Robust Online Algorithms for Peak-Minimizing EV Charging under Multi-Stage Uncertainty

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 robust online control algorithms for EV (electrical vehicle) charging under the scenario of an aggregator serving a large number of EVs together with its background load, using both its own renewable energy (for free) and the energy procured from the external grid. The goal of the aggregator is to minimize its peak procurement from the grid, subject to the constraint that each EV has to be fully charged before its deadline. Further, the aggregator can predict the future demand and the renewable energy supply with some levels of uncertainty. The key challenge here is how to develop a model that captures the prior knowledge from such prediction, and how to best utilize this prior knowledge to reduce the peak under future uncertainty. In this paper, we first propose a 2-level increasing precision model (2-IPM), to capture the system uncertainty. We develop a powerful computational approach that can compute the optimal competitive ratio under 2-IPM over any online algorithm, and also online algorithms that can achieve the optimal competitive ratio. A dilemma for online algorithm design is that an online algorithm with good competitive ratio may exhibit poor average-case performance. We then propose a new *Algorithm-Robustification* procedure that can convert an online algorithm with reasonable average-case performance to one with both the optimal competitive ratio and good average-case performance. The robustified version of a well-known heuristic algorithm, Receding Horizon Control (RHC), is found to demonstrate superior performance via trace-based simulations.**

I. Introduction

Replacing fossil fuels by renewable energy is a major priority all over the world (see, e.g., [2][3] and the references therein). However, high penetration of renewable energy poses an immense challenge to the existing power grid. Specifically, renewable energy from wind and solar is known to exhibit high variability and uncertainty. As renewable generation varies, the grid needs additional flexibility to balance the demand and supply [4]. In this paper, we focus on balancing the variability and uncertainty of the renewable supply by exploiting the flexibility from electric vehicle (EV) charging demand [5], which is a typical example of deferrable demands [6]. We expect that future EV demand can potentially be huge. Currently, transportation consumes 29% of the total energy in the US, while electricity consumes 40%. If a large fraction of the vehicles are electrified, their charging jobs will provide an enormous amount of demand-side flexibility, which could be used to compensate the variability and uncertainty due to high penetration of renewable energy.

Our goal in this paper is thus to develop robust online control algorithms for EV charging that minimizes the impact of variability and uncertainty of renewable energy to the grid. Specifically, we consider a demand-aggregator who has its own background demand and renewable energy supply (the latter is assumed to be of no cost), and who manages a large number of EVs. Such an aggregator could represent an apartment or office building with a parking garage, a campus, or a micro-grid [7]. As the EVs arrive and are connected to the charging stations, each of them specifies a deadline for the charging request to be completed. We model the objective of the aggregator as minimizing the peak consumption from the grid (the background load plus the EV charging rate, minus the renewable supply) under the constraints that all EVs must be charged before their deadlines. Our choice of the peak-minimization objective is motivated by the following two considerations. First, a large peak consumption-level requires the grid to provision the corresponding generation and transmission capacity in order to meet the demand. Thus, a large peak not only increases the overall cost of supplying energy, but also poses danger to grid-stability. Second, utility companies have already developed peak-based pricing schemes to encourage large customers (including aggregators) to reduce their peak and smoothen their demand. In this type of pricing schemes, the customers are charged based on not only the total usage in a billing period, but also 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 $c_1 \sum_i E_i + c_2 \max_i\{E_i\}$ [8]. In typical schemes (e.g., Wisconsin electric power company [9]), the unit charge for peak usage c_2 (between \$9.03/kW-month and \$9.38/kW-month) is approximately 200 times the unit charge for total usage c_1 (between \$0.03/kWh and \$0.05/kWh). As a result, the peak charge can constitute a large fraction (ranging from 30% to 50% [10]) of the total electricity bill. Under this type of pricing schemes, when the aggregator reschedules EV charging jobs, the total energy consumption from the grid does not change. It is the peak demand that is changed. Hence, minimizing the aggregator's operating cost is also equivalent to minimizing its peak consumption. Further, the potential benefit of peak reduction is huge. For campus-level aggregators (e.g., [7]), the peak energy is usually in the

An earlier version of this work has been presented at IEEE INFOCOM 2015 [1].

order of 20MW. Then, every one percent of peak reduction will correspond to $0.01 \times 20\text{MW} \times \$9/\text{kW-month} \times 12 = \$21600$ saving per year.

The main difficulty in the above EV charging problem comes from the *sequentially-revealed* uncertainty in both the demand and the renewable supply. If all the demand and the supply could be precisely predicted in advance, one could have used an offline algorithm to compute the optimal charging schedule that minimizes the peak. Specifically, there exist both centralized algorithms (e.g., the YDS algorithm in [11]) and decentralized algorithms (e.g., the valley-filling algorithm in [12]) for peak-minimizing EV-charging in an offline setting. However, in reality both the future demand and supply can exhibit significant uncertainty. Further, such uncertainty is typically sequentially revealed. That is, at any point in time, the past demand and supply are revealed, yet the future uncertainty is still unknown. Receding Horizon Control (RHC) can be used to modify an offline algorithm to deal with sequentially-revealed uncertainty [13]. However, as readers will see in the example in Section II-C, a RHC-based algorithm could lead to much larger peak consumption levels (a similar observation was also reported in [12]).

Although control problems with uncertainty have been widely studied, existing approaches are not suitable for the peak-minimizing EV-charging problem that we study in this paper. If a probabilistic model is known for future uncertainty, then the problem can be cast as a stochastic control problem. However, obtaining an accurate probabilistic model of uncertainty can be challenging, especially when the renewable supply is non-stationary and highly-correlated across time. Further, the complexity of solving the optimal control decision for a given probabilistic distribution, e.g., using Markov Decision Processes [14], is extremely high. Another way to deal with uncertainty is robust optimization [15][16], where future uncertainty is modeled in a set. The resulting solution is designed to optimize the worst-case performance for all possible realizations of the uncertainty in this set. However, robust optimization typically does not deal with sequential decisions.

The third approach, which we adopt in this work, is to design competitive online algorithms [17]. Compared to robust optimization, online algorithms are specifically designed for sequential decisions. In a peak-minimizing problem closely related to ours [18], it was shown that, even without any future information of job arrivals and deadlines, one can design a competitive online algorithm whose peak consumption is at most a constant factor $e = 2.718$ above the offline optimal (where the latter assumes that the future information is known in advance). This constant factor is referred to as the competitive ratio of the online algorithm. However, this line of research also encounters a number of challenges. First, existing results on competitive online algorithms either are based on very simple models of future uncertainty [19], or do not assume any model at all. As a result, the worst-case performance and the corresponding competitive ratio are often quite poor. In practice, both renewable supply and EV demands can be predicted to a certain degree. Intuitively, such prediction can provide very useful information for eliminating uninteresting worst cases, and thus sharpening the competitive ratio of online algorithms. However, to the best of our knowledge, there is no systematic methodologies for designing competitive online algorithms under more sophisticated models of future uncertainty. The second challenge, which in fact applies to typical robust-optimization results as well [16], is that the algorithms are only optimized for the worst-case. As a result, their average-case performance can be quite poor [19]. Given that the worst-case input may only occur very rarely, the aggregator may then be hesitant to endorse the resulting algorithm.

In this work, we make two contributions that precisely address these difficulties. First, from the *methodology* point of view, we extend the framework of online algorithm to incorporate available future knowledge captured by prediction. Specifically, we propose a more general set-based model, called 2-IPM (2-level increasing precision model), to capture the *sequentially* revealed uncertainty of renewable supply, EV demand, and background load. Compared to the traditional set-based model used in robust optimization [15], a key novelty of 2-IPM is that it can model the *sequential* nature of multiple predictions, i.e., predictions can be made at multiple instants (e.g., day-ahead prediction versus intra-day prediction), and the predictions closer to the target time tend to be more accurate (e.g., intra-day prediction is usually more accurate than day-ahead prediction). For any given 2-IPM model, we then develop a powerful computation procedure to find the smallest competitive ratio in terms of the peak consumption. This smallest competitive ratio can thus be viewed as a measure of the "price of uncertainty" under the 2-IPM. As readers will see in Section V-B, our 2-IPM yields much lower price-of-uncertainty compared to the uncertainty models in [19].

Second, from the *algorithm* point of view, we propose a general "robustification" procedure to design online algorithms to address both worst-case performance and average-case performance. Specifically, given any online algorithm with good average performance (in terms of the peak), this robustification procedure can convert it to one with not only good average-case performance, but also the optimal competitive ratio. We apply this robustification procedure to a well-known online algorithm, called Receding Horizon Control (RHC), which demonstrates good average-case performance, but poor worst-case competitive ratio. Our numerical results in Section V-C indicates that the robustified-RHC algorithm achieves both good average-case and worst-case performance.

The rest of the paper is organized as follows. Section II defines the system model that captures the renewable uncertainty, and motivates the design of online algorithms with the optimal competitive ratio. We derive a fundamental lower bound on the competitive ratio in Section III, and propose a general framework based on Algorithm Robustification in Section IV to design online algorithms that achieve the above lower bound. Real-trace based simulation results are provided in Section V to demonstrate both the improvement on the optimal competitive ratio and the effectiveness of our Algorithm Robustification procedure. In Section VI, we conclude.

II. System Model

We consider an aggregator serving its EV demand and background demand using both its own renewable energy (which is assumed to be cost-free) and the energy procured from the external grid. We assume that time is slotted, and index a time-slot by an integer in $\mathbb{T} = \{1, ..., T\}$, where T is the time-horizon considered. We represent the EV demand by a $T \times T$ upper-triangular matrix $\mathbf{a} = [a_{i,j}]$, where $a_{i,j}$ is the total deferrable (EV) demand with arrival time i and deadline $j \geq i$. We represent the net non-deferrable demand by a $T \times 1$ vector $\mathbf{b} = [b_i]$, where b_i is the background demand at time i minus the renewable energy available at time i. Note that when the penetration of renewable energy is high, the net non-deferrable demand will exhibit significant uncertainty. Using the flexibility in the EV demand, the goal of the aggregator is to schedule EV charging jobs against high renewable uncertainty such that the peak energy procured from the grid is minimized.

A. Model for Prediction and Uncertainty

In practice, there exists considerable uncertainty in both the net non-deferrable demand and the deferrable demand. Specifically, we define a $(T - t + 2) \times 1$ vector $\mathbf{x}(t) = [a_{t,t}, ..., a_{t,T}, b_t]^T$ to include both the EV demand with arrival time t and the net non-deferrable demand at time t. Note that the aggregator will know the precise value of $\mathbf{x}(t)$ only at and after time-slot t. In the rest of this paper, we will say that "the value of $\mathbf{x}(t)$ is revealed at time t". At a time $s < t$, the value of $\mathbf{x}(t)$ is uncertain to the aggregator. However, the aggregator can use various sources of information (such as weather forecast) to predict the future values of these uncertain quantities in order to improve its decision. In practice, such predictions can be taken multiple times, e.g., if the operating time-horizon is a day, one prediction can be made before the day (called "day-ahead" prediction), and another prediction can be made a few hours before time t (called "intra-day" prediction). In general, intra-day prediction is more accurate than the day-ahead prediction because it is closer to the real time. Next, we will present a model, called 2-IPM (2-Level Increasing Precision Model), to model the uncertainty associated with such prediction procedures. We note that, although for ease of exposition the model below only assume one intra-day prediction, both 2-IPM and the subsequent results can be easily generalized to multiple intra-day predictions.

Specifically, we assume that at time 0 (before the first time-slot), a day-ahead prediction is available for every $\mathbf{x}(t), t \in \mathbb{T}$. For each future time-slot t, the day-ahead prediction provides two $(T-t+2)\times 1$ vectors $\hat{\mathbf{x}}^L(0,t), \hat{\mathbf{x}}^U(0,t)$, which are lower and upper bounds, respectively, to $\mathbf{x}(t)$. In other words, the future value of $\mathbf{x}(t)$ must lie within

$$\hat{\mathbf{x}}^L(0,t) \leq \mathbf{x}(t) \leq \hat{\mathbf{x}}^U(0,t). \quad (1)$$

Then, at a later time $u_t, 1 \leq u_t < t$, another intra-day prediction is performed. (One example of u_t could be $u_t = \max\{1, t - L\}$, i.e., the intra-day prediction is performed L time-slots ahead.) The intra-day prediction provides another two $(T-t+2)\times 1$ vectors $\hat{\mathbf{x}}^L(u_t,t), \hat{\mathbf{x}}^U(u_t,t)$, that are *better* lower and upper bounds to $\mathbf{x}(t)$ than the day-ahead prediction. In other words, the following will hold:

$$\hat{\mathbf{x}}^L(0,t) \leq \hat{\mathbf{x}}^L(u_t,t) \leq \mathbf{x}(t) \leq \hat{\mathbf{x}}^U(u_t,t) \leq \hat{\mathbf{x}}^U(0,t). \quad (2)$$

Obviously, a key difference between day-ahead prediction and intra-day prediction is that they are performed at different times. Thus, while the value of day-ahead prediction, $\hat{\mathbf{x}}^L(0,t), \hat{\mathbf{x}}^U(0,t)$ for all t, are known even before time-slot 1, the value of $\hat{\mathbf{x}}^L(u_t,t)$ and $\hat{\mathbf{x}}^U(u_t,t)$ will not be known until time-slot u_t. (We will say that the value of $\hat{\mathbf{x}}^L(u_t,t)$ and $\hat{\mathbf{x}}^U(u_t,t)$ are revealed at time u_t.) Thus, from time-slot 0 to time-slot $u_t - 1$, although the aggregator does not know the future intra-day prediction for $\mathbf{x}(t)$ that will be performed at time u_t, it does know that this future intra-day prediction will be more accurate. In order to model this knowledge, we assume that there exists a $(T - t + 2) \times 1$ vector $W(u_t, t) \leq \hat{\mathbf{x}}^U(0,t) - \hat{\mathbf{x}}^L(0,t)$, *which is known at time* 0, that bounds the (future) intra-day prediction gap $\hat{\mathbf{x}}^U(u_t,t) - \hat{\mathbf{x}}^L(u_t,t)$, i.e.,

$$\hat{\mathbf{x}}^U(u_t,t) - \hat{\mathbf{x}}^L(u_t,t) \leq W(u_t,t). \quad (3)$$

In other words, the aggregator knows the (increased) precision level of future intra-day predictions that will be performed at time u_t, even though it does not know the exact bounds of this intra-day prediction before time u_t.

