Peak-Minimizing Online EV Charging: Price-of-Uncertainty and Algorithm Robustification

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 competitive online 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 computation 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 [1]. 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 [2]. 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, which is a typical example of deferrable demands [3]. 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 intelligent scheduling algorithms for EV charging that minimizes the impact of variability and uncertainty of renewable energy to the grid. Specifically, we consider an (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 [4]. 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 $c_1 \sum_i E_i + c_2 \max_i \{E_i\}$ [5]. In typical schemes (e.g., Wisconsin electric power company [6]), the unit charge for peak usage c_2 (between 9.03\$/kW and 9.38\$/kW) is approximately 200 times the unit charge for total usage c_1 (between 0.03\$/kWh and 0.05\$/kWh), thus giving the customers a strong incentive to reduce the peak. 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., [4]), the peak energy is usually in the order of 20MW. Then, every one percent of peak reduction will correspond to $0.01 \times 20\text{MW} \times 9\$/\text{kW} \times 12 = 21600\$$ saving per year.

However, designing good scheduling algorithms for EV-charging that minimize the peak demand to the grid is a challenging problem due to the inherent uncertainty in both the demand and the renewable energy 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 [7]. Unfortunately, in practice such prediction is often quite inaccurate. For example, the maximum day-ahead prediction error for wind can be above 20% [8]. Without accounting for such uncertainty, the resulting peak could be much higher than what we can achieve. As readers will see in the numerical results in Section V-A, an algorithm that is oblivious to such inherent uncertainty will likely lead to significantly larger peak consumption levels.

In the literature, there are two general technical approaches to deal with uncertainty. The first approach is to assume a probabilistic model for future uncertainty and cast the problem as a stochastic decision control problem. This is the approach taken, e.g., in risk limiting dispatch [9], where a two-stage stochastic control problem was studied. In contrast, since the amount of renewable generation is revealed sequentially, here we are faced with a multi-stage stochastic control problem. However, this stochastic-control approach is known to have a number of difficulties. First, as the problem size increases, a multi-stage stochastic control problem quickly becomes computationally intractable [10]. Second, even obtaining the probabilistic model of uncertainty can be challenging, especially when the renewable supply is non-stationary and highly-correlated across time. If the probabilistic model is inaccurate, the resulting performance guarantee also becomes questionable.

The second approach, which we will adopt in this paper, is to model the uncertainty in a set, and develop algorithms that can achieve provable performance guarantees for the worst-case uncertainty within that set. Note that there is no need to obtain a probabilistic model. Further, as we will show in the rest of the paper, searching for the worst case could potentially be more tractable even for fairly complicated problem settings. This approach is related to robust optimization [11] and two-stage adaptive robust control [12]. As we discussed earlier, due to the sequential nature with which renewable generation is revealed, here we are interested in multi-stage decisions. This multi-stage problem is also closely related to the problem of designing competitive online algorithms [13]. For example, it was shown in [14] 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 are often based on very simple models of future uncertainty [15], or do not assume any model at all. As a result, the worst-case performance (and the corresponding competitive ratio) is 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 complicated models of future uncertainty. The second challenge, which in fact applies to many "robust optimization" results as well [12], is that the algorithms are only optimized for the worst-case. As a result, their average-case performance can be quite poor [15]. Given that the worst-case input may only occur very rarely, the aggregator may then be hesitant to endorse the resulting algorithm.

In this paper, we make two contributions to address the above challenges. First, we propose a very general model, called 2-IPM (2-level increasing precision model), to capture the uncertainty of predicting renewable energy, EV demand, and background load. Compared to existing uncertainty models, e.g. [15], a key novelty of 2-IPM is that it can model the scenario where predictions are made at multiple instants (e.g., day-ahead prediction versus intra-day prediction), and that 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 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 "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 [15].

We then study online algorithms that attain the optimal competitive ratio under the 2-IPM. One can easily generalize the EPS algorithm in [15] to obtain an online algorithm with the optimal competitive ratio. However, such generalized EPS algorithm still suffers from the second weakness discussed earlier, i.e., it is optimized for the worst-case input, and its average-case performance can be quite poor. Our second contribution is to propose a general "robustification" procedure to design online algorithms with both the optimal competitive ratio and good average-case performance. 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.

