any online scheduling algorithm would likely be quite poor. For example, in a peak-minimizing problem closely related to ours [11], the peak consumption of the best online algorithm could be, in the worst case, $e = 2.718$ times of that of the offline-optimal solution. This constant factor is referred to as the *competitive ratio* of the online algorithm. Between these two extremes, we enter the highly practical regime where the aggregator may have some *partial* information of the future, but not *all*. Intuitively, the more future information is made available to the aggregator, the "easier" it will be to minimize the peak. Thus, it is to the aggregator's benefit to acquire and utilize various types of partial future information. For example, the aggregator may ask EV owners to make reservations for their EV charging jobs L slots in advance, and may even offer monetary incentives to encourage more customers to make reservations ahead of time. However, the difficulty in managing and exploiting these types of partial future information is that they are often quite "coarse." For example, with a given level of monetary incentive, the aggregator may be able to know that at least a certain fraction p of the customers are willing to make reservations in advance. Still, it may not know exactly how many reserved or "walk-in"[1] EV-charging jobs will arrive at each time-interval. Thus, the open question that we wish to answer is then: how can such coarse and partial future information be utilized in aggregator's decision making, and how to quantify the value of such partial information in advance, even before the actual EV charging jobs arrive?

C. Our contribution

We propose to build upon the framework of competitive online algorithms to develop a systematic approach for utilizing and evaluating partial future information. Intuitively, as more partial future information is available, we would expect that the achievable competitive ratio of online algorithms will become closer to 1. If we assume that future realization of EV charging jobs will be the same regardless of what type of partial information is provided to the aggregator ahead of time, then the offline minimum peak (assuming that the entire future is known) is thus fixed. Hence, a smaller competitive ratio is directly related to a lower worst-case peak faced by the online algorithm. Thus, in this paper we propose to quantify the value of a particular piece of partial future information by the optimal competitive ratio that can be achieved by online algorithms operating with that partial information. Note that existing competitive online algorithms (such as [11]) usually assume no future information. Thus, our first contribution is to develop a powerful computational framework both to compute the optimal competitive ratio and to find the corresponding optimal online algorithms under various types of partial future information. In other words, instead of providing just one online algorithm and one competitive ratio, we provide a computational framework that maps a given set of partial future information to the corresponding optimal competitive ratio as well as the optimal online algorithm. We demonstrate that this computational framework can be applied to several types of partial information, including reservation, minimum job-duration, and maximum job-duration (detailed models will be provided in Section II and Section V). In each case, the optimal competitive ratio can be computed as a function of the various parameters of the partial future information.

Based on this computational framework, we then conduct numerical studies to evaluate the relative value of the different types of partial future information. Our findings indicate that the partial future information revealed by job reservation is of the highest value. Even a moderate level of job reservation can reduce the optimal competitive ratio significantly. For example, when 60% of the jobs are reserved $1/2$ of the total time horizon ahead of time, the optimal competitive ratio is reduced to 1.39 (corresponding to 49% reduction from the competitive ratio of $e = 2.718$ for the case of no reservation [11]). The minimum job-duration constraint is comparatively less useful, but still can further reduce the optimal competitive ratio by as much as 10%. In contrast, we find the maximum job-duration constraint less effective in reducing the competitive ratio. We believe that these quantitative knowledge on the value of the different types of partial information can be extremely valuable for aggregators to design their prospective incentive mechanisms for managing peak-minimizing EV charging.

D. Related work

In the literature, there are two main approaches to model the uncertainty that is associated with partial future knowledge. If the future demand is uncertain but its distribution is known, one can potentially formulate a stochastic control problem, e.g., as a Markov Decision Program (MDP) [12]. However, as we illustrated earlier, the partial future information that we wish to deal with may be so "coarse" that even this distribution is difficult to specify. If the distribution is incorrect, the

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

performance guarantee from the MDP solution will likely be unreliable. If the distribution is uncertain but is known to vary within a certain range, one can adopt the approach of Robust MDP [13]. However, both MDP and Robust MDP also suffer prohibitively high computational complexity (i.e., the "curse of dimensionality") when the problem size is large.

Another approach to account for partial information is to model the uncertainty by set-based constraints, as commonly used in robust optimization [14][15]. The goal is then to optimize the worst-case performance for any future realization that lies in the uncertainty set. Not only are such set constraints easier to describe, the corresponding decision problems are usually also of lower complexity compared to MDP. However, most studies of robust optimization do not deal with multi-stage sequential decisions. Even when they do, their focus on the *absolute* worst-case cost can often be too conservative in practice. In contrast, in competitive online algorithms [16], one focuses on the *relative* competitive ratio between the online cost and the offline optimal cost, which requires very different solution approaches. For EV charging, competitive online algorithms have been developed for minimizing the total cost (instead of minimizing the peak demand) in [17]. Still, as we discussed earlier, these studies of online algorithms do not exploit any partial future information. As a result, the competitive ratio may be high. The results of this paper are also related to our more recent work [18][19], where we study competitive online algorithms that utilize imprecise prediction. While prediction is also a form of partial future information, it is significantly different from the types of partial future information that we studied in this paper, including job reservation and job-length constraints. For example, with prediction, one is given upper and lower bounds of the demand in each time-slot. In contrast, with reservation, one only knows how large a relative fraction of the demand will be reserved a certain time-interval ahead of time. In this sense, reservation (as well as job-length constraints) can be viewed as an even coarser form of partial future information. This difference also leads to significant difference in the complexity of finding the optimal competitive ratio. See Remark 2 for further discussions.