Remark 1: The novelty of our 2-IPM is in modeling the increasing precision of interval predictions. Note that compared to point predictions [20], interval predictions (i.e., (1) and (2)) provide additional information on the accuracy level of the prediction. While there have been many recent studies of interval predictions [21][22], the 2-IPM allows us to capture the increasing precision as time evolves and to rigorously study its effect on the corresponding sequential decision problem.

Remark 2: Readers may question what happens when the real value $\mathbf{x}(t)$ or the bounds for the intra-day prediction falls outside of the day-ahead predicted interval (1). This is indeed possible in practice, because some predictions may be wrong. Nevertheless, for ease of theoretical analysis, we will assume that (1) and (2) are always satisfied. This assumption is justified because in practice these bounds are usually chosen such that the value of $x(t)$ will fall into the predicted intervals with high probability, and we can always tighten the intra-day prediction bounds so that (2) is satisfied. The case where these bounds are violated will be studied in Section IV-C and Section V-D.

We summarize how the variables defined above are revealed in time. At time 0, the aggregator only knows $Y = \{\hat{\mathbf{x}}^L(0,t), \hat{\mathbf{x}}^U(0,t), W(u_t,t), t = 1, 2, ..., T\}$. At time-slot t, the aggregator knows the revealed $\mathbf{x}(s)$ for all $s \leq t$, and the intra-day prediction for any time-slot s such that $u_s \leq t$. This set of information is summarized in $Z_t = \{\mathbf{x}(s), s = 1, 2, ..., t\} \bigcup \{\hat{\mathbf{x}}^L(u_s,s), \hat{\mathbf{x}}^U(u_s,s), u_s \leq t\}$. Note that the set Z_t increases with time t. Let $Z = \bigcup_{t\in\mathbb{T}} Z_t$ denote all quantities that were *not* known day-ahead. Thus, at time t, the aggregator knows both Y and Z_t, but not those quantities in $Z \setminus Z_t$.

Throughout this paper, we will view Y as given, because Y is known day-ahead (before any scheduling decisions are made). The uncertainty comes entirely from Z. Even though

we do not know the exact realization of Z before-hand, the knowledge of Y (the day-ahead prediction) restricts Z (the realization) into a smaller search space. Specifically, for a given Y, all elements of Z and Y must satisfy constraints (1)-(3), and all the qualified Z's form a demand-trace set $\mathcal{Z}_Y$, i.e.,

$$\mathcal{Z}_Y = \{Z \quad : \quad \text{the elements in } Y \text{ and } Z \text{ satisfy } (1)-(3) \text{ for the fixed } Y\}.$$

B. Objective

We are interested in designing online algorithms for scheduling EV demand that minimize the peak energy drawn from the grid. At each time $t = 1, 2, ..., T$, an online algorithm π must determine the amount of energy $E_t(Z_t, \pi)$ drawn from the grid, *based only on the knowledge of* Y *and* Z_t. (Note that we have assumed that Y is fixed. Hence, we have omitted Y in the notation $E_t(Z_t, \pi)$ for simplicity, but the dependency of $E_t(Z_t, \pi)$ on Y is implicitly assumed.) In other words, the decision at time t cannot be based on the values of any quantity in $Z \setminus Z_t$ that will be revealed in the future. The online algorithm π is said to be feasible if all the EV demands can be completely served before deadlines using the sequence of energy-procurement decisions $[E_t(Z_t, \pi), t \in \mathbb{T}]$ minus the revealed non-deferrable demand (i.e., background demand minus renewable energy). Let $E^p_\pi(Z) = \max_t\{E_t(Z_t, \pi)\}$ be the peak energy drawn from the grid using a feasible online algorithm π. The aggregator is interested on reducing $E^p_\pi(Z)$. However, it is not possible for one online algorithm to minimize $E^p_\pi(Z)$ for all Z's. Instead, we consider an *offline* solution provided by a "genie" that knows the entire future Z in advance. This genie can set the energy procurement $E_t(Z)$ at each time-slot t based on Z. This genie can then solve the following problem offline:

$$\min_{\text{All demand can be completely served}} \max_t\{E_t(Z)\}. \tag{4}$$

Let $E^*_{\text{off}}(Z)$ be the optimal offline solution to (4). Clearly, for any online algorithm π, we will have $E^*_{\text{off}}(Z) \le E^p_\pi(Z)$. We can then evaluate the performance of an online algorithm π by comparing it to the above offline optimal. Specifically, for a fixed Y, define the competitive ratio (CR) $\eta_Y(\pi)$ of an online algorithm π as the maximum ratio between $E^p_\pi(Z)$ and $E^*_{\text{off}}(Z)$ under all possible $Z \in \mathcal{Z}_Y$, i.e., $\eta_Y(\pi) = \max_{Z\in\mathcal{Z}_Y}\left\{\frac{E^p_\pi(Z)}{E^*_{\text{off}}(Z)}\right\}$. In other words, the competitive ratio characterizes how *in the worst case* the online algorithm can perform more poorly compared to the offline optimal.

In the rest of the paper, we will first find an achievable lower bound on the competitive ratio $\eta_Y(\pi)$ under 2-IPM, which characterizes the fundamental limits how 2-level prediction can improve the worst-case performance. Then, we will propose a systematic approach to design online algorithms with both the optimal competitive ratio and good average-case performance.

Remark 3: Related to the above peak-minimization objective, another approach is to set an upper limit for the peak power consumption, and schedule EV charging within the power limit [13]. However, this approach does not minimize peak demand charges.

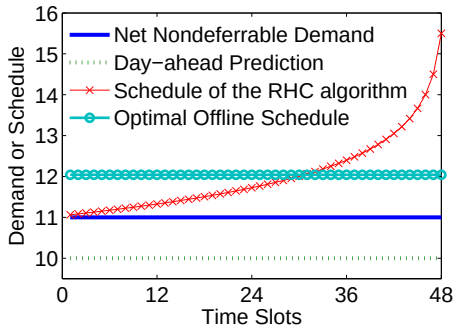


Fig. 1. Poor performance of the RHC algorithm. In this example, the day-ahead prediction consistently underestimates the real demand, and the RHC algorithm leads to a peak that is 1.29 times higher than the optimal offline peak. In comparison, an optimal online algorithm can achieve a peak that is at most 1.06 times higher than the optimal offline peak.

C. A Motivating Example

Before describing our main results, we use an example to illustrate that it is not trivial to design an online algorithm with good competitive ratio.

Consider Receding Horizon Control (RHC) [23], which is a popular approach for dealing with sequential-revealed uncertainty. At each time-slot, using the offline solution, RHC computes the entire schedule for the future based on past information that has been revealed and most-recently predicted future demand/supply. However, RHC only executes the first step of the schedule in the current time-slot. Then, in the next time-slot, as new information is revealed, RHC repeats this procedure with the newly-revealed information and again applies the first step. Clearly, if the future demand/supply is close to their predicted values (which likely holds in the average case), the performance of RHC will be quite good. However, the following example illustrates that the worst-case performance of RHC can be very poor, especially when the future prediction is persistently wrong. Specifically, there are 48 time-slots. One EV charging job arrives at the beginning of time-slot 1, departs at the end of time-slot 48, and the total demand is 48. (In other words, there is no uncertainty for the EV demand.) The day-ahead predictions of the net non-deferrable demands are 10 for all time-slots, with an uncertain interval of $[8, 12]$, i.e., future non-deferrable demands are assumed to fall within this interval. Suppose that the real values of non-deferrable demands turn out to be 11 for all time-slots (note that this realization falls within the uncertainty set, and there is no intra-day prediction in this example). The offline optimal peak for this example is simply $48/48+11 = 12$. If we apply the RHC algorithm to this example, the resulting peak will be 15.5 (see Fig. 1), and thus the worst-case competitive ratio would be no smaller than $15.5/12 = 1.29$. In contrast, applying the computational approach in Section III to this uncertainty set, the optimal competitive ratio can be shown to be 1.06, which is much smaller than 1.29 in the case of RHC. As shown in Fig. 1, the poor performance of the RHC algorithm can be understood as follows. In the first time-slot, RHC sees that the current non-deferrable demand (i.e., 11) is

TABLE I. LIST OF NOTATIONS.

T	Total Number of time-slots.
$\mathbf{a} = [a_{i,j}]$	A $T \times T$ matrix representing EV demand.
$\mathbf{b} = [b_i]$	A $T \times 1$ matrix representing net non-deferrable demand.
$\mathbf{x}(t)$	Demand variable revealed at time t, including $a_{t,t}, ..., a_{t,T}$ and b_t.
$\hat{\mathbf{x}}^L(0,t)$, $\hat{\mathbf{x}}^U(0,t)$	Lower bound and upper bound of $\mathbf{x}(t)$. Known at time 0 (day-ahead prediction).
$\hat{\mathbf{x}}^L(u_t,t)$, $\hat{\mathbf{x}}^U(u_t,t)$	Lower bound and upper bound of $\mathbf{x}(t)$. Known at time u_t (intra-day prediction).
$W(u_t,t)$	Maximum gap between $\hat{\mathbf{x}}^L(u_t,t)$ and $\hat{\mathbf{x}}^U(u_t,t)$. Known at time 0 (day-ahead prediction).
Y	All quantities known at time 0, including $\hat{\mathbf{x}}^L(0,t)$, $\hat{\mathbf{x}}^U(0,t)$ and $W(u_t,t)$ for any t.
Z_t	All uncertain quantities revealed at or before time t, including $\{\mathbf{x}(s), s \le t\}$ and $\{\hat{\mathbf{x}}^L(u_s,s), \hat{\mathbf{x}}^U(u_s,s), u_s \le t\}$.
Z	All uncertain quantities revealed after time 0, including all Z_t's.
$E_t(Z_t,\pi)$	Total amount of energy drawn from the grid under an causal online algorithm π.
$E^*_{\text{off}}(Z)$	Offline optimal peak energy consumption (assuming Z is known at time 0, which is non-causal).
$\eta_Y(\pi)$	Competitive ratio of the online algorithm π.

higher than the predicted value (i.e., 10). However, RHC still assumes that the future non-deferrable demand is the same as the predicted value (i.e., 10). Because the EV demand is flexible, RHC computes a schedule that smoothens out the future demand. Thus, the charging rate at the time-slot 1 only increases slightly and the corresponding amount of EV demand is deferred to the future. This procedure is repeated, until towards the end of the horizon, RHC realizes that the earlier predictions have been consistently wrong. However, now the decisions in the earlier time-slots cannot be reverted. RHC has no choice but to increase the peak to accommodate the EV demand deferred from the past. In summary, since RHC fails to account for the possible future deviations from the predicted values, it leads to poor performance in the worst case. In contrast, the methodology that we propose below will explicitly account for future uncertainty, and hence will provide much better worst-case guarantees.

D. Summary of Notations

For ease of reference, we list all the notations in this paper in Table I.

III. FUNDAMENTAL LIMIT OF THE COMPETITIVE RATIO

In this section, we extend the computation framework in [19] to find a fundamental lower bound on the competitive ratio $\eta_Y(\pi)$ of any algorithm π. This lower bound will be given by the solution of the optimization problem (9). However, solving (9) is much more difficult than that in [19]. In Section III-B, we will develop a general convexification technique to convexify (9). Such a convexification technique generalizes fractional-linear programs [24], and thus may be of independent interest.

We need the following three lemmas throughout this section.