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. Using the flexibility in the EV demand, the goal of the aggregator is to schedule EV charging jobs 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 value 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

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

Then, at a later time $u_t, 1 \le u_t < t$, 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:

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

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 predictioin, $\hat{\mathbf{x}}^L(0,t), \hat{\mathbf{x}}^U(0,t)$ 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 $W(u_t,t) \le \hat{\mathbf{x}}^U(0,t) - \hat{\mathbf{x}}^L(0,t)$, *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.,

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

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: Readers may question what happens when the bounds for the intra-day prediction falls outside of the day-ahead predicted interval (1). If this happens, we suggest tighten the intra-day prediction bounds so that (2) is satisfied. This procedure 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. In other words, the competitive online algorithms defined below implicitly ignore those cases where these bounds are violated.

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 $s \le t$, and the intra-day prediction for any time-slot s such that $u_s \le t$. This set of information is summarized in $Z_t = \{\mathbf{x}(s), s = 1,2,...,t\} \bigcup \{\hat{\mathbf{x}}^L(u_s,s), \hat{\mathbf{x}}^U(u_s,s), u_s \le t\}$. 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 $Z \setminus Z_t$.

B. Objective

We are interested in designing online algorithms for scheduling EV demand that minimize the peak energy drawn from the grid. Since Y is known day-ahead (before any scheduling decisions are made), we define our objectives for a fixed Y as follows. For a fixed Y, any possible realization Z must be in the following set: $\mathcal{Z}_Y = \{Z : Y, Z \text{ satisfy } (1)-(3)\}$.

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*. 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:

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

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.

III. Fundamental Limit of the Competitive Ratio

In this section, we extend the computation framework in [15] 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 [15]. In Section III-B, we will develop a general convexification technique to convexify (9). Such a convexification technique generalizes fractional-linear programs [16], and thus may be of independent interest.

We need the following two lemmas throughout this section.

Lemma 1: An online algorithm π is feasible if and only if for all $Z \in \mathcal{Z}_Y$ and all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$, the following inequality holds,

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

Proof: See Appendix A. ■

Lemma 1 is a generalization of Lemma 6 in [15]. It states that, in order for an online algorithm 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 (22) is also sufficient. Specifically, if the service profile of the algorithm π satisfies (22), and the algorithm π uses the Earliest-Deadline-First (EDF) policy to serve the demand, then this algorithm π can finish all the demands before their corresponding deadlines.

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

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

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. This result is easy to show based on the offline optimal algorithm proposed in [7]. The reason that we have a "0" term in Eqn. (7) is that the power procured from the grid must be non-negative.

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 the competitive ratio $\eta_Y(\pi)$ holds for all the possible future uncertainty. 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.

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

To obtain Lemma 3, let Z^* be the realization that attains the minimum value of $E^*_{\text{off}}(Z')$ in (8). Then, the optimal offline peak would be $E_t^{pe}(Z_t)$ if the future realization turned out to be Z^*. Thus, in order for π to have a competitive ratio $\eta_Y(\pi)$ for input Z^*, its decision $E_t(Z_t, \pi)$ cannot exceed $\eta_Y(\pi) E_t^{pe}(Z_t)$.

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 [15]. However, computing η^*_Y here is much more difficult than that in [15]. The computation of η^*_Y requires solving the optimization problem (9). Like in [15], 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 [17]. In [15], 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 [15] critically rely on the property that the input Z can be freely scaled up or down without violating system constraints. Unfortunately, this property does not hold in this paper. Specifically, if 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 [15] 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 $\vec{x}$, 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 2: 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 $1 \geq uf(\vec{x}'/u)$. 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 $f(\cdot)$ [18]. 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 [16], which requires $f(\vec{x})$ to be linear. The detailed proof is available in Appendix B.

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 C. ■

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.

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 [15]. 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 [15], 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 (22) 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 [13]. 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.,

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

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

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

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)). Then, we have

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

Note that (13) must hold for all possible future inputs. Define the following optimization problem that maximizes the right hand side of (13) over all possible future inputs:

$$\sup_{Z' \in \mathcal{Z}_Y, Z'_t = Z_t} \sum_{s=t+1}^{t_1}\left(\sum_{w=s}^{t_1} a'_{s,w} + b'_s - \eta_Y^* E_s^{pe}(Z'_s)\right) \quad (14)$$

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 (14). Then, in order to attain the optimal competitive ratio, the following must hold

$$E_t(Z_t, \pi) \geq r_{t,t_1} + R^*_{\eta^*_Y}(Z_t, t_1). \tag{15}$$

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

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

Remark 3: Note that $E^{pe}_s(Z'_s)$ is a convex function (see Section III-B). Therefore, the objective of (14) is a concave function. Further, both constraints ($Z' \in \mathcal{Z}_Y$ and $Z'_t = Z_t$) of (14) are linear constraints. Hence, (14) 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