II. System Model

We study deferrable load control in the context of an aggregator managing the EV-charging jobs[2] of its customers (although our model may also be used for other types of deferrable jobs, such as low priority jobs in cloud). 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 new and is used to model the first type of partial future information, i.e., reservation. Specifically, we expect that some customers may be able to reserve their EV-charging jobs in advance, in which case $v_k < s_k$. Otherwise, if $v_k = s_k$, we refer to the job as a "walk-in" job. In practice, we expect that the aggregator will offer price incentives to 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. (Other types of partial future information, i.e., through job-duration constraints, will be introduced in Section V.)

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 $r^J_{i,j}$ be the total *reserved* demand with arrival time i and departure time j, and let $R^J_{i,j} = \sum_{j'=i}^{j} r^J_{i,j'}$ be the total reserved demand with arrival time i and departure time no greater than j. Similarly, let $a^J_{i,j}$ be the total *walk-in* demand with arrival time i and departure time j, and let $A^J_{i,j} = \sum_{j'=i}^{j} a^J_{i,j'}$ be the total walk-in demand with arrival time i and departure time no greater than j. According to our reservation model, all $r^J_{i,j}$'s are known at least L time-slots ahead of time i, while $a^J_{i,j}$'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,

$$p_l(R^J_{i,j} + A^J_{i,j}) \leq R^J_{i,j} \leq p_u(R^J_{i,j} + A^J_{i,j}), \tag{1}$$

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 $R^J_{i,j}$ as the mean of the reserved demand, and use $A^J_{i,j}$ 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 exhibit 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

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

[20]. 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 [9].

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

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 can quantify 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.,

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

An feasible online algorithm π is called optimal, if it attains the smallest competitive ratio. Assume that the future realization of EV charging jobs are the same regardless of the different levels of partial future information revealed to the aggregator (e.g., different values of L and p in our reservation model). Then, the offline minimum peak does not change with L and p either. Thus, a smaller value of the optimal competitive ratio under a particular piece of partial information also translates to a lower peak achievable by the online algorithm. Hence, in this paper we propose to quantify the value of different types of partial future information by the corresponding optimal competitive ratios that online algorithms can achieve. Towards this end, we will develop a general computational framework that takes a particular piece of partial future information as input, and produces the optimal competitive ratio and the corresponding optimal online algorithm as output.

III. Difficulty in the Online Control of Deferrable Load

Unfortunately, making competitive online decisions 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. In other words, although EV demand is deferrable, it may also lead to a large peak if not scheduled properly. Therefore, it is important to find better algorithms for online EV-charging.

A. Review of the Offline-Optimal Solution

To start with, we briefly review how to compute the offline-optimal solution (2). 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

$$g_J(I) = \frac{\sum_{i'=i}^{j}(R^J_{i',j} + A^J_{i',j})}{j - i + 1}. \tag{3}$$

Then, the optimal offline value $E^*_{J,\text{off}}$ of the peak is given by the maximum intensity over all possible intervals, i.e.,

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

The above offline optimal peak $E^*_{J,\text{off}}$ can be achieved by the YDS algorithm [9]. The details are available in Appendix A.

B. A Myopic Online Algorithm

Offline optimal algorithms (e.g., 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 expression (4) 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 [10]. 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.*

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 $T \to \infty$.*

The detailed proofs of Lemma 1 and Lemma 2 are in Appendix B and Appendix C. From the above two lemmas, we can see that it is a highly non-trivial task to make online decisions, either with or without partial future information.

In fact, if there is no reservation, an online algorithm called BKP [11] was shown to achieve a CR of e. Further, this CR e is optimal, which can be viewed as a baseline that captures the performance loss when there is no future information at

all. Since e is still a large number, we are interested in how limited future knowledge may help us to significantly improve the competitive ratio. Unfortunately, the techniques for proving the competitive ratio and its optimality in [11] are very specific and seems difficult to account for the partial future information revealed by our reservation model. In the next section, we will develop a very general framework that can both compute the optimal competitive ratio and find the optimal online algorithm under an arbitrary set of reservation parameters.

IV. Optimal Peak-Minimizing Online EV Charing

In this section, we propose a general framework for computing the optimal competitive ratio 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 $p_u \neq 1$, 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 $E_{J_n}(\pi_n) = \{E^{J_n}_1(\pi_n), E^{J_n}_2(\pi_n), ..., E^{J_n}_n(\pi_n)\}$ under an EV-arrival sequence $J_n \in \mathcal{J}_n$ must satisfy*

$$E^{J_n}_t(\pi_n) \le \eta_n E^{J_n}_{pe}(t), t = 1, 2, ..., n$$

where

$$E^{J_n}_{pe}(t) = \max_{j=1,...,h_n(t+L)} \left\{ \frac{\sum_{i=j}^{t} A^{J_n}_{i,n} + \sum_{i=j}^{h_n(t+L)} R^{J_n}_{i,n}}{n-j+1} \right\}, \tag{7}$$

and $h_n(t') = \min\{t', n\}$. (In (7), the subscript "pe" stands for "peak estimation".)