Lemma 1: Given a demand realization Z, a sufficient and necessary condition for a service profile $E = [E_1, E_2, ..., E_T]$ to be feasible, i.e., all demand can be completed before the corresponding deadlines, is that for all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$, the following inequality holds,

$$\sum_{t=t_1}^{t_2} \sum_{s=t}^{t_2} a_{t,s} + \sum_{t=t_1}^{t_2} b_t \le \sum_{t=t_1}^{t_2} E_t. \tag{5}$$

Proof: See Appendix A. ■

Lemma 1 is a generalization of Lemma 6 in [19]. It states that, in order for a service profile $E = [E_1, E_2, ..., E_T]$ to be feasible, the total energy procured from the grid plus the renewable energy supply in any time interval $[t_1, t_2]$ must be no smaller than the total demand that must be served in the same interval. Further, the condition (5) is also sufficient. Specifically, if we use the Earliest-Deadline-First (EDF) policy to serve the demand, then all the demand can be finished before the corresponding deadlines.

Clearly, Lemma 1 also applies to an online algorithm π. Specifically, a feasible online algorithm π must be able to support all possible demand trace $Z \in \mathcal{Z}_Y$. Then, according to Lemma 1, the service profile $[E_t(Z_t,\pi), t = 1, 2, ..., T]$ must satisfy (5) for all possible demand trace $Z \in \mathcal{Z}_Y$.

When studying online algorithms, we usually compare the peak of an online algorithm with the peak of an offline optimal algorithm. The following lemma gives a close-form formula of the offline optimal peak.

Lemma 2: Given a realization of Z, for any $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$, define the intensity of an interval $J = [t_1, t_2]$ as

$$g_J(Z) = \frac{\sum_{t=t_1}^{t_2} (\sum_{s=t}^{t_2} a_{t,s} + b_t)}{t_2 - t_1 + 1}. \tag{6}$$

Then, the offline optimal peak is given by

$$E^*_{\text{off}}(Z) = \max\{0, \max_J \{g_J(Z)\}\}. \tag{7}$$

Lemma 2 states that the offline optimal peak is equal to the maximum intensity over all possible intervals. The reason that we have a "0" term in Eqn. (7) is that the power procured from the grid must be non-negative. Lemma 2 is easy to prove. Clearly, the offline optimal peak must be no smaller than the maximum intensity given by (7). Otherwise, the demand inside the interval with the maximum intensity cannot be finished before the end of the interval. Further, there exists an offline solution with a peak exactly equal to $E^*_{\text{off}}(Z)$. Specifically, we can construct an offline solution that always procure $E^*_{\text{off}}(Z)$ amount of energy at each time-slot. According to Lemma 1, the service profile of this offline solution can indeed finish all the demand before the corresponding deadlines.

A. Lower Bound

Consider an online algorithm π with competitive ratio $\eta_Y(\pi)$. We first study the maximum value for $E_t(Z_t, \pi)$ given a realization Z. Recall that the decision $E_t(Z_t, \pi)$ should only depend on Z_t. Further, we note that there may exist different realizations Z that yield the same value of Z_t. Thus, the value of $E_t(Z_t, \pi)$ must be chosen such that it is no greater than $\eta_Y(\pi)$ times the offline-optimal peak for any possible future realization. Let

$$E_t^{pe}(Z_t) = \inf_{Z' \in \mathcal{Z}_Y, Z'_t = Z_t} E^*_{\text{off}}(Z'), \tag{8}$$

where the superscript "pe" stands for "peak estimation". Then, we have the following lemma. The detailed proof is in Appendix B.

Lemma 3: Given an online algorithm π with competitive ratio $\eta_Y(\pi)$, we must have $E_t(Z_t, \pi) \le \eta_Y(\pi) E_t^{pe}(Z_t)$.

We now apply Lemma 1. If π is feasible, then for all $Z \in \mathcal{Z}_Y$ and all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$, we must have

$$\sum_{t=t_1}^{t_2} \left(\sum_{s=t}^{t_2} a_{t,s} + b_t \right) \le \sum_{t=t_1}^{t_2} E_t(Z_t, \pi) \le \eta_Y(\pi) \sum_{t=t_1}^{t_2} E_t^{pe}(Z_t).$$

Define the following optimization problem:

$$\eta^*_{t_1,t_2}(Y) = \sup_{Z \in \mathcal{Z}_Y} \quad \frac{\sum_{t=t_1}^{t_2} \left(\sum_{s=t}^{t_2} a_{t,s} + b_t \right)}{\sum_{t=t_1}^{t_2} E_t^{pe}(Z_t)} \tag{9}$$

Let $\eta^*_Y = \max_{t_1 \le t_2, t_1, t_2 \in \mathbb{T}} \{\eta^*_{t_1,t_2}(Y)\}$. Then, η^*_Y provides a lower bound on the competitive ratio, which is stated below.

Theorem 4: For any feasible online algorithm π, its competitive ratio must be no smaller than η^*_Y, i.e., $\eta_Y(\pi) \ge \eta^*_Y$.

The above arguments share some similarity to Theorem 4 in [19]. However, computing η^*_Y here is much more difficult than that in [19]. The computation of η^*_Y requires solving the optimization problem (9). Like in [19], the denominator of the objective function in (9) is the optimal value of another optimization problem. In general, such a bi-level optimization problem is NP-hard [25]. In [19], special structures of the problem are exploited to convert a similar bi-level optimization problem to a convex problem, which is then easier to solve. However, the techniques in [19] critically rely on the property that the uncertain quantities (i.e., the EV demand matrix) can be freely scaled up or down without violating system constraints. Unfortunately, this property does not hold in this paper. Specifically, if all the uncertain quantities in Z is component-wise multiplied by a large constant, it may violate the bounds from day-ahead prediction in (1). In the next subsection, we will develop a more general convexification technique than that in [19] to convexify the optimization problem (9).

B. Convexification of Problem (9)

We present the key convexification technique in Lemma 5.

Lemma 5: Consider the following optimization problem:

$$\begin{aligned} M_1 = \sup_{\vec{x}, y} \quad & (c^T \vec{x} + \alpha)/y \\ \text{subject to} \quad & y = f(\vec{x}), A\vec{x} \le b, \end{aligned} \tag{10}$$

where $\vec{x}, c$ are $n \times 1$ vectors, A is a $m \times n$ matrix, b is a $m \times 1$ vector, and α, y are scalars. Suppose that the following two conditions hold:

(a) $f(\cdot)$ is a convex function of $\vec{x}$, and $f(\vec{x}) > 0$ over the entire constrained region of $A\vec{x} \le b$;

(b) There exists $\vec{x}$ satisfying $A\vec{x} \le b$, such that $c^T\vec{x} + \alpha > 0$.

Then, the optimal value M_1 of (10) is equal to the optimal value M_2 of the following optimization problem:

$$\begin{aligned} M_2 = \sup_{\vec{x}', u} \quad & c^T \vec{x}' + \alpha u \\ \text{subject to} \quad & 1 \ge u f(\vec{x}'/u), A\vec{x}' \le bu, u > 0. \end{aligned} \tag{11}$$

Remark 4: The optimization problem (11) can be transformed from (10) as follows. First, we let $\vec{x}' = \vec{x}/y, u = 1/y$. Then, the resulting optimization problem will be similar to (11), except that we have a constraint $1 = uf(\vec{x}'/u)$ instead of $1 \ge uf(\vec{x}'/u)$. Note that $f(\vec{x})$ is a convex function. $uf(\vec{x}'/u)$ must also be convex in $(\vec{x}, u)$ because it is the perspective of $f(\cdot)$ [26]. Therefore, after relaxing the constraint $1 = uf(\vec{x}'/u)$, the optimization problem (11) becomes a convex problem, and can be efficiently solved. The result of Lemma 5 can be viewed as a generalization of fractional-linear program [24], which requires $f(\vec{x})$ to be linear. The detailed proof is available in Appendix C.

We are now ready to convexify (9). We assume that the condition (b) holds in our problem, i.e., $\sum_{t=t_1}^{t_2} (\sum_{s=t}^{t_2} a_{t,s} + b_t) > 0$ for some $Z \in \mathcal{Z}_Y$. In other words, we cannot serve all the EV demand and the background demand using only the renewable energy. This assumption is reasonable, because that the renewable energy is highly variable, and we need to procure energy from the external grid when the renewable energy turns out to be low. It remains to show that the condition (a) also holds for (9), i.e., $\sum_{t=t_1}^{t_2} E_t^{pe}(Z_t)$ is a convex function of Z. Obviously, it is sufficient to show that $E_t^{pe}(Z_t)$ is a convex function of Z_t. The convexity of $E_t^{pe}(Z_t)$ is ensured by the following lemma.

Lemma 6: Suppose that $f(x, y)$ is a convex function defined on a convex set D. Let $D_x = \{y : (x, y) \in D\}$, then $g(x) = \inf_{y \in D_x} f(x, y)$ is also a convex function.

Proof: See Appendix D. ∎

Specifically, we can view Z_t as x, and $Z' \setminus Z_t$ as y. Then, based on (8), we can rewrite $E_t^{pe}(Z_t)$ as $E_t^{pe}(x) = \inf_y \{E^*_{\text{off}}(x, y)\}$. The region of (x, y) is a convex set because all the constraints in (1)-(3) are linear constraints. Further, it is easy to verify that $E^*_{\text{off}}(x, y)$ is a convex function of (x, y) according to (7). Therefore, $E_t^{pe}(Z_t)$ is a convex function.

Remark 5: Although the 2-IPM in this paper assumes only one intra-day prediction, the results in this section can also be generalized to general prediction models, with multiple intra-day predictions. Specifically, each intra-day prediction can be characterized by a similar set of inequalities as (2)-(3). As a result, the set $\mathcal{Z}_Y$ of all possible realizations can still be

described by a set of linear constraints. Then, $E_t^{pe}(Z_t)$ is still a convex function based on Lemma 6, and thus the optimal competitive ratio in (9) can be computed efficiently using the convexification technique in Lemma 5. Further, due to the same reason, the Algorithm Robustification procedure to be developed in Section IV will also work for such more general prediction models.

IV. Algorithm Design and Robustification

Note that we have obtained a lower bound η_Y^* for the competitive ratio of any online algorithm, the next step is to design an online algorithm that can attain this lower bound. It turns out that we can use the idea of the EPS algorithm proposed in [19]. Specifically, at each time, an online algorithm can set $E_t(Z_t, \pi) = \eta_Y^* E_t^{pe}(Z_t)$. We also refer to this algorithm as the EPS (Estimated Peak Scaling) algorithm because it always scales up the estimated value $E_t^{pe}(Z_t)$ of the lowest possible future peak by the competitive ratio η_Y^*. Like in [19], it is not difficult to prove from the definition of η_Y^* that this EPS algorithm is feasible for any input $Z \in \mathcal{Z}_Y$ because the condition (5) is always satisfied. Thus, the EPS algorithm attains the optimal competitive ratio η_Y^*.

The problem of this EPS algorithm, however, is that although it achieves the optimal competitive ratio for the worst-case input, its average-case performance can be quite poor, i.e., its peak can be high for many other inputs. To understand this dilemma, note that according to Lemma 3, any online algorithm with optimal competitive ratio η_Y^* should set $E_t(Z_t, \pi)$ to be no larger than $\eta_Y^* E_t^{pe}(Z_t)$. In the case of the EPS algorithm, it always set $E_t(Z_t, \pi)$ to the highest possible value. Thus, it can be viewed as the most *conservative* algorithm. If the future input indeed followed the worst-case, such conservatism would have been essential to attain the optimal competitive ratio: by serving more demand up-front, the EPS algorithm avoids a potentially large peak in the future. However, if the future input is different from the worst case, the EPS algorithm will likely be *too conservative*. For example, if the future input followed precisely the one that produces the value $E_t^{pe}(Z_t)$ in (8), then using a rate $E_t(Z_t, \pi) = E_t^{pe}(Z_t)$ would have been sufficient. Thus, one could argue that, since the worst-case perhaps occurs very rarely, using EPS may turn to be a poor choice in most scenarios.

This conflict between worst-case performance and average-case performance is not uncommon in the context of competitive online algorithms [17]. An algorithm designed for the worst-case can exhibit poor average-case performance, making it less appealing for practical implementation. Ideally, we would like to design an algorithm with both good worst-case and good average-case performance. In the rest of this section, we will present a novel "robustification" procedure to design such an algorithm. Our key idea is as follows. We first identify not one, but a class of algorithms that all attain the optimal competitive ratio. Then, starting from any algorithm with reasonable average-case performance, we "robustify" its decision by comparing it to the above class of algorithms. The resulting algorithm will then achieve both the optimal competitive ratio and good average-case performance.

A. Online Algorithms with the Optimal Competitive Ratio

Suppose that π is an optimal online algorithm with competitive ratio η_Y^*. For any realization $Z \in \mathcal{Z}_Y$, we next study all possible values of $E_t(Z_t, \pi)$ that the algorithm π can take. The upper bound on $E_t(Z_t, \pi)$ is given by Lemma 3, i.e.,

$$E_t(Z_t, \pi) \leq \eta_Y^* E_t^{pe}(Z_t). \tag{12}$$

Next, we derive a lower bound for $E_t(Z_t, \pi)$.