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

Remark 4: 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 the 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.
Compute the lower bound (16) and upper bound (12),
and let $E_t(Z_t, \pi)$ be any value in between.
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 (16) 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

$$\max_{t_1 \geq t} \{r_{t,t_1} + R^*_{\eta^*_Y}(Z_t, t_1)\} \leq \eta^*_Y E^{pe}_t(Z_t). \tag{17}$$

Lemma 8 is the key of this section, and its proof is nontrivial. We can see that both sides of (17) depend on η^*_Y. In fact, η^*_Y is the smallest value such that (17) always holds. For any $\eta < \eta^*_Y$, it is possible to construct a case $Z' \in \mathcal{Z}_Y$ such that $\max_{t_1 \geq t}\{r_{t,t_1} + E^*_\eta(Z_t, t_1)\} > \eta E^{pe}_t(Z_t)$ for some t.

In order to prove Lemma 8, it suffices to show that

$$\eta^*_Y E^{pe}_t(Z_t) \geq \text{right hand side of (13)} \tag{18}$$

for all possible future realizations of the input. We note that r_{t,t_1} in (13) can be written as $r_{t,t_1} = \tilde{r}_{t,t_1} + b_t + \sum_{s=t}^{t_1} a_{t,s}$, where $\tilde{r}_{t,t_1}$ is the total remaining demand (with deadline no greater than t_1) from the previous time slots. Then, we can rewrite the inequality (18) as follows:

$$\eta^*_Y \sum_{s=t}^{t_1} E^{pe}_s(Z_s) \geq \tilde{r}_{t,t_1} + \sum_{s=t}^{t_1} \left(b_s + \sum_{w=s}^{t_1} a_{s,w} \right). \tag{19}$$

The main difficulty of proving (19) comes from the term "$\tilde{r}_{t,t_1}$". If $\tilde{r}_{t,t_1} = 0$, then (19) would have hold trivially because of the definition of η^*_Y (see (9)). If $\tilde{r}_{t,t_1} > 0$, we have to develop new techniques to prove (19). The key idea here is to apply induction from time $t-1$ to time t. The detailed proof is in Appendix D.

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 (15). 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.

In practice, in Section V-C, we will robustify a well-known online algorithm, called receding-horizontal-control (RHC).

```
Input: A realization $Z \in \mathcal{Z}_Y$, the optimal competitive
       ratio $\eta_Y^*$ and any online algorithm $\pi$.
Output: An optimal online algorithm $\pi_{\text{robust}}$ and its
        schedules $E_t(Z_t, \pi_{\text{Robust}})$.
for $t = 1 : T$ do
    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 $\pi$.
    Set $E_t(Z_t, \pi_{\text{Robust}}) = M_\alpha^\beta(E_t(Z_t, \pi))$, where
    $M_\alpha^\beta(x) = \max\{\min\{x, \beta\}, \alpha\}$.
end
```

Algorithm 2: Algorithm-Robustification Procedure

The RHC algorithm usually exhibits good average-case performance [19]. However, its worst-case competitive ratio can be very poor. 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.

V. Simulation

We conduct simulation using real traces from two data sets. Elia [20], 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 [20] does not provide data for intra-day prediction.) The National Household Travel Survey (NHTS) dataset [21] 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 [21] to synthesize a model for the EV demand (see Fig. 1), including EV arrival time, deadline and amount of energy charging demand, as has been done in earlier works in [22]. See Appendix E for more details about the traces in [20] and [21].

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 [20] 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 [20], we can then generate the upper and lower bounds for day-ahead predictions as required in our model (see Appendix F). In Fig. 2 (a), we apply 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 [20], and the upper and lower bounds of the real-time values as constructed above. From Fig. 2 (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 [20] 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. 1) 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 [20]) 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. 2 (b), where we plot the energy procurement schedule of 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. 1). 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 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 artificially 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 [15]. Note that the uncertainty model in [15] assumes that the ratio between the future uncertainty (i.e., the walk-in demand in [15]) and the predicted value (i.e., the reserved demand in [15]) is bounded. However, the absolute quantity of the predicted value is not specified.

[1] This EV trace [21] 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.

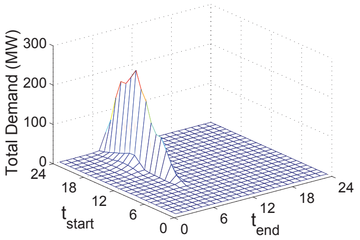
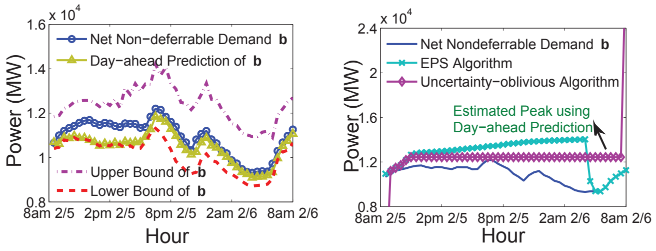


Fig. 1. Synthesized EV demand [21]. (t_{start}: arrival time of an EV; t_{end}: departure time of an EV.)