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 $E^{J_n}_{\text{pe}}(t)$ as the estimate of the peak consumption at time t. In fact, if there were no more new jobs after time t, $E^{J_n}_{\text{pe}}(t)$ would have been the offline-optimal peak service rate. If $E^{J_n}_t(\pi_n) > \eta_n E^{J_n}_{\text{pe}}(t)$, 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

$$\sum_{t=1}^{n} E^{J_n}_t(\pi_n) \ge \sum_{t=1}^{n} \left(A^{J_n}_{t,n} + R^{J_n}_{t,n}\right). \tag{8}$$

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

$$\eta_n \ge \frac{\sum_{t=1}^{n} \left(A^{J_n}_{t,n} + R^{J_n}_{t,n}\right)}{\sum_{t=1}^{n} E^{J_n}_{\text{pe}}(t)}.$$

Define the following optimization problem:

$$\sup_{J_n} \frac{\sum_{t=1}^{n} \left(A^{J_n}_{t,n} + R^{J_n}_{t,n}\right)}{\sum_{t=1}^{n} E^{J_n}_{\text{pe}}(t)} \text{ subject to } (6), (7) \tag{9}$$

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^*.*

In general, the optimization problem (9) can be easily converted into a linear programming problem and solved using standard solvers. See Appendix D for more details.

Remark 1. *Our formulation of the lower bound in (9) shares some similarity to the results in [21]. However, [21] does not consider reservation, and there is substantial difficulty in extending the techniques in [21] to the case with reservation. Specifically, a key step in [21] 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 [21]). 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 how to generalize the techniques of [21] 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.*

Remark 2. *Since the conference version of this paper [1] was published, the above methodology has also been extended in our later work in [18][19] to the case where partial future information is revealed by forecasts (instead of reservations). We note, however, a key difference between (9) and the lower bound in [18]: the lower bound in (9) only needs to consider job sequences with a common deadline. In contrast, the lower bound in [18] must consider job sequences with varying deadlines. Thus, the lower bound in (9) incurs much lower complexity. On the other hand, it is less obvious why the lower bound in (9) is tight, which will be the key contribution of the next subsection.*

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.

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 1: EPS algorithm

The following theorem states that the EPS algorithm is a feasible online algorithm with competitive ratio η^*. Thus, the 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 $E^J_t \le \eta^* E^*_{J,off}$;*
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,*

$$\sum_{t=t_1}^{t_2} (A^J_{t,t_2} + R^J_{t,t_2}) \le \sum_{t=t_1}^{t_2} E^J_t. \tag{10}$$

Proof. See Appendix E □

Now, we are ready to explain the intuition behind the second part of Theorem 5. According Lemma 6, we only need to show that (10) always holds if $E^J_t = \eta^* E^J_{\text{pe}}(t)$. First, if all the jobs contained in A^J_{t,t_2} or R^J_{t,t_2} have the common deadline of t_2, then (10) must hold based on the definition (9) of η^*. Second, if some jobs in A^J_{t,t_2} or R^J_{t,t_2} have deadlines smaller than t_2, we can consider an alternate system where these jobs' deadlines are all extended to t_2. It is easy to show that extending the deadlines will only reduce the service rates E^J_t of the EPS algorithm. If (10) was violated for the original system, it would have been also violated in the alternate system with a common deadline. However, according to (9), the condition (10) must hold for the alternate system with a common deadline, which leads to a contradiction. Thus, (10) must hold for the original system. The detailed proof of Theorem 5 is available in Appendix F.

Remark 3. *The above results can be viewed as a superset of the results in [11][21]. Specifically, when there is no reservation ($C = \infty$), the above algorithm reduces to one that is similar to the BKP algorithm [11]. The competitive ratio is also close to e. (It is not exactly e because the time horizon is finite [21].) However, with reservation, the competitive ratio will improve as can be seen soon in Section VI-A.*

Finally, we note that the EPS algorithm is just one of the online algorithms that achieve the optimal competitive ratio. Since the focus of this paper is on using the optimal competitive ratio to quantify the value of partial future information, it suffices to use the EPS algorithm to show that the optimal competitive ratio can be attained. Readers may refer to our companion paper [18] for other ideas to design online algorithms with not only the optimal competitive ratio, but also better average-case performance.

V. Integrating Job-Duration Information

In addition to reservation, in this paper we are also interested in using the computational framework to study the value of other types of partial future information, i.e., their capability in further reducing the competitive ratio. In particular, in

this section, we study the value of additional job-duration constraints. If we dive into the details of the counter example for Lemma 1 and Lemma 2 (see Appendix B and Appendix C), we can see that one difficulty for online decisions is from the possible future jobs that arrive at the end of the time horizon, but with very short duration. These jobs leave very little flexibility for the online algorithm. Thus, if the aggregrator can impose a lower bound on the job duration, it may be at a better position to reduce the competitive ratio. Similarly, an upper bound on the job duration may also help with reducing the competitive ratio. Below, we will demonstrate how the computational framework in Section IV can be easily extended to this scenario.

In the following discussion, we assume that there exist a lower bound α_l and an upper bound α_u such that, for each job $k \in J$, its duration must be within $[\alpha_l, \alpha_u]$, i.e.,

$$\alpha_l \leq d_k - a_k + 1 \leq \alpha_u. \tag{11}$$

Note that the cases with no lower bound or no upper bound can also be captured by setting $\alpha_l = 0$ or $\alpha_u = T$ in (11).