At time t, we use r_{t,t_1} to represent the total not-yet-served demand with deadline no greater than t_1, which includes all the remaining demand (with deadline no greater than t_1) from the previous time-slots, the newly arrived net demand, and the newly arrived EV demand with deadline no greater than t_1. Consider any time instant $t_1 \geq t$, given any input Z with the first part being Z_t, we must have

$$E_t(Z_t, \pi) + \eta_Y^* \sum_{s=t+1}^{t_1} E_s^{pe}(Z_s) \geq r_{t,t_1} + \sum_{s=t+1}^{t_1} \left(\sum_{w=s}^{t_1} a_{s,w} + b_s \right). \tag{13}$$

Here, $a_{s,w}$ and b_s are the elements of Z. The right hand side is the total demand that has to be served within $[t, t_1]$, while the left hand side is the maximum possible energy procurement from the grid (assuming that each future energy procurement rate is set to the upper bound in (12)).

We move the term "$\eta_Y^* \sum_{s=t+1}^{t_1} E_s^{pe}(Z_s)$" from the left-hand-side of (13) to the right-hand-side, and then we have

$$E_t(Z_t, \pi) \geq r_{t,t_1} + \sum_{s=t+1}^{t_1} \left(\sum_{w=s}^{t_1} a_{s,w} + b_s - \eta_Y^* E_s^{pe}(Z_s) \right). \tag{14}$$

Note that (14) must hold for all possible future inputs. Define the following optimization problem that maximizes the right hand side of (14) over all possible future inputs:

$$\sup_{Z' \in \mathcal{Z}_Y, Z'_t = Z_t} \sum_{s=t+1}^{t_1} \left(\sum_{w=s}^{t_1} a'_{s,w} + b'_s - \eta_Y^* E_s^{pe}(Z'_s) \right) \tag{15}$$

where $a'_{s,w}, b'_s$ are the corresponding elements of Z'. Let $R^*_{\eta_Y^*}(Z_t, t_1)$ be the optimal value of (15). Then, in order to attain the optimal competitive ratio, the following must hold

$$E_t(Z_t, \pi) \geq r_{t,t_1} + R^*_{\eta_Y^*}(Z_t, t_1). \tag{16}$$

Finally, the above inequality must hold for all $t_1 \geq t$. Therefore, we obtain the following lower bound for $E_t(Z_t, \pi)$:

$$E_t(Z_t, \pi) \geq \max_{t_1 \geq t} \{ r_{t,t_1} + R^*_{\eta_Y^*}(Z_t, t_1) \}. \tag{17}$$

Remark 6: Note that $E_s^{pe}(Z'_s)$ is a convex function (see Section III-B). Therefore, the objective of (15) is a concave function. Further, both constraints ($Z' \in \mathcal{Z}_Y$ and $Z'_t = Z_t$) of (15) are linear constraints. Hence, (15) is a convex optimization problem, and thus can be efficiently solved.

We summarize the above discussion into Lemma 7.

Lemma 7: For any feasible η_Y^*-competitive online algorithm, we must have

$$\max_{t_1 \geq t} \{ r_{t,t_1} + R^*_{\eta_Y^*}(Z_t, t_1) \} \leq E_t(Z_t, \pi) \leq \eta_Y^* E_t^{pe}(Z_t).$$

Remark 7: We note a key difference in the qualitative nature of the upper and lower bounds. The upper bound of $E_t(Z_t,\pi)$ depends only on the optimal competitive ratio η_Y^* and the past realization Z_t, but is independent of the past decisions $E_s(Z_s,\pi), s<t$. In contrast, the lower bound of $E_t(Z_t,\pi)$ also depends on the past energy procurement $E_s(Z_s,\pi), s<t$. Due to this reason, the lower bound is more *adaptive*: if the energy procured from the grid in the previous time-slots is large, we will have less remaining demand r_{t,t_1}, and thus have a smaller value for the lower bound. Such an ability to adjust based on the past decisions is the key reason that we can robustify an algorithm with good average-case performance to have optimal competitive ratio.

Input: Time-slot t, the remaining demand r_{t,t_1} and the part Z_t that has been revealed.

1 Compute the lower bound (17) and upper bound (12), and let $E_t(Z_t,\pi)$ be any value in between.

2 The aggregator purchases $E_t(Z_t,\pi)$ amount of energy from the external power grid, and uses the renewable energy and the purchased energy $E_t(Z_t,\pi)$ to serve the existing demand. The aggregator first serves the background demand b_t, and then serves the deferrable demand by the *earliest deadline first* (EDF) policy (i.e., demand with earlier deadline gets served first). The aggregator will stop serving demand if all the available demand at time t is completely served or the amount of energy $E_t(Z_t,\pi)$ is exhausted.

Algorithm 1: A Class of Optimal Online Algorithms

Motivated by Lemma 7, we define a class of online algorithms, called ABS (Adaptive Bound-based Scheduling), in Algorithm 1. We first show that all ABS algorithms are well-defined. Specifically, we show that the lower bound (17) is always no greater than the upper bound (12). Therefore, it is always feasible to pick a value for $E_t(Z_t,\pi)$ at each slot.

Lemma 8: Given $Z\in\mathcal{Z}_Y$ and an algorithm π in the class ABS, at each time-slot t, we must have

$$\max_{t_1\geq t}\{r_{t,t_1}+R^*_{\eta_Y^*}(Z_t,t_1)\}\leq \eta_Y^* E_t^{pe}(Z_t). \tag{18}$$

Lemma 8 is the key of this section, and its proof is non-trivial. We can see that both sides of (18) depend on η_Y^*. In fact, η_Y^* is the smallest value such that (18) always holds. For any $\eta<\eta_Y^*$, it is possible to construct a case $Z'\in\mathcal{Z}_Y$ such that $\max_{t_1\geq t}\{r_{t,t_1}+E^*_\eta(Z_t,t_1)\}>\eta E_t^{pe}(Z_t)$ for some t.

Proof: Recall that $\max_{t_1\geq t}\{r_{t,t_1}+R^*_{\eta_Y^*}(Z_t,t_1)\}$ is the smallest value of $E_t(Z_t,\pi)$ that satisfies (14) for all possible t_1's and all possible future realizations. Therefore, in order to prove (18), it suffices to show that $\eta_Y^*E_t^{pe}(Z_t)$ also satisfies (14), i.e.,

$$\eta_Y^*\sum_{s=t}^{t_1}E_s^{pe}(Z_s)\geq r_{t,t_1}+\sum_{s=t+1}^{t_1}\left(\sum_{w=s}^{t_1}a_{s,w}+b_s\right), \tag{19}$$

for all $t_1\geq t$, and all possible $a_{s,w}$ and b_s.

We prove by induction on t. When $t=1$, $r_{1,t_1}=b_1+\sum_{s=1}^{t_1}a_{1,s}$. Therefore, the right hand side of (19) is $\sum_{s=1}^{t_1}(\sum_{w=s}^{t_1}a_{s,w}+b_s)$. Based on the definition of η_Y^* in (9), (19) holds trivially for all $t_1\geq t$ when $t=1$.

Assume that (19) holds for a given t and all $t_1\geq t$. We will show that (19) holds for $t+1$ and all $t_1\geq t+1$. Note that

$$r_{t+1,t_1}=(r_{t,t_1}-E_t(Z_t,\pi))^+ + b_{t+1}+\sum_{s=t+1}^{t_1}a_{t+1,s},$$

where $(x)^+=\max\{x,0\}$, $(r_{t,t_1}-E_t(Z_t,\pi))^+$ is the remaining demand with deadline no greater than t_1 and $b_{t+1}+\sum_{s=t+1}^{t_1}a_{t+1,s}$ is the new arrival demand with deadline no greater than t_1. If $(r_{t,t_1}-E_t(Z_t,\pi))^+=0$, then (19) holds trivially based on the definition of η_Y^*. In the rest of this proof, we only need to consider $(r_{t,t_1}-E_t(Z_t,\pi))^+>0$. In this case,

$$r_{t+1,t_1}=r_{t,t_1}-E_t(Z_t,\pi)+b_{t+1}+\sum_{s=t+1}^{t_1}a_{t+1,s}.$$

We prove by contradiction. Assume that there exist $\tilde{t}_1,\tilde{a}_{s,w},\tilde{b}_s$, such that $(r_{t,\tilde{t}_1}-E_t(\tilde{Z}_t,\pi))^+>0$ and

$$\eta_Y^*\sum_{s=t+1}^{\tilde{t}_1}E_s^{pe}(\tilde{Z}_s)<r_{t+1,\tilde{t}_1}+\sum_{s=t+2}^{\tilde{t}_1}\left(\sum_{w=s}^{\tilde{t}_1}\tilde{a}_{s,w}+\tilde{b}_s\right).$$

Then,

$$\begin{aligned}
&0\\
<\;& r_{t+1,\tilde{t}_1}+\sum_{s=t+2}^{\tilde{t}_1}\left(\sum_{w=s}^{\tilde{t}_1}\tilde{a}_{s,w}+\tilde{b}_s\right)-\eta_Y^*\sum_{s=t+1}^{\tilde{t}_1}E_s^{pe}(\tilde{Z}_s)\\
=\;& r_{t,\tilde{t}_1}-E_t(\tilde{Z}_t,\pi)+\tilde{b}_{t+1}+\sum_{s=t+1}^{\tilde{t}_1}\tilde{a}_{t+1,s}\\
&+\sum_{s=t+2}^{\tilde{t}_1}\left(\sum_{w=s}^{\tilde{t}_1}\tilde{a}_{s,w}+\tilde{b}_s\right)-\eta_Y^*\sum_{s=t+1}^{\tilde{t}_1}E_s^{pe}(\tilde{Z}_s)\\
=\;&\sum_{s=t+1}^{\tilde{t}_1}\left(\sum_{w=s}^{\tilde{t}_1}\tilde{a}_{s,w}+\tilde{b}_s\right)-\eta_Y^*\sum_{s=t+1}^{\tilde{t}_1}E_s^{pe}(\tilde{Z}_s)\\
&+r_{t,\tilde{t}_1}-E_t(\tilde{Z}_t,\pi)\\
\leq\;& R^*_{\eta_Y^*}(\tilde{Z}_t,\tilde{t}_1)+r_{t,\tilde{t}_1}-E_t(\tilde{Z}_t,\pi).
\end{aligned}$$

The last inequality holds based on the definition of the optimization problem (15). The above derivation implies that

$$E_t(\tilde{Z}_t,\pi)<R^*_{\eta_Y^*}(\tilde{Z}_t,\tilde{t}_1)+r_{t,\tilde{t}_1},$$

which contradicts to our choice of $E_t(\tilde{Z}_t,\pi)$.

Hence, (19) holds for $t+1$ and all $t_1\geq t+1$. By induction, (19) holds for all t's and $t_1\geq t$. Thus, Lemma 8 holds. ■

Next, we show that all ABS algorithms are indeed optimal.

Lemma 9: Any algorithm π in the class of ABS is feasible and achieves the optimal competitive ratio of η_Y^*.

Proof: The proof is straightforward. First, based on the choice of $E_t(Z_t,\pi)$, it is easy to see that the peak of the

algorithm π never exceeds η_Y^* times the offline optimal peak. Thus, the algorithm π is η_Y^*-competitive. Second, let $t_1 = t$ in (16). It is easy to check that $R^*_{\eta_Y^*}(Z_t, t) = 0$. Therefore, $E_t(Z_t, \pi) \geq r_{t,t}$, which implies that no demand will violate its deadline at time t. This completes the proof. ■

B. Algorithm Robustification

We have characterized the structure of optimal online algorithms. It only remains to find an online algorithm in ABS that also has good average performance. Our strategy is to take any algorithm with reasonable average-case performance, and convert it into one in the class ABS. We call this procedure *Algorithm-Robustification*. The Algorithm-Robustification procedure is formally stated in Algorithm 2. Specifically, Step 3 of the procedure states that, if $E_t(Z_t, \pi)$ is between the upper bound and the lower bound, then we use the decision of the original algorithm π. Otherwise, we "robustify" the decision by setting $E_t(Z_t, \pi_{\text{Robust}})$ to one of the bounds, so that the resulting "robustified" algorithm belongs to ABS. Intuitively, this procedure implies that for most inputs the robust version of π will likely behave in the same way as the original algorithm. Hence, the average-case performance will likely be similar. However, if there is a danger that the competitive ratio may be violated in the future, the robustified algorithm will then take the more conservative decision represented by the bounds.

Input: A realization $Z \in \mathcal{Z}_Y$, the optimal competitive ratio η_Y^* and any online algorithm π.
Output: An optimal online algorithm π_{robust} and its schedules $E_t(Z_t, \pi_{\text{Robust}})$.

1 **for** $t = 1 : T$ **do**
2 Compute $\alpha = \max_{t_1 \geq t}\{r_{t,t_1} + R^*_{\eta_Y^*}(Z_t, t_1)\}$, $\beta = \eta_Y^* E_t^{pe}(Z_t)$, and the schedule $E_t(Z_t, \pi)$ of the online algorithm π.
3 Set $E_t(Z_t, \pi_{\text{Robust}}) = M_\alpha^\beta(E_t(Z_t, \pi))$, where $M_\alpha^\beta(x) = \max\{\min\{x, \beta\}, \alpha\}$.
4 **end**

Algorithm 2: Algorithm-Robustification Procedure

In practice, in Section V-C, we will robustify a well-known online algorithm, called receding-horizontal-control (RHC). The RHC algorithm usually exhibits good average-case performance [23]. However, its worst-case competitive ratio can be very poor (see Section II-C). We then apply this Algorithm-Robustification procedure to the RHC algorithm. This robustified RHC algorithm will then achieve optimal competitive ratio in the worst case. Further, our numerical results demonstrate that the robustified RHC algorithm achieves almost the same average-case performance as the RHC algorithm.