Recall from Section IV-A that, without the job-duration constraints, the optimal competitive ratio was computed based on a set of job sequences J_n. In this section, we need to modify J_n to incorporate the job-duration constraints. Specifically, for any fixed $n = \alpha_l, \alpha_l + 1, ..., T$, we consider the following job sequence $\hat{J}_n$. The arrival time of each job $k \in \hat{J}_n$ satisfies $1 \leq s_k \leq n - \alpha_l + 1$, and the deadline of each job $k \in \hat{J}_n$ is $d_k = \min\{s_k + \alpha_u - 1, n\}$. Note that all jobs in $\hat{J}_n$ expire at or before time slot n. Further, the condition $d_k = \min\{s_k + \alpha_u - 1, n\}$ ensures that the duration of job k is no larger than α_u, and the condition $s_k \leq n - \alpha_l + 1$ ensures that the duration of job k is no smaller than α_l. Finally, 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 $\hat{\mathcal{J}}_n$ be the set of all job sequences $\hat{J}_n$'s with such a form. Compared to the job sequence J_n in Section IV-A, jobs in $\hat{J}_n$ have additional constraints on their arrival times and deadlines due to the constraints on α_l and α_u. However, both J_n and $\hat{J}_n$ share one common feature, i.e., the deadlines of all jobs are as large as possible.

It turns out that, with the job-duration constraints, the optimal competitive ratio can be computed based on $\hat{J}_n$, $n = \alpha_l, \alpha_l + 1, ..., T$. Specifically, we consider an arbitrary online algorithm $\hat{\pi}_n$ with competitive ratio $\hat{\eta}_n$, and apply this algorithm to an EV-arrival sequence $\hat{J}_n \in \hat{\mathcal{J}}_n$. Similar to Lemma 3, we have the following lemma.

Lemma 7. *Given an online algorithm $\hat{\pi}_n$ with CR $\hat{\eta}_n$, its service profile $E_{\hat{J}_n}(\hat{\pi}_n) = \{E_1^{\hat{J}_n}(\hat{\pi}_n), E_2^{\hat{J}_n}(\hat{\pi}_n), ..., E_n^{\hat{J}_n}(\hat{\pi}_n)\}$ under an EV-arrival sequence $\hat{J}_n \in \hat{\mathcal{J}}_n$ must satisfy*

$$E_t^{\hat{J}_n}(\hat{\pi}_n) \leq \hat{\eta}_n E_{pe}^{\hat{J}_n}(t), t = 1, 2, ..., n,$$

where

$$E_{pe}^{\hat{J}_n}(t) = \max_{\substack{j_1 = 1, ..., h_n^{(0)}(t+L) \\ j_2 = h_n^{(1)}(j_1), ..., h_n^{(1)}(t+L)}} \left\{ \frac{\sum_{i=j_1}^{h_n^{(0)}(t)} A_{i,j_2}^{\hat{J}_n} + \sum_{i=j_1}^{h_n^{(0)}(t+L)} R_{i,j_2}^{\hat{J}_n}}{j_2 - j_1 + 1} \right\}, \tag{12}$$

$h_n^{(0)}(t') = \min\{t', n - \alpha_l + 1\}$ *and* $h_n^{(1)}(t') = \min\{t' + \alpha_u - 1, n\}$. *(Note that $A_{i,j_2}^{\hat{J}_n} = R_{i,j_2}^{\hat{J}_n} = 0$ if $i > j_2$.)*

Note that the algorithm $\hat{\pi}_n$ must also finish all jobs before their corresponding deadlines. With Lemma 7 in mind, we can easily write down the following inequality:

$$\hat{\eta}_n \sum_{t=1}^{n} E_{\text{pe}}^{\hat{J}_n}(t) \geq \sum_{t=1}^{n} E_t^{\hat{J}_n}(\pi_n) \geq \sum_{t=1}^{n} \left(A_{t,n}^{\hat{J}_n} + R_{t,n}^{\hat{J}_n}\right). \tag{13}$$

Define the following optimization problem:

$$\begin{aligned} \sup_{\hat{J}_n} \quad & \frac{\sum_{t=1}^{n} \left(A_{t,n}^{\hat{J}_n} + R_{t,n}^{\hat{J}_n}\right)}{\sum_{t=1}^{n} E_{\text{pe}}^{\hat{J}_n}(t)} \\ \text{subject to} \quad & (6), (12) \end{aligned} \tag{14}$$

Let $\hat{\eta}_n^*$ be the optimal solution to the optimization problem (14), and let

$$\hat{\eta}^* = \min_{n=\alpha_l}^{T} \{\eta_n^*\}.$$

We can then show that $\hat{\eta}^*$ is the optimal competitive ratio using the same technique in Section IV-A.

Specifically, this $\hat{\eta}^*$ is achievable by the EPS algorithm proposed in Section IV-B (we only need to replace η^* by $\hat{\eta}^*$ in the second step of Algorithm 1). The optimality is guaranteed by the following theorem:

Theorem 8. *Given any job sequence J satisfying the job-duration constraint (11), the EPS algorithm (with scaling factor of $\hat{\eta}^*$) satisfies the following two requirements:*

1) *($\hat{\eta}^*$ optimality) at each time slot t, the service rate E_t^J satisfies $E_t^J \leq \hat{\eta}^* E_{J,off}^*$;*
2) *(feasibility) all jobs can be completed before their deadlines.*

The intuition behind Theorem 8 is essentially the same as Theorem 5. According Lemma 6, we only need to show that (10) always holds if $E_t^J = \hat{\eta}^* E_{\text{pe}}^J(t)$. First, if all the jobs contained in A_{t,t_2}^J or R_{t,t_2}^J have the largest possible deadlines (i.e., this job sequence is of the form $\hat{\mathcal{J}}_n$), then (10) must hold based on the definition (14) of $\hat{\eta}^*$. Second, if some jobs in A_{t,t_2}^J or R_{t,t_2}^J have deadlines smaller than their largest possible deadlines, we can consider an alternate system where these jobs' deadlines are all extended to their largest possible deadlines. It is easy to show that extending the deadlines will only reduce the service rates E_t^J of the EPS algorithm. If (10) was violated for the original system, it would have been also violated in the alternate system with a common deadline. However, according to (14), the condition (10) must hold for the alternate system, which leads to a contradiction. Thus, (10) must hold for the original system.

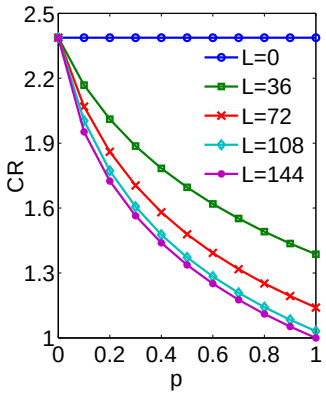
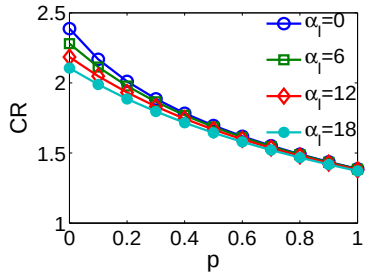
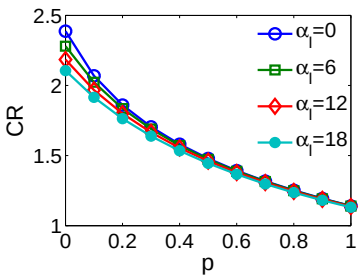


Fig. 1. Impact of Reservation on the optimal CR η^*.

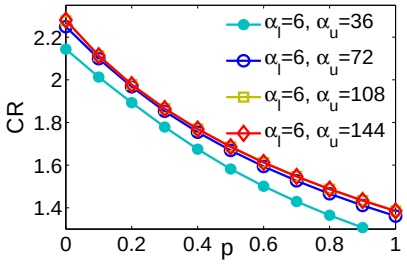


(a) $L = 36$.

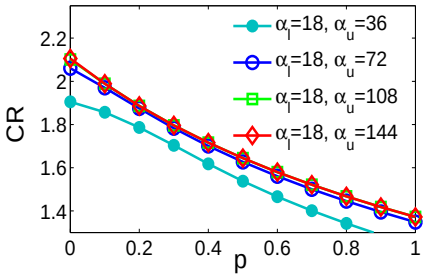


(b) $L = 72$.

Fig. 2. Impact of the minimum job-duration constraint on the optimal CR.



(a) $\alpha_l = 6$.



(b) $\alpha_l = 18$.

Fig. 3. Impact of maximum job-duration Constraint on the optimal CR $\hat{\eta}^*$.

VI. Simulation

We have proposed a general methodology to compute the optimal competitive ratio under different types of partial future knowledge. Such optimal competitive ratio can then be used to assess the value of partial future knowledge. Note that, without any future information, the optimal competitive ratio for peak minimization is $e = 2.718$. In this section, we will conduct numerical simulations to study how the optimal competitive ratio can be reduced under different types and different parameter settings of the partial future knowledge. In this section, we assume that the billing period is a day, and the duration of each time slot is 10 minutes. Thus, the entire time horizon $T = 144$.

A. Impact of Job Reservation

We first study how valuable is the partial future knowledge revealed by job-reservation. Specifically, for different reservation advance $L = 0, 36, 72, 108, 144$, we compute the optimal competitive ratio η^* for different p's. From Fig. 1, we can see that when $L = 0$, η^* remains at the highest value[3] 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 $\eta^* = 1$.

[3]Note that here we have $\eta^* < e$ because the time horizon $T = 144$ is finite. If $T \to \infty$, we will have $\eta^* \to e$ [21].

B. Impact of Minimum Job-Duration Constraint

We then study how the minimum job-duration constraint may further improve the optimal competitive ratio. Specifically, for different reservation settings (different p's and L's), we compute the optimal competitive ratio $\hat{\eta}^*$ for different α_l's (here, we simply set $\alpha_u = 144$). From Fig. 2 (a) and (b), we can see that, as the minimum job duration α_l increases, the optimal competitive ratio decreases. For example, when $p = 0$ (no reservation), the optimal competitive ratio reduces from 2.39 with $\alpha_l = 0$ to 2.11 with $\alpha_l = 18$ (i.e., 3 hours), which corresponds to approximately 10% improvement. However, at higher values of p (with reservation), the gain due to α_l becomes smaller. In other words, the future knowledge revealed by the minimum job-duration constraint seems to have some overlap with the future information revealed by reservation (i.e., p and L). To understand this behavior, note that the minimum job-duration constraint implies that there will be no EV charging jobs arriving after time $T - \alpha_l + 1$, which then implies that there is no reserved job with arrival time larger than $T - \alpha_l + 1$. This observation suggests that the reservation-based model shares some information with the minimum job-duration constraint. Thus, the larger are the values of p and L, the more future knowledge is revealed

by reservation, then the additional knowledge revealed by the minimum job-duration constraint becomes less critical.