C. Accommodating Incorrect Predictions

Our prediction model in Section II-A has implicitly assumed that the day-ahead and intra-day predictions are always "correct." This assumption implies that the upper and lower bounds of intra-day prediction are always within the upper and lower bounds of day-ahead prediction, and the real-time values are always within the upper and lower bounds or intra-day predictions. Then, under this assumption, Lemma 8 guarantees that the robustification procedure in Algorithm 2 will always work, i.e., the lower limit α computed by Step 2 of Algorithm 2) is always no greater than the upper limit β.

What if the predictions are incorrect? In reality, the predicted bounds are based on some confidence intervals. Thus, there will always be some small probability that the future realization falls outside of these bounds. In that case, Lemma 8 may not hold, and we may have $\alpha > \beta$ in Algorithm 2). Obviously, our robustification procedure will fail.

Interestingly, Lemma 8 also suggests a way to "fix" the robustification procedure when the above situation happens. Essentially, when $\alpha > \beta$ in Algorithm 2), it implies that the originally-computed competitive ratio η_Y^* (assuming that the day-ahead prediction is correct) is no longer the correct competitive ratio for the amount of uncertainty faced by the aggregator. The value of η_Y^* will have to be increased. We note the monotonicity of α and β with respect to η^*: it is easy to check that α is a monotone decreasing function of η_Y^*, while β is a monotone increasing function of η_Y^*. Thus, if we keep increasing η_Y^*, eventually we can make $\alpha \leq \beta$. Based on this discussion, we add the following *parameter-tuning step* between Step 2 and Step 3 in Algorithm 2:

(Parameter-Tuning Step): If $\alpha > \beta$, increase η_Y^* until $\alpha \leq \beta$.

With the above parameter-tuning step, the algorithm robustification procedure can proceed even when the predictions are incorrect. Of course, incorect predictions could negatively impact the performance of the system. We will study this impact using simulation in Section V-D.

V. Simulation

We conduct simulation using real traces from two data sets. Elia [27], Belgium's electricity transmission system operator, provides day-ahead predictions and real-time values of background demand and renewable energy for every hour of each day. (However, Elia [27] does not provide data for intra-day prediction.) The National Household Travel Survey (NHTS) dataset [28] provides vehicle driving records for 150147 households. By assuming that future EV driving patterns are similar, it is not difficult to use the data in [28] to synthesize a model for the EV demand (see Fig. 2 and Appendix E1), including EV arrival time, deadline and amount of energy charging demand, as has been done in earlier works in [29].

A. The importance of Accounting for Uncertainty

We note that the day-ahead prediction in our 2-IPM consists of an upper bound and a lower bound for each time-slot. In contrast, the day-ahead prediction in Elia data-set [27] only contains one predicted value. Nonetheless, by comparing the difference between day-ahead predicted value and the real-time value over long periods of time (e.g., a year), it is easy to compute upper and lower bounds of the prediction error (for a given confidence level). Combining them with the day-ahead predicted values of [27], we can then generate the upper and lower bounds for day-ahead predictions as required in our model (see Appendices E2, E3, and F). In Fig. 3 (a), we apply

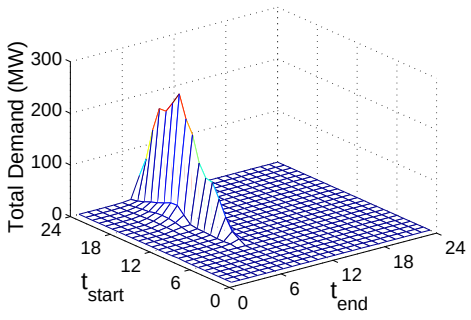
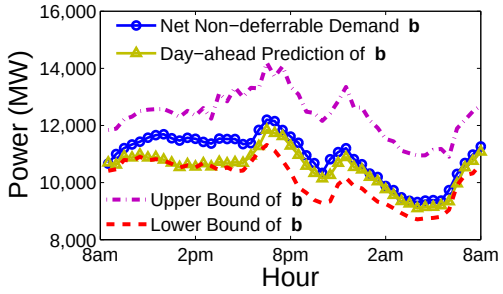


Fig. 2. Synthesized EV demand [28]. For any pair of discretized arrival time t_{start} and discretized departure time t_{end}, this figure plots the total EV demand.

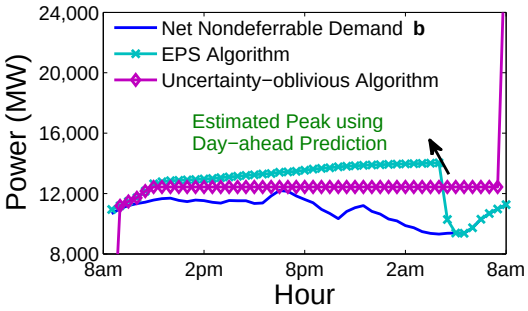
this methodology to Elia's data on background demand and renewable energy over a 24-hour period from 8am 02/05/2013 to 8am 02/06/2013, and plot the following versions of net non-deferrable demand $\mathbf{b}$ (as the background demand minus the renewable energy): the real-time value, the day-ahead predicted value directly from [27], and the upper and lower bounds of the real-time values as constructed above. From Fig. 3 (a), we can see that the gap between the upper and lower bounds can be quite large (up to 20% of the day-ahead predicted value). The dataset in [27] does not provide explicit intra-day prediction. Hence, in our first experiment, we only consider day-ahead prediction. Lastly, for EV demand, we scale up[1] the synthesized model (see Fig. 2) by a factor 20, and assume that the day-ahead prediction of the EV demand is always accurate. In other words, we consider the uncertainty of background demand and renewable energy only.

We next demonstrate that, even for the scenario with low uncertainty, an algorithm that is oblivious to future uncertainty may lead to large peak consumption levels. Specifically, we consider the following uncertainty-oblivious algorithm. At day-ahead, this uncertainty-oblivious algorithm assumes that the day-ahead predicted values of the background demand, the renewable energy (both from [27]) and the EV demand, are accurate. It thus computes the offline optimal peak and the corresponding charging schedule (e.g., one possible schedule is to procure at each time-slot the amount of energy equal to this offline optimal peak), and then applies this schedule during real-time operation. Note that there is a chance that this schedule may not meet the deadline constraints of some EV demands because the real-time values will differ from the predicted values. In that case, this uncertainty-oblivious algorithm will then need to procure additional energy at the time of the deadlines to meet the requirement of these EV demands. Intuitively, this algorithm will perform poorly even if there is only a slight deviation between the real-time values and the predicted values because it always wait until the last minute to remediate the prediction error. This is confirmed from Fig. 3 (b), where we plot the energy procurement schedule of

[1]This EV trace [28] is obtained based on 150147 households. However, Belgium has 4 million households. Scaling the EV demand up by 20 will correspond to the future scenario where all vehicles in Belgium are electrified.



(a) Net non-deferrable demand and its day-ahead prediction (including both point prediction and interval prediction) from 8am 02/05/2013 to 8am 02/06/2013.



(b) The decision of the uncertainty-oblivious algorithm vs. the EPS algorithm. The uncertainty-oblivious algorithm produces a large peak, when the demand turns out to be higher in real time. In contrast, the peak of the EPS algorithm is much lower.

Fig. 3. The EPS algorithm vs. the uncertainty-oblivious algorithm.

this uncertain-oblivious algorithm versus the EPS algorithm (discussed at the beginning of Section IV). The uncertainty-oblivious algorithm suffers a large peak at the last minute because the deadlines of most EV demands are 8am (see Fig. 2). In contrast, since the EPS algorithm increases the amount of energy procured early on, it avoids this last-minute peak. (We will see shortly that algorithms in the class of ABS will tend to have even lower peak than that of the EPS algorithm.) Hence, this figure clearly illustrates the importance of explicitly accounting for future uncertainty in the system.

B. 2-IPM and the Price of Uncertainty

We next evaluate the merit of the proposed 2-IPM in capturing the uncertainty of prediction. Note that given specific parameters of 2-IPM, we can calculate the lowest competitive ratio over all online algorithms (see Section III). This optimal competitive ratio can thus be viewed as a measure of the "price of uncertainty", i.e., it represents the increase in cost (compared to the offline optimal peak) due to the inherent uncertainty captured by 2-IPM. Note that we have simulated based entirely on real traces in Section V-A. In the rest of the numerical experiments, we will manipulate the trace to observe the performance in different settings.

We first compare the competitive ratio under 2-IPM versus that under the prediction model in [19]. Note that the uncertainty model in [19] assumes that the ratio between the future uncertainty (i.e., the walk-in demand in [19]) and the predicted value (i.e., the reserved demand in [19]) is bounded. However, the absolute quantity of the predicted value is not specified. Thus, we refer to the uncertainty model in [19] as a *relative uncertainty model*. In contrast, in 2-IPM the absolute quantities for the predicted upper/lower bounds are specified. Hence, we refer to 2-IPM as an *absolute uncertainty model*. One can map absolute uncertainty in this paper to relative uncertainty in [19] by using only the ratio between the prediction error and the predicted value. For instance, suppose $\mathbf{x}^{\text{DA}}(t)$ is the day-ahead predicted value. In 2-IPM, the upper and lower bounds of day-ahead prediction are specified as

$$\hat{\mathbf{x}}^L(0,t) = \mathbf{x}^{\text{DA}}(t)\times(1-\epsilon), \hat{\mathbf{x}}^U(0,t) = \mathbf{x}^{\text{DA}}(t)\times(1+\epsilon). \quad (20)$$

In contrast, with the relative uncertainty model in [19], only ϵ is specified, but not $\mathbf{x}^{\text{DA}}(t)$.

Intuitively, absolute uncertainty contains more information than relative uncertainty, and thus 2-IPM should yield lower competitive ratios. To confirmed this point, we use the day-ahead predicted values as shown in Fig. 3 (a), but varies the upper/lower bounds of day-ahead prediction by varying ϵ in (20). In Fig. 4, we plot the optimal competitive ratios under both 2-IPM and under the relative uncertainty model from [19], as ϵ varies from 0.05 to 0.2. We can see that, even with only day-ahead prediction, the optimal competitive ratios under 2-IPM are lower. For example, when $\epsilon = 0.2$, the competitive ratio reduces from 1.2 to 1.16, which corresponds to approximately 4% reduction on the peak demand (which is significant as 1% reduction corresponds to $0.01 \times 20\text{MW} \times \$9/\text{kW-month} \times 12 = \$21600$ saving per year for campus-level aggregators [7] with peak energy in the order of 20MW). In this sense, we argue that the price of uncertainty under 2-IPM is lower than that under a comparable model of relative uncertainty as in [19].

We next evaluate the impact of intra-day prediction. Note that the Elia data set [27] does not have intra-day prediction data. Thus, in the following we will artificially vary the parameters of intra-day prediction and evaluate the corresponding optimal competitive ratios. Such an evaluation methodology has a unique advantage: even before the operator carries out the intra-day prediction, our methodology will be able to reveal how useful such information will be in terms of reducing the optimal competitive ratio. Again, this knowledge of "price of uncertainty", i.e., how much the cost can be reduced by intra-day prediction, could be very useful in deciding which types of intra-day prediction to perform and how accurate they need to be. Specifically, we evaluate three types of intra-day prediction, i.e., hour-ahead prediction, 12-hour-ahead prediction and 18-hour-ahead intra-day prediction. For each type of intra-day prediction, we vary the intra-day prediction gap as

$$W(u_t,t) = \min\{2\epsilon_{\text{intra}} \times \mathbf{x}^{\text{DA}}(t), \hat{\mathbf{x}}^U(0,t) - \hat{\mathbf{x}}^L(0,t)\},$$

where $\hat{\mathbf{x}}^L(0,t), \hat{\mathbf{x}}^U(0,t)$ are the day-ahead predicted bounds specified in (20), and ϵ_{intra} is the parameter we can vary. In Fig. 4, we plot the corresponding optimal competitive ratios under several choices of ϵ_{intra}, as the ϵ (i.e., error of day-ahead prediction) varies from 0.05 to 0.2. We can make a number of interesting observations. First, even if the hour-ahead prediction is perfect (i.e., $\epsilon_{\text{intra}} = 0$), the optimal competitive ratio barely changes from the case with only day-ahead prediction. Intuitively, this is because the hour-ahead prediction is too late: most of the decisions have already been made well before such hour-ahead prediction becomes available. In contrast, a perfect 12-hour-ahead prediction reduces the optimal competitive ratio by 2%. Interestingly, even an imperfect 18-hour-ahead prediction can be very helpful. For example, when $\epsilon = 0.2$, 18-hour-ahead prediction with $\epsilon_{\text{intra}} = 0.08$ reduces the optimal competitive ratio from 1.16 (no intra-day prediction) to 1.13, which is comparable to the gain from a perfect 12-hour-ahead prediction. In practice, the earlier the intra-day prediction is performed, the less accurate it will likely be. Thus, the results in Fig. 4 will allow the operator to evaluate which type of intra-day prediction will be most useful, i.e., in reducing the cost of uncertainty.