C. Impact of Maximum Job-Duration Constraint

We next study the impact of the maximum job-duration constraint on the optimal competitive ratio. In the simulation, we fix the reservation advance $L = 36$, and pick two different values of α_l, i.e., 6 and 18. We vary the maximum job length α_u from 36 to 144, and compute the optimal competitive ratio $\hat{\eta}^*$, with respect to different values of p. From Fig. 3 (a) and (b), we can see that as the maximum job length α_u decreases, i.e., the job-duration constraint becomes more stringent, the optimal competitive ratio will decrease. However, such improvement is minimal for $\alpha_n \geq 72$. Even when $\alpha_u = 72$, i.e., the duration of an EV charging job must be no longer than 12 hours, the improvement is less than 2%. Readers may notice that the improvement can be as large as 10% when $\alpha_u = 36$, i.e., the duration of an EV charging job must be no longer than 6 hours. However, a maximum limit of 6 hours may be too short in practice (e.g., a car may have to be left in the garage for more than 8 hours at work). Based on these observations, we conclude that setting maximum job-duration constraints is not very effective in reducing the competitive ratio.

D. Summary of Numerical Results

From Fig. 1, Fig. 2 and Fig. 3, we conclude that the future information revealed by job reservation is of the highest value, because the optimal competitive ratio reduces significantly as more EV jobs are reserved in advance. The minimum job-duration constraint can further reduce the optimal competitive ratio by as much as 10%. Even though the improvement is not as significant as that of job-reservation, such improvement can be still valuable because even a 1%-reduction could save $0.01 \times 20\text{MW} \times 9\$/\text{kW} \times 12 = 21600\$$ per year for campus-level aggregators [22] with peak energy on the order of 20MW. We also note that the minimum job-duration constraint has less and less impact on the optimal competitive ratio as more jobs reserve in advance. Compared to job reservation and the minimum job-duration constraint, the maximum job-duration constraint is less effective.

We believe that these results provide useful guidance to aggregators for assessing the value of various types of partial future information, which will also help them design suitable price incentives for acquiring partial future knowledge.

VII. Conclusion

We study online deferrable load scheduling in the context of peak-minimizing EV charging. Existing algorithms either require precise future knowledge or do not make use of any future knowledge at all. In contrast, we focus on a more practical scenario where limited and partial future knowledge can be obtained, and take the initiative to quantify the value of such partial future information. Specifically, we consider scenarios where such limited future knowledge is revealed by job reservation and/or the job-duration constraints. We propose a general and systematic approach to design competitive online algorithms with the optimal competitive ratios (CR). Such optimal CRs can then be used to evaluate the inherent benefit of the corresponding type of partial future information. Compared to the optimal CR e (achieved by the BKP algorithm [11]) for the scenario where no future information is available, our numerical results demonstrate that limited future information (especially those from reservation) is indeed very effective in reducing the optimal CR.

References

[1] S. Zhao, X. Lin, and M. Chen, "Peak-minimizing online EV charging," in *51st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, US, Oct. 2013.

[2] U.S. Energy Information Administration, "Energy Explained," https://www.eia.gov/energyexplained/.

[3] A. Kumar, *et. al.*, "Bwe: Flexible, hierarchical bandwidth allocation for wan distributed computing," in *Proc. of ACM SIGCOMM*, Aug. 2015.

[4] NationalGrid, "Understanding Electric Demand," https://www9.nationalgridus.com/niagaramohawk/non_html/eff_elec-demand.pdf.

[5] A. Odlyzko, "Internet pricing and the history of communications," *Computer Networks*, vol. 36, no. 5, pp. 493–517, August 2001.

[6] M. Neely, A. Tehrani, and A. Dimakis, "Efficient algorithms for renewable energy allocation to delay tolerant consumers," in *Smart Grid Communications (SmartGridComm)*, oct. 2010, pp. 549–554.

[7] 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 (JSAC)*, 2013.

[8] W. Z. D. Paul and S. K. Bose, "Demand response in data centers through energy-efficient scheduling and simple incentivization," *IEEE Systems Journal*, vol. 11, no. 2, pp. 613–624, June 2017.

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

[10] L. Gan, U. Topcu, and S. Low, "Optimal Decentralized Protocol for Electric Vehicle Charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, May 2013.

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

[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[13] A. Nilim and L. El Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Oper. Res.*, vol. 53, no. 5, pp. 780–798, September-October 2005.

[14] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of Robust Optimization," *SIAM review*, vol. 53, no. 3, 2011.

[15] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Trans. on Power Systems*, vol. 28, no. 1, 2013.

[16] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. cambridge university press, 2005.

[17] W. Tang, S. Bi, and Y. Zhang, "Online coordinated charging decision algorithm for electric vehicles without future information," in *IEEE SmartGridComm*, Oct. 2013.

[18] S. Zhao, X. Lin, and M. Chen, "Peak-Minimizing Online EV Charging: Price of Uncertainty and Algorithm Robustification," in *IEEE INFOCOM*, Hong Kong, China, April 2015.

[19] ——, "Robust Online Algorithms for Peak-Minimizing EV Charging under Multi-Stage Uncertainty," *IEEE Transactions on Automatic Control*, April 2017.

[20] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic Provisioning in Next-Generation Data Centers with On-Site Power Production," in *IEEE INFOCOM*, Hong Kong, China, April 2015.

[21] 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.

[22] http://iitmicrogrid.net/.

[23] in *http://cvxr.com/cvx/*.

APPENDIX

A. YDS Algorithm

The YDS algorithm [9] is re-stated in Algorithm 2.

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 2: 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.,

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

B. Proof of Lemma 1

Proof. Consider the job arrival pattern depicted in Fig. 4. 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 optimal offline YDS solution is 2 (Fig. 5.(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 algorithm will behave quite differently (Fig. 5.(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. 5.(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 $T \to \infty$. □

C. Proof of Lemma 2

Proof. Consider the same job arrival pattern shown in Fig. 4. 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 $2p_lT$, 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

$$x_1 = \frac{2p_lT + (1-p_l)T}{T} = 1 + p_l. \tag{15}$$

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 $2p_lT$, 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

$$x_k = \frac{2p_lT + (1-p_l)T(\sum_{s=0}^{k-1}2^{-s}) - T\sum_{s=1}^{k-1}2^{-s}x_s}{2^{-(k-1)}T}. \tag{16}$$

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 $T \to \infty$. □

D. 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 $i = 1, 2, ..., n$.

Consider an arbitrary J_n satisfying (6). We pick any $t_0 = 1, 2, ..., n$, 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

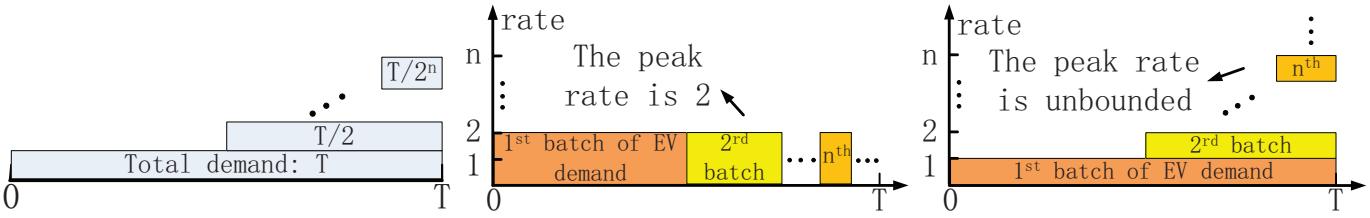
$$A_{i,n}^{J_n'} = CR_{i,n}^{J_n'}, A_{i,n}^{J_n'} + R_{i,n}^{J_n'} = A_{i,n}^{J_n} + R_{i,n}^{J_n}. \tag{17}$$

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

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

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

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



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

Fig. 4. EV-demand arrival pattern.

Fig. 5. The service profiles of two algorithms.

We can apply the above procedure for $t_0 = 1, 2, ..., n$ 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

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

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

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

and simplify (9) as

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

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

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

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

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

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

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

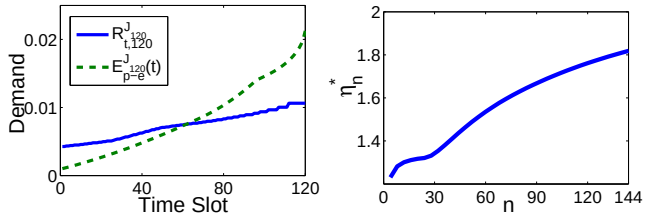
Then, the fractional LP problem (21) can be converted to the following equivalent LP problem, i.e.,

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

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 [23] 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. 6.(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. 6. Example.

Fig. 6.(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. 6.(b) shows how η_n^* varies with respect to n. Based on the values of η_n^*'s, we finally obtain $\eta^* = 1.8185$.

E. Proof of Lemma 6

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

Suppose that $\sum_{t=t_1}^{t_2}(A_{t,t_2}^J + R_{t,t_2}^J) \le \sum_{t=t_1}^{t_2} E_t^J$ 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 $\sum_{t=1}^{d}(A_{t,d}^J + R_{t,d}^J) \le \sum_{t=1}^{d} E_t^J$. 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 $\sum_{t=t_b+1}^{d}(A_{t,d}^J + R_{t,d}^J) \le \sum_{t=t_b+1}^{d} E_t^J$. Then, job k must be finished before time d, which contradicts to our assumption. □

F. Proof of Theorem 5

Proof. We only focus on the second part of Lemma 6. According to Lemma 6, we only need to show that for all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$,

$$\sum_{t=t_1}^{t_2}(A_{t,t_2}^J + R_{t,t_2}^J) \le \sum_{t=t_1}^{t_2} \eta^* E_{J(t),\text{off}}^*.$$

Equivalently, we need to show that

$$\eta^* \ge \frac{\sum_{t=t_1}^{t_2}(A_{t,t_2}^J + R_{t,t_2}^J)}{\sum_{t=t_1}^{t_2} E_{J(t),\text{off}}^*}. \tag{24}$$

To show inequality (24), we need to draw a connection between the right hand side (R.H.S.) of (24) and the optimization problem (9). We first simplify (9) by substituting $A_{t,n}^{J_n}$ by a_t, $R_{t,n}^{J_n}$ by r_t, and $E_{\text{pe}}^{J_n}(t)$ by b_t. Then, (9) can be transformed to the following equivalent optimization problem:

$$\begin{aligned} \max_{a_t, r_t \ge 0} \quad & \frac{\sum_{t=1}^{n}(a_t + r_t)}{\sum_{t=1}^{n} b_t} \\ \text{subject to} \quad & b_t = \max_{j=1,...,h_n(t)} \left\{ \frac{\sum_{i=j}^{t} a_t + \sum_{i=j}^{h_n(t)} r_t}{n-j+1} \right\} \\ & 0 \le a_t \le C r_t \end{aligned} \tag{25}$$

For $n = t_2 - t_1 + 1$, the optimal solution of the optimization problem (25) is then $\eta_{t_2-t_1+1}^*$.