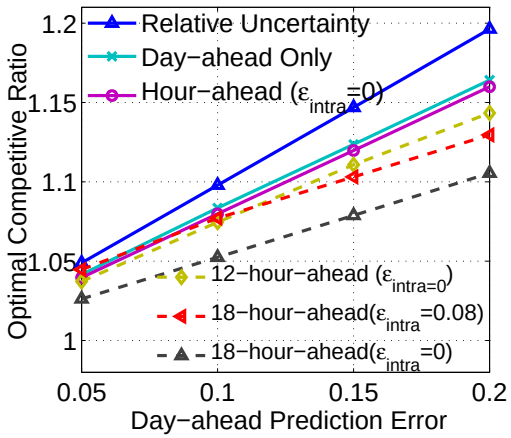
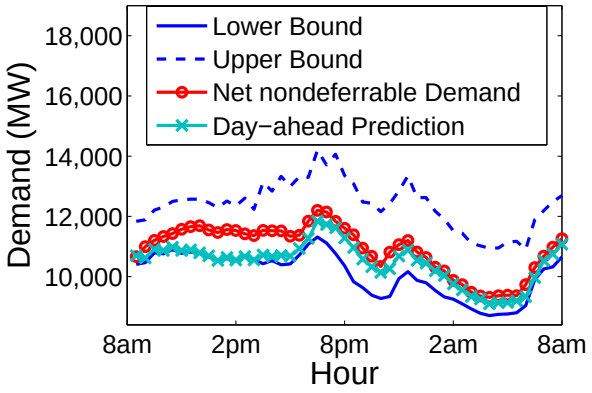


Fig. 4. Price of Uncertainty. Here, we compare the optimal competitive ratios between having only day-ahead prediction vs. having both day-ahead & intra-day predictions.

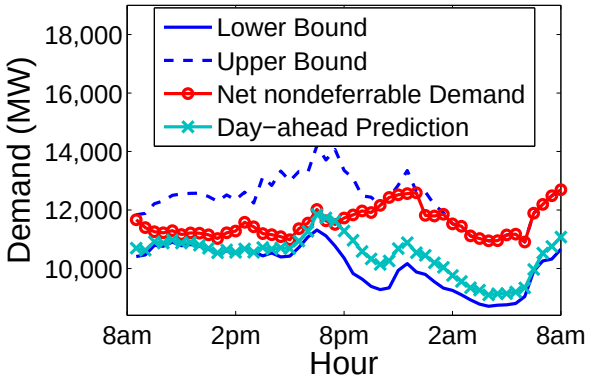
C. Worst-case vs. Average-case Performance

Until now we have focused on evaluating the worst-case competitive ratio. This worst-case competitive ratio is achievable by the EPS algorithm. However, as we discussed in Section IV, the EPS algorithm has poor average-case performance. In Section IV, we also present a robustification procedure that can be used to design algorithms with both good average-case performance and worst-case guarantees. Our next set of simulations will demonstrate this point.

Specifically, we will robustify a well-known heuristic algorithm, called Receding Horizon Control (RHC) [23]. Empirically, the RHC algorithm is often found to exhibit good average-case performance, especially when the future values of uncertain quantities are close to the predicted values. Nevertheless, we have also constructed a scenario in Section

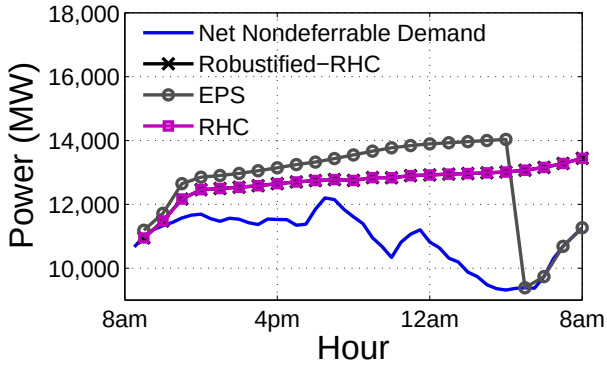


(a) Easy trace. The day-ahead predicted values are close to the real-time values of the net non-deferrable demand.

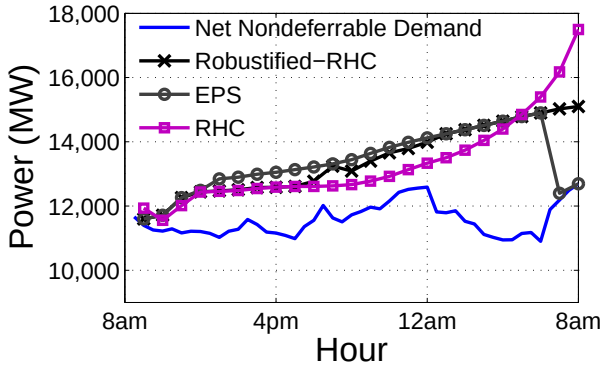


(b) Difficult trace. The day-ahead predicted values deviates significantly from the real-time values of the net non-deferrable demand.

Fig. 5. Net non-deferrable load of two Simulation traces.



(a) Easy trace. Note that the curve for "robustified-RHC" overlaps with that of "RHC". Compared to the RHC algorithm and the robustified-RHC algorithm, the EPS algorithm produces an unnecessarily larger peak.



(b) Difficult trace. The RHC algorithm cannot handle large uncertainty gracefully, and may lead to a large peak towards the end. Our robustified-RHC algorithm can take actions earlier to prevent a large peak in the future.

Fig. 6. Schedules under two Simulation traces.

II-C where the RHC algorithm performs much poorer than the optimal competitive ratio achieved by the EPS algorithm.

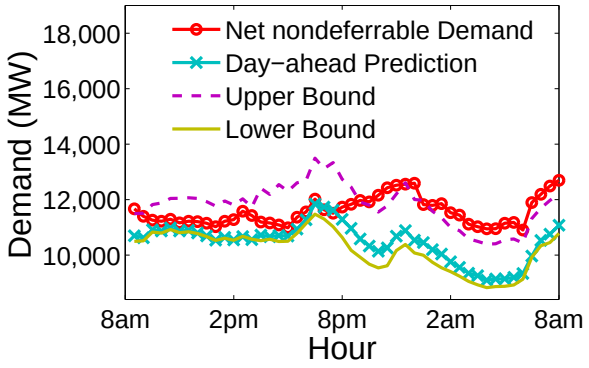
We next show that the robustified version of the RHC algorithm (according to Section IV-B), will achieve both good worst-case and average-case performance. We will use two traces (see Fig. 5). In both traces, the day-ahead predicted values of background demand and renewable energy, and their corresponding upper-bounds and lower-bounds, are the same and are obtained using the methodology in Section V-A (see Fig. 3 (a)). Both traces also employ the same intra-day prediction model that uses the values of the respective quantities one time-slot ahead as the slot-ahead prediction for the next time-slot, and the intra-day prediction gap $W(u_t, t)$ is set according to the maximum difference between the corresponding quantities in adjacent time-slots (see Eqn. (24) in Appendix F). Further, both traces use the same EV traces. Specifically, we use the EV demand shown in Fig. 2 as the day-ahead predicted value, and assume that the real demand vary uniformly randomly between 0.8 to 1.2 times the day-ahead predicted value. (We do not use intra-day prediction for EV demand.) However, the two figures differ in their revealed values of the net non-deferrable demand. In Fig. 5 (a), the revealed values of the net non-deferrable demand are closer to their day-ahead predicted values, while in Fig. 5 (b), the difference is much bigger (particularly at the end of the time-horizon). We will also refer to the trace in Fig. 5 (a) as the "easy trace", and the trace in Fig. 5 (b) as the "difficult trace".

In Fig. 6, we compare the schedules of the EPS algorithm, the RHC algorithm and the robustified-RHC algorithm under both traces. By comparing Fig. 6 (a) and 6 (b), we observe that the EPS algorithm cannot distinguish between the easy trace and the difficult trace, and its peaks are similar high in both traces. In other words, the EPS algorithm is too conservative: it treats every trace as the worst trace, and scales up $E_t^{pe}(Z_t)$ by the maximum value η_Y^*. In contrast, the RHC algorithm produces a much lower peak in the easy trace, when the day-ahead prediction is fairly accurate. However, its performance in the difficult trace is very poor. In the difficult trace, the day-ahead predicted values consistently underestimate the net non-deferrable load. As a result, the RHC algorithm sets its

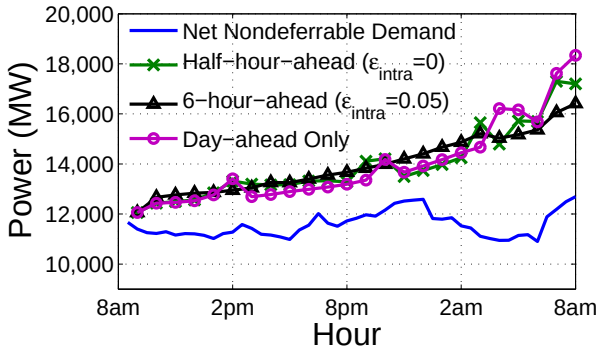
service rate too low at the beginning, and has to use a much higher rate when all the EV demand approaches the deadlines. Our robustified-RHC algorithm, on the other hand, inherits the benefits of both the EPS algorithm and the RHC algorithm. For the easy trace, the robustified-RHC algorithm gives virtually the same schedule as the RHC algorithm. For the difficult trace, the robustified-RHC algorithm detects that the service rate of the RHC algorithm is too low at about 6pm. It then increases the service rate afterwards, and avoids the potential peak in the end. In summary, the robustified-RHC algorithm achieves both good average-case and good worst-case performance.

D. Impact of Incorrect Predictions

The above simulations have assumed that the predictions are always correct, i.e., future realizations always fall within the bounds of earlier predictions, in which case the Robustified-RHC algorithm achieves both efficiency and robustness. We next simulate a setting with incorrect day-ahead predictions, in which case we use the parameter-tuning step introduced in Section IV-C. Interestingly, our key observation below is that, although finer intra-day predictions play less of a role when all predictions are correct (see Fig. 4), they produce a bigger impact on the system performance when day-ahead predictions may be incorrect. Specifically, our simulation is based on the difficult trace in Fig. 5 (b), except that we tighten the day-ahead predicted upper/lower bounds to simulate an incorrect day-ahead prediction (see Fig. 7 (a)). Thus, the real net-demand may go beyond the bounds. The simulation setup for EV demand and intra-day prediction are the same as that in Section V-C and Section V-B, respectively. We choose different parameter settings for the intra-day prediction, and compare the performance of the Robustified-RHC algorithm. From Fig. 7 (b), a key observation is that intra-day predictions reduce the peak significantly when day-ahead predictions are wrong. As a reference point for comparison, we compute the optimal competitive ratios under the three scenarios (day-ahead only, half-hour-ahead intra-day prediction with $\epsilon_{\text{intra}} = 0$, and 6-hour-ahead intra-day prediction with $\epsilon_{\text{intra}} = 0.05$), and find them to be 1.081, 1.077, and 1.063, respectively. In other words, intra-day predictions only reduce the competitive ratio slightly when all day-ahead predictions are correct. In contrast, when day-ahead predictions are incorrect, the resulting peaks in Fig. 7 (b) can differ substantially. For example, if we have intra-day prediction that is 6 hours ahead with $\epsilon_{\text{intra}} = 0.05$, the peak can be reduced by as much as 10% compared to the scenario with only day-ahead prediction. This simulation result demonstrate that, even though intra-day predictions are not very effective in reducing the optimal competitive ratio, they are still very important to the overall robustness of the system, especially when day-ahead predictions are incorrect.



(a) Difficult trace. (Note that the net-demand is higher than the day-ahead predicted upper bound after about 10pm.)



(b) Schedules of the Robustified-RHC algorithm under 3 different parameter settings. Intra-day prediction becomes more useful when day-ahead prediction is wrong. For example, with 6-hour-ahead intra-day prediction, even though the optimal competitive ratio only reduces by 1.7% (from 1.081 to 1.063), the real peak is reduced by 10%.

Fig. 7. Impact of wrong day-ahead predictions.

VI. Conclusion

We study competitive online EV-charging algorithms for an aggregator to reduce the peak procurement from the grid. We model the uncertainty of the system using the 2-IPM, which captures both day-ahead and intra-day predictions of the demand and the renewable energy supply. We then develop a powerful computation approach that can compute the optimal competitive ratio under 2-IPM over any online algorithms, and also develop a class of online algorithms that can achieve the optimal competitive ratio. Noting that algorithms with the optimal competitive ratio (e.g., the EPS algorithm) may have poor average-case performance, we then propose a new *Algorithm Robustification* procedure that can convert an online algorithm with reasonable average-case performance to one with both the optimal competitive ratio and good average-case performance. We demonstrate the superior performance of such robustified algorithms via trace-based simulations.