We now consider (24). Since the job sequence J satisfies (6), we must have $0 \le A_{t,t_2}^J \le C R_{t,t_2}^J$ for all $t = t_1, ..., t_2$. Suppose that the following inequality holds,

$$E_{J(t),\text{off}}^* \ge \max_{j=t_1,...,h'(t)} \left\{ \frac{\sum_{i=j}^{t} A_{i,t_2}^J + \sum_{i=j}^{h'(t)} R_{i,t_2}^J}{t_2 - j + 1} \right\}, \tag{26}$$

where $h'(t) = \min\{t + L, t_2\}$. Then, if we substitute A_{t,t_2}^J by a'_{t-t_1+1}, R_{t,t_2}^J 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 (24) is no greater than the optimal value of the following optimization problem.

$$\begin{aligned} \max_{a'_t, r'_t \ge 0} \quad & \frac{\sum_{t=1}^{t_2-t_1+1}(a'_t + r'_t)}{\sum_{t=1}^{t_2-t_1+1} b'_t} \\ \text{subject to} \quad & 0 \le a'_t \le C r'_t \\ & b'_t \ge \max_{j=1,...,h_{t_2-t_1+1}(t)} \left\{ \frac{\sum_{i=j}^{t} a'_t + \sum_{i=j}^{h_{t_2-t_1+1}(t)} r'_t}{t_2 - t_1 + 1 - j + 1} \right\} \end{aligned} \tag{27}$$

It is easy to see that the optimal value of (27) is smaller than or equal to the optimal value of (25) with n replaced by $t_2 - t_1 + 1$. Therefore,

$$\text{R.H.S. of } (24) \le \eta_{t_2-t_1+1}^* \le \eta^*,$$

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. (26). 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

$$E_{J(t),\text{off}}^* = \max_{I}\{g_{J(t)(I)}\} \ge \max_{I \in \mathcal{I}}\{g_{J(t)(I)}\}. \tag{28}$$

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

$$g_{J(t)}(I) = \frac{\sum_{i=j}^{t_2}(A_{i,t_2}^{J(t)} + R_{i,t_2}^{J(t)})}{t_2 - j + 1}. \tag{29}$$

Note that at time $t = t_1, ..., t_2$, for any walk-in job k that contributes to the term $\sum_{i=j}^{t} A_{i,t_2}^J$ (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 $\sum_{i=j}^{t_2} A_{i,t_2}^{J(t)}$. 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

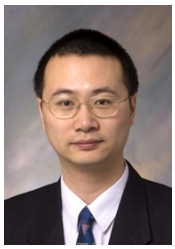
$$\sum_{i=j}^{t_2}(A_{i,t_2}^{J(t)} + R_{i,t_2}^{J(t)}) \ge \sum_{i=j}^{t} A_{i,t_2}^J + \sum_{i=j}^{h'(t)} R_{i,t_2}^J. \tag{30}$$

Combining Eqn. (29), (28) and (30), we immediately obtain

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

Therefore, Eqn. (26) holds, and thus Eqn. (24) follows. We then conclude that the EPS algorithm is a feasible online algorithm with CR η^*. □

Shizhen Zhao received his B.S. from Shanghai Jiao Tong University, China in 2010, and Ph.D. degree from Purdue University, West Lafayette, IN, in 2015. He is currently working in Google Platform Networking team. His research interests are in the analysis, control and optimization in wireless networks, smart grid, and software defined networks.



Xiaojun Lin (S'02 M'05 SM'12) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 2000 and 2005, respectively. He is currently an Associate Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin's research interests are in the analysis, control and optimization of wireless and wireline communication networks. He received the IEEE INFOCOM 2008 best paper and 2005 best paper of the year award from Journal of Communications and Networks. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005. He received the NSF CAREER award in 2007. He was the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, the TPC co-chair for ACM MobiHoc 2009, and the Mini-Conference co-chair for IEEE INFOCOM 2012. He is currently serving as an Associate Editor for IEEE/ACM Transactions on Networking and an Area Editor for (Elsevier) Computer Networks Journal, and has served as a Guest Editor for (Elsevier) Ad Hoc Networks journal.

Minghua Chen (S'04 M'06 SM'13) received his B.Eng. and M.S. degrees from the Dept. of Electronic Engineering at Tsinghua University in 1999 and 2001, respectively. He received his Ph.D. degree from the Dept. of Electrical Engineering and Computer Sciences at University of California at Berkeley in 2006. He spent one year visiting Microsoft Research Redmond as a Postdoc Researcher. He joined the Dept. of Information Engineering, the Chinese University of Hong Kong in 2007, where he is currently an Associate Professor. He is also an Adjunct Associate Professor in Institute of Interdisciplinary Information Sciences, Tsinghua University. He received the Eli Jury award from UC Berkeley in 2007 (presented to a graduate student or recent alumnus for outstanding achievement in the area of Systems, Communications, Control, or Signal Processing) and The Chinese University of Hong Kong Young Researcher Award in 2013. He also received several best paper awards, including the IEEE ICME Best Paper Award in 2009, the IEEE Transactions on Multimedia Prize Paper Award in 2009, and the ACM Multimedia Best Paper Award in 2012. He is currently an Associate Editor of the IEEE/ACM Transactions on Networking. He serves as TPC Co-Chair of ACM e-Energy 2016 and General Chair of ACM e-Energy 2017. His current research interests include energy systems (e.g., smart power grids and energy-efficient data centers), intelligent transportation system, online competitive optimization, distributed optimization, multimedia networking, wireless networking, and delay-constrained network coding.