There are a number of interesting directions for future work. First, we can study the impact of batteries on peak shedding. In this paper, we assume that EVs can only be charged. In contrast, dedicated batteries can not only be charged during idle hours, but also be discharged during peak hours. This additional flexibility could further reduce the peak demand. Second, in this work we have focused only on one aggregator. In practice, such an aggregator needs to participate in the

overall electric power market. Then, it would be interesting to study the potential benefits of smart EV charging schemes for the entire power grid.

REFERENCES

[1] 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.

[2] J. Cottrell, F. Fortier, and K. Schlegelmilch, "Fossil fuel to renewable energy," Feb. 2015.

[3] Department of Communications, Energy & Natural Resources, "Ireland's Transition to a Low Carbon Energy Future," Dec. 2015.

[4] T. Mai, D. Sandor, R. Wiser, and T. Schneider, "Renewable electricity futures study. executive summary," National Renewable Energy Laboratory (NREL), Golden, CO., Tech. Rep., 2012.

[5] X. Hu, S. J. Moura, N. Murgovski, B. Egardt, and D. Cao, "Integrated Optimization of Battery Sizing, Charging, and Power Management in Plug-in Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, p. 1, 2015.

[6] R. Sioshansi, "Modeling the impacts of electricity tariffs on plug-in hybrid electric vehicle charging, costs, and emissions," *Mathematics of Operations Research*, vol. 60, no. 2, pp. 1–11, 2012.

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

[8] A. Bar-Noy, M. P. Johnson, and O. Liu, "Peak shaving through resource buffering," *Proceedings of WAOA*, vol. 5426, pp. 147–159, 2009.

[9] http://www.we-energies.com.

[10] Stem Inc., "Reducing demand charges," March 2015, [online] available at: http://goo.gl/ofv414.

[11] 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.

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

[13] R. Hermans, M. Almassalkhi, and I. Hiskens, "Incentive-based coordinated charging control of plug-in electric vehicles at the distribution-transformer level," in *American Control Conference (ACC)*, 2012, pp. 264–269.

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

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

[16] 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.

[17] S. Albers, "Competitive online algorithms," in *BRICS LS-96-2*, 1996.

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

[19] 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.

[20] B. Hodge and M. Milligan, "Wind Power Forecasting Error Distributions over Multiple Timescales," in *Power and Energy Society General Meeting*, Detroit, Michigan, July 2011.

[21] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power System*, vol. 29, no. 3, pp. 1033–1044, 2014.

[22] D. Saez, F. Avila, D. Olivares, C. Canizares, and L. Marin, "Fuzzy Prediction Interval Models for Forecasting Renewable Resources and Loads in Microgrids," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 548–556, 2015.

[23] A. Subramanian, M. Garcia, A. D. Garcia, D. Callaway, K. Poolla, and P. Varaiya, "Realtime Scheduling of Deferrable Electric Loads," in *American Control Conference (ACC)*, 2012.

[24] A. Charnes and W. W. Cooper, "Programming with linear fractional functions," vol. 9, no. 3–4, pp. 181–186, 1962.

[25] B. Colson, P. Marcotte, and G. Savard, "An Overview of Bilevel Optimization," *Annals of operations research*, vol. 153, no. 1, 2007.

[26] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[27] http://www.elia.be/en/grid-data/.

[28] U.S. Department of Transportation and Federal Highway Administration, "2009 National Household Travel Survey," http://nhts.ornl.gov.

[29] D. Wu, D. C. Aliprantis, and K. Gkritza, "Electric energy and power consumption by light-duty plug-in electric vehicles," *IEEE transactions on power systems*, vol. 26, no. 2, pp. 738–746, 2011.

[30] http://www.nissanusa.com/electric-cars/leaf/.

APPENDIX

A. Proof of Lemma 1

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

Suppose that the condition (5) holds for all $t_1 \leq t_2, t_1, t_2 \in \mathbb{T}$. We will show that the earliest-deadline-first policy can finish all the demand in Z before the corresponding deadlines.

We prove by contradiction. Suppose that some demand misses its deadline. Without loss of generality, we assume that this demand'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 the demand with deadline no later than d. It is easy to see that the time-slot d is always good.

If all the time-slots $t = 1, 2, ..., d-1$ are good, then there is no energy wasted[2] during the first d time-slots, and all the energy is used to serve demand with deadlines no later than d. Note that $\sum_{t=1}^{d}\sum_{s=t}^{d} a_{t,s} + \sum_{t=1}^{d} b_t \leq \sum_{t=1}^{d} E_t$. Then, all demand with deadline no later than d can be finished before the end of the time-slot 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 demand with deadline smaller or equal to d is served. Furthermore, all demand 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_b} could have been used to serve this part of demand according to the earliest-deadline-first policy.) Therefore, only demand with arrival time larger than t_b and deadline no later than d is served from $t_b + 1$ to d. On the other hand, we note that $\sum_{t=t_b+1}^{d}\sum_{s=t}^{d} a_{t,s} + \sum_{t=t_b+1}^{d} b_t \leq \sum_{t=t_b+1}^{d} E_t$. Then, all demand with deadline no later than d can be finished before the end of the time-slot d, which contradicts to our assumption again. ■

B. Proof of Lemma 3

Proof: Recall the definition (8) of $E_t^{pe}(Z_t)$. Since $E_{\text{off}}^*(Z')$ is nonnegative, $E_t^{pe}(Z_t)$ is also nonnegative, and thus is greater than $-\infty$. Then, for any $\epsilon > 0$, there must exist Z^*, such that

$$E_{\text{off}}^*(Z') < E_t^{pe}(Z_t) + \epsilon.$$

[2]If at some t, the available demand to serve is less than E_t, we will say that part of E_t is wasted.

Thus, in order for the online algorithm π to have a competitive ratio $\eta_Y(\pi)$ for input Z^*, its decision $E_t(Z_t, \pi)$ must satisfy

$$E_t(Z_t, \pi) \leq \eta_Y(\pi) E^*_{\text{off}}(Z') < \eta_Y(\pi)(E^{pe}_t(Z_t) + \epsilon).$$

Letting $\epsilon \to 0$, we immediately have

$$E_t(Z_t, \pi) \leq \eta_Y(\pi) E^{pe}_t(Z_t).$$

■

C. Proof of Lemma 5

Proof: We first show that $M_2 \geq M_1$. Let $\{M^n_1\}$ be an increasing sequence satisfying $M^n_1 > 0$ for any $n = 1, 2, ...,$ and $\lim_{n\to\infty} M^n_1 = M_1$. Since $M_1 > 0$ (this can be easily proved based on the conditions (a) and (b) of the lemma), such a sequence $\{M^n_1\}$ always exists. Recall that M_1 is the optimal value of the optimization problem in (10). Then, for any n, there exists $(\vec{x}_n, y_n)$ satisfying the constraints of (10), such that $(c^T\vec{x}_n + \alpha)/y_n > M^n_1$. Note that $y_n = f(\vec{x}_n) > 0$ (by condition (a) of this lemma). Let

$$\vec{x}' = \frac{\vec{x}_n}{y_n}, u = \frac{1}{y_n}.$$

Then, $(\vec{x}', u)$ satisfies the constraints in (11), and $c^T\vec{x}' + \alpha u = (c^T\vec{x}_n + \alpha)/y_n > M^n_1$. Noting that M_2 is the optimal value of (11), we must have $M_2 > M^n_1$. Letting $n \to \infty$, we then have $M_2 \geq M_1 > 0$.

We next show that $M_2 \leq M_1$. Since $M_2 > 0$, there must exist an increasing sequence $\{M^n_2\}$ satisfying $M^n_2 > 0$ for any $n = 1, 2, ...,$ and $\lim_{n\to\infty} M^n_2 = M_2$. Then, according to the definition of M_2, for any n, there exists $(\vec{x}'_n, u_n)$ satisfying the constraints of (11), such that $c^T\vec{x}'_n + \alpha u_n > M^n_2 > 0$. Note that $u_n > 0$. Let

$$\vec{x} = \frac{\vec{x}'_n}{u_n}, y = \frac{1}{u_n}.$$

Then, it is easy to check that $y \geq f(\vec{x})$ and $A\vec{x} \leq b$. Let $y_0 = f(\vec{x}) \leq y$. Then,

$$\frac{c^T\vec{x} + \alpha}{y_0} \geq \frac{c^T\vec{x} + \alpha}{y} = c^T\vec{x}'_n + \alpha u_n > M^n_2.$$

Noting that M_1 is the optimal value of (10), we must have $M_1 > M^n_2$. Letting $n \to \infty$, we then have $M_1 \geq M_2$.

Combining the above analysis, we then have $M_1 = M_2$. ■

D. Proof of Lemma 6

Proof: It suffices to show that for any x_1, x_2 and $x_0 = \lambda x_1 + (1-\lambda)x_2$ where $0 < \lambda < 1$,

$$g(x_0) \leq \lambda g(x_1) + (1-\lambda) g(x_2). \tag{21}$$

Recall that $g(x_1) = \inf_{y \in D_{x_1}} f(x_1, y)$ and $g(x_2) = \inf_{y \in D_{x_2}} f(x_2, y)$. Hence, for any fixed $\epsilon > 0$, there exist y_1, y_2, such that $f(x_1, y_1) < g(x_1) + \epsilon$ and $f(x_2, y_2) < g(x_2) + \epsilon$. Note that $f(x, y)$ is a convex function of (x, y). We then have,

$$\begin{aligned} & \lambda g(x_1) + (1-\lambda) g(x_2) \\ > \; & \lambda f(x_1, y_1) + (1-\lambda) f(x_2, y_2) - \epsilon \\ \geq \; & f(x_0, y_0) - \epsilon, \end{aligned} \tag{22}$$

where $y_0 = \lambda y_1 + (1-\lambda) y_2$. Since D is a convex set, if $(x_1, y_1), (x_2, y_2) \in D$, we must have $(x_0, y_0) \in D$. Therefore, $y_0 \in D_{x_0}$. Then, based on the definition of $g(x)$, we must have

$$g(x_0) = \inf_{y \in D_{x_0}} f(x_0, y) \leq f(x_0, y_0). \tag{23}$$

Combining Eqn. (22) and (23), we then have

$$\lambda g(x_1) + (1-\lambda) g(x_2) > g(x_0) - \epsilon.$$

Letting $\epsilon \to 0$, we then obtain (21). Thus, $g(x)$ is a convex function. ■

E. Trace

We first describe the traces for the EV demand and the net non-deferrable load, which we will use for evaluation. We obtain a synthesized EV demand pattern based on the National Household Travel Survey (NHTS) dataset [28] in Appendix E1. We then focus on the net non-deferrable load. Recall that the net non-deferrable load is equal to the background demand minus the renewable energy. Hence, we will show the statistics of both the background demand and the renewable energy in Appendix E2 and E3, respectively. We have used $\mathbf{b}$ to represent the net non-deferrable load. In Appendix E2 and E3, we will use B and R to represent the background demand and the renewable energy, respectively.

1) EV Demand: We first estimate the future EV demand based on the National Household Travel Survey (NHTS) dataset [28]. This dataset provides the transportation record for 150147 households. Even though the vehicles in the NHTS dataset are mainly gasoline- or diesel- fueled vehicles, we assume that people's driving behavior will not change much after they upgrade their vehicles to EVs.

Human's driving behavior exhibits a strong diurnal pattern. According to the NHTS dataset, many drivers leave home at around 8am in weekdays. Thus, if all EVs are charged overnight at home, the EV-charging deadlines should be earlier than 8am the next day. Recall that our system model in Section II assumes a finite time-horizon. Due to the above considerations, we take the decision horizon as a day (24 hours) from 8am today to 8am tomorrow.

Using the NHTS dataset, we collect the starting time t_{start} for each overnight EV-charging job based on the time that a vehicle comes back home, collect the charging deadline t_{end} based on the time that the vehicle leaves home, and estimate the charging demand based on the mileage that the vehicle travels in a day. Specifically, based on the NHTS dataset, we can obtain the exact time that a vehicle arrives home. We only care about the last arrival-time that the vehicle arrives home in a day from $0:00$ to $24:00$. (By the "last arrival-time", we mean that the vehicle will not leave home in the same day.) If this "last arrival-time" is from $0:00$ to $8:00$ (based on the NHTS dataset, only 0.19% of the vehicles have such "last arrival-times"), we will set $t_{\text{start}} = 8:00$; otherwise, we will set t_{start} to be exactly equal to the "last arrival-time". The NHTS dataset also provides the time that a vehicle leaves home. Here, we only care about the first leaving-time that the vehicle leaves home in a day from $0:00$ to $24:00$. If this "first leaving-time" is from $8:00$ to $24:00$, we will set $t_{\text{end}} = 8:00$; otherwise, we will set t_{end} to be exactly equal to the "first leaving-time". Such approximation is reasonable due to the following reasons. First, a driver may not know the exact leaving time in the next day when an EV starts charging. Therefore, he/she has to set t_{end} more conservatively to ensure that the EV can be fully charged before leaving home the next day. Second, the "busy hours" for background demand start around $8:00$ (see Fig. 8 (b)) at each day. Therefore, further deferring the charging deadlines may not help much in reducing the overall peak. We can also estimate the total mileage that a vehicle travels in a day from $0:00$ to $24:00$ based on the NHTS dataset. On average, the energy consumption of an EV is 34kwh/100miles [30]. Therefore, we use "*total mileage*$\times 0.34$" to compute an estimation of the charging demand of each EV.

We discretize the t_{start} and the t_{end} into 48 time slots, each of which lasts for half an hour. For each 2-tuple (t_{start},t_{end})=(i,j), we compute the total EV demand $a^{\text{trace}}_{i,j}$ by summing up all the demand with (t_{start},t_{end}) =(i,j). Finally, we plot the EV demand $\mathbf{a}^{\text{trace}}$ as a function of t_{start} and t_{end} in Fig. 2. From Fig. 2, we can see that the peak of t_{start} occurs at around 6pm.

2) Background Demand: We obtain the background demand and its day-ahead forecast from Elia [27], Belgium's electricity transmission system operator. We plot the revealed background demand and the day-ahead predicted background demand of one month in Fig. 8 (a). Recall that we have divided a day into 48 time slots. Let $B^{\text{real}}_t(k)$ be the realized background load at time slot t of the k-th day; let $B^{\text{day-ahead}}_t(k)$ be the day-ahead predicted background load at time slot t of the k-th day. Even though the mean prediction error, i.e.,

$$E_{t,k}\left[|B^{\text{real}}_t(k) - B^{\text{day-ahead}}_t(k)|/B^{\text{day-ahead}}_t(k)\right],$$

is small[3] (5.67%), the prediction error in a specific day can be very large. For example, on 01/03/2013, the prediction error is always above 10% (Fig. 8 (b)).

We are interested in the range of the prediction error. Let $e^{\text{day-ahead}}_B(t,k)$ be the day-ahead prediction error at time slot t of the k-th day, i.e.,

$$e^{\text{day-ahead}}_B(t,k) = \frac{B^{\text{real}}_t(k) - B^{\text{day-ahead}}_t(k)}{B^{\text{day-ahead}}_t(k)}.$$

For each fixed time-slot t, we have different $e^{\text{day-ahead}}_B(t,k)$'s for different days. In Fig. 8 (c), we plot the 95th percentile and the 5th percentile values of $e^{\text{day-ahead}}_B(t,k)$ for each fixed t. These percentile values can be then used to set the day-ahead prediction bounds as required by our system model (see Appendix F). Another observation here is that the day-ahead prediction is highly unbalanced. The realized background load tends to be higher than the day-ahead predicted value.

We are also interested in the slot-ahead prediction error. (Recall that one slot is equal to half an hour, since we have 48 slots in a day.) The Elia trace does not provide slot-ahead predictions. Here, we simply use the realized background load as a slot-ahead prediction of the background load in the next time slot, i.e., the slot-ahead prediction $B^{\text{slot-ahead}}_t(k) = B^{\text{real}}_{t-1}(k)$. Similarly, we can define the slot-ahead prediction error $e^{\text{slot-ahead}}_B(t,k)$ at time slot t of the k-th day as follows,

$$e^{\text{slot-ahead}}_B(t,k) = \frac{b^{\text{real}}_t(k) - b^{\text{slot-ahead}}_t(k)}{b^{\text{day-ahead}}_t(k)}.$$

We caution that the slot-ahead prediction errors are also with respect to the day-ahead predicted values. We plot the 95th percentile and the 5th percentile values of $e^{\text{slot-ahead}}_B(t,k)$ for each fixed t in Fig. 8 (c). We can see that the slot-ahead prediction is much more accurate than the day-ahead prediction. Further, these percentile values can be used to set

[3]Here, $E_{t,k}[\cdot]$ means taking the average over all possible t's and k's.

the intra-day prediction as required by our system model (see Appendix F).

3) Renewable Energy: We obtain the renewable energy and its day-ahead forecast from Elia [27]. The renewable energy is from wind power. The total installed capacity of the wind power is $R_{\text{capacity}} = 930.65$MW.

We plot the realized renewable energy and the day-ahead predicted value of one month in Fig. 9 (a). We can see that the renewable energy is highly variable.

Similar to the background load, we use $R_t^{\text{real}}(k)$ to denote the realized wind energy at time slot t of the k-th day, and use $R_t^{\text{day-ahead}}(k)$ to denote the day-ahead predicted wind energy at time slot t of the k-th day. We also use the revealed wind energy as a slot-ahead prediction of the renewable energy in the next time slot, i.e., the slot-ahead prediction $R_t^{\text{slot-ahead}}(k) = R_{t-1}^{\text{real}}(k)$. we also define the day-ahead prediction error and the slot-ahead prediction error for the renewable energy as follows.

$$e_R^{\text{day-ahead}}(t,k) = (R_t^{\text{real}}(k) - R_t^{\text{day-ahead}}(k))/R_{\text{capacity}},$$
$$e_R^{\text{slot-ahead}}(t,k) = (R_t^{\text{real}}(k) - R_t^{\text{slot-ahead}}(k))/R_{\text{capacity}}.$$

We caution here that the prediction errors for the renewable energy are with respect to the total installed capacity, while the prediction errors for the background load are with respect to the day-ahead predicted values. This is because that the renewable energy is more variable, and does not exhibits any predictable diurnal pattern. We plot the 95th percentile and the 5th percentile values of $e_R^{\text{day-ahead}}(t,k)$ and $e_R^{\text{slot-ahead}}(t,k)$ for each fixed t in Fig. 9 (b). We can see that the slot-ahead predictions are more accurate than the day-ahead predictions. Further, these percentile values can be used to set the day-ahead prediction bounds and the intra-day prediction as required by our system model (see Appendix F).

F. Simulation Parameter Setting

We have synthesized an EV demand pattern from [28], and have obtained the day-ahead predicted values and the real values of both the background demand and the renewable energy from [27]. However, these data sets are not ready to use in our numerical experiments. Recall our system model (Section II) that the pre-known knowledge Y of the aggregator includes both the day-ahead prediction bounds ($\hat{\mathbf{x}}^L(0,t)$ and $\hat{\mathbf{x}}^U(0,t)$) and the intra-day prediction gap ($W(u_t,t)$). In the following, we will show how to generate the required data set of Y using the data sets available in [28] and [27].

We first focus on the EV demand. We use the synthesized EV demand (see Fig. 2), denoted by $\mathbf{a}^{\text{day-ahead}}$, as the day-ahead prediction of the EV demand. We use a day-ahead prediction error ϵ_a to generate the day-ahead upper and lower bounds, i.e.,

$$\hat{\mathbf{x}}_a^L(0,t) = \mathbf{a}^{\text{day-ahead}} \times (1-\epsilon_a), \hat{\mathbf{x}}_a^U(0,t) = \mathbf{a}^{\text{day-ahead}} \times (1+\epsilon_a),$$

where $\hat{\mathbf{x}}_a^L(0,t)$ ($\hat{\mathbf{x}}_a^U(0,t)$) represents all the entries in $\hat{\mathbf{x}}^L(0,t)$ ($\hat{\mathbf{x}}^U(0,t)$) that correspond to the EV demand. We assume that there is no intra-day prediction for the EV demand. Therefore, we set

$$W_a(u_t,t) = \infty,$$

where $W_a(u_t,t)$ represents all the entries in $W(u_t,t)$ that correspond to the EV demand.

We then focus on the net non-deferrable load. From the day-ahead prediction errors ($e_B^{\text{day-ahead}}(t,k)$ and $e_R^{\text{day-ahead}}(t,k)$) of both the background load and the renewable energy (see Appendix E), we first find their 95th percentile and 5th percentile values, which are denoted by $e_{B,95}^{\text{day-ahead}}(t)$, $e_{B,5}^{\text{day-ahead}}(t)$, $e_{R,95}^{\text{day-ahead}}(t)$ and $e_{R,5}^{\text{day-ahead}}(t)$, respectively. Then, the day-ahead prediction bounds for the net non-deferrable load $\mathbf{b} = B - R$ can be set as follows:

$$\begin{aligned}\hat{\mathbf{x}}_b^L(0,t) &= B_t^{\text{day-ahead}} \times (1 + e_{B,5}^{\text{day-ahead}}(t)) \\ &\quad -\big(R_t^{\text{day-ahead}} + R_{\text{capacity}} \times e_{R,95}^{\text{day-ahead}}(t)\big),\end{aligned}$$

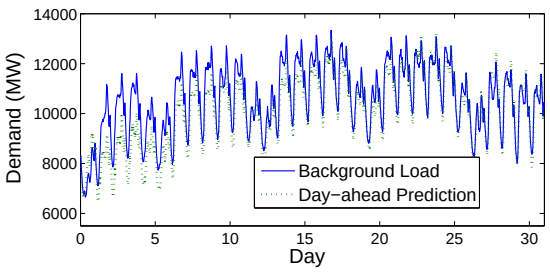
$$\begin{aligned}\hat{\mathbf{x}}_b^U(0,t) &= B_t^{\text{day-ahead}} \times (1 + e_{B,95}^{\text{day-ahead}}(t)) \\ &\quad -\big(R_t^{\text{day-ahead}} + R_{\text{capacity}} \times e_{R,5}^{\text{day-ahead}}(t)\big),\end{aligned}$$

where $\hat{\mathbf{x}}_b^L(0,t)$ ($\hat{\mathbf{x}}_b^U(0,t)$) represents all the entries in $\hat{\mathbf{x}}^L(0,t)$ ($\hat{\mathbf{x}}^U(0,t)$) that correspond to the net non-deferrable load.

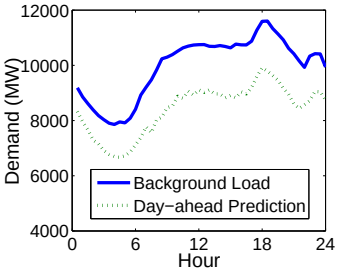
As for the intra-day prediction, we assume that the intra-day prediction is conducted 1-slot ahead. (Recall from Appendix E that we use the revealed background demand or wind energy in time-slot $t-1$ as a slot-ahead prediction of the background demand or wind energy in time-slot t.) Hence, $u_t = t-1$. From the slot-ahead prediction errors ($e_B^{\text{slot-ahead}}(t,k)$ and $e_R^{\text{slot-ahead}}(t,k)$) of both the background load and the renewable energy (see Appendix E), we can find their 95th percentile and 5th percentile values, which are denoted by $e_{B,95}^{\text{slot-ahead}}(t)$, $e_{B,5}^{\text{slot-ahead}}(t)$, $e_{R,95}^{\text{slot-ahead}}(t)$ and $e_{R,5}^{\text{slot-ahead}}(t)$, respectively. Then, we set the intra-day prediction gap as follows:

$$\begin{aligned}W_b(u_t,t) &= B_t^{\text{day-ahead}} \times \big(e_{B,95}^{\text{slot-ahead}}(t) - e_{B,5}^{\text{slot-ahead}}(t)\big) \\ &\quad R_{\text{capacity}} \times \big(e_{R,95}^{\text{day-ahead}}(t) - e_{R,5}^{\text{day-ahead}}(t)\big),\end{aligned} \quad (24)$$

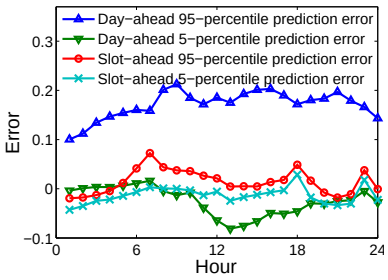
where $W_b(u_t,t)$ represents all the entries in $W(u_t,t)$ that correspond to the net non-deferrable load.



(a) Background load vs. its day-ahead forecast (Jan. 2013).

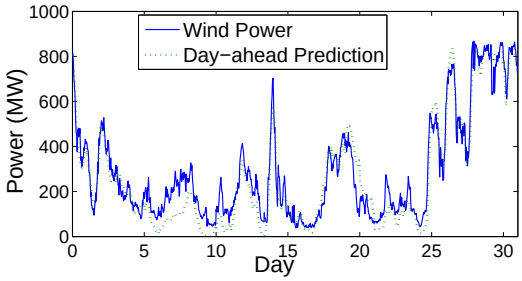


(b) Background load vs. its day-ahead forecast (01/03/2013).

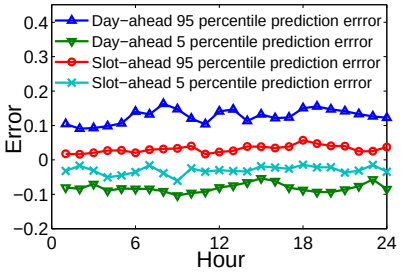


(c) Prediction errors computed from a one-month (Jan. 2013) trace.

Fig. 8. Background demand and its forecast.



(a) Wind power vs. its day-ahead forecast (Jan. 2013).



(b) Prediction errors computed from a one-month (Jan. 2013) trace.

Fig. 9. Wind power and its forecasts.