(a)

(b)

Fig. 2. The EPS algorithm vs. the uncertainty-oblivious algorithm.

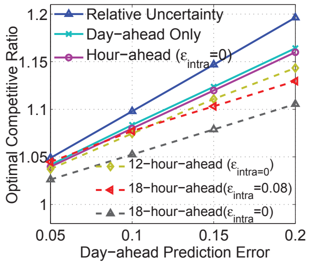
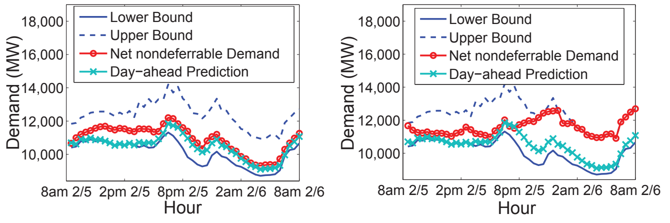


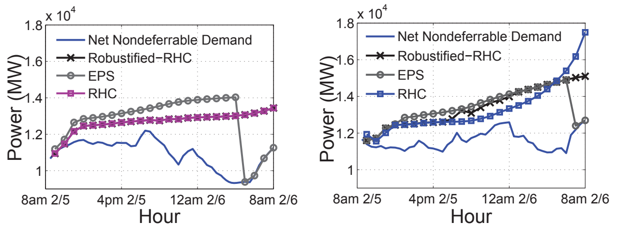
Fig. 3. Price of Uncertainty.



(a) Easy trace.

(b) Difficult trace.

Fig. 4. Net non-deferrable load of two Simulation traces.



(a) Easy trace.

(b) Difficult trace.

Fig. 5. Schedules under 2 Simulation traces.

Thus, we refer to the uncertainty model in [15] 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 [15] 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 [15], 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 in Section V-A, but varies the upper/lower bounds of day-ahead prediction by varying ϵ in (20). In Fig. 3, we plot the optimal competitive ratios under both 2-IPM and under the relative uncertainty model from [15], 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 $0.01\times 20\text{MW}\times 9\$/\text{kW}\times 12 = 21600\$$ saving per year for campus-level aggregators [4] 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 [15].

We next evaluate the impact of intra-day prediction. Note that the Elia data set [20] 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 must 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. 3, 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. 3 will allow the operator to evaluate which type of intra-day prediction will be most useful, i.e., in reducing the cost of uncertainty.

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) [19]. In our setting, RHC means that, at each time-slot, the aggregator assumes that *future demand and supply will be exactly equal to their most-recently predicted values*, and then computes the schedule that minimizes the future peak based on the remaining EV demand and the currently-known background demand and renewable energy supply. The aggregator will then apply the *first* time-slot of the schedule. In the next time-slot, this procedure is repeated with the newly-revealed information. 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. However, it is not difficult to construct cases where the RHC algorithm will perform much poorer than the optimal competitive ratio achieved by the EPS algorithm. (Details of such an example are available in Appendix G.)

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. 4). 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. 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 Appendix F for more details). Further, they use the same EV traces as in Section V-A, although here we also allow prediction errors of the EV demand. Specifically, we use EV demand model in Section V-A 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. 4 (a), the revealed values of the net non-deferrable demand are closer to their day-ahead predicted values, while in Fig. 4 (b), the difference is much bigger (particularly at the end of the time-horizon). We will also refer to the trace in Fig. 4 (a) as the "easy trace", and the trace in Fig. 4 (b) as the "difficult trace".

In Fig. 5, we compare the schedules of the EPS algorithm, the RHC algorithm and the robustified-RHC algorithm under both traces. By comparing Fig. 5 (a) and 5 (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.

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.

Acknowledgements

This work has been partially supported by the NSF through grant CCF-1442726 and the University Grants Committee of the Hong Kong Special Administrative Region, China (Theme-based Research Scheme Project No. T23-407/13-N).

References

[1] http://www.cpuc.ca.gov/PUC/energy/Renewables/overview.htm.

[2] T. Mai, D. Sandor, R. Wiser, and T. Schneider, "Renewable Electricity Futures Study Executive Summary," http://www.nrel.gov/docs/fy13osti/52409-ES.pdf.

[3] 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.

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

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

[6] http://www.we-energies.com/pdfs/etariffs/michigan/emi_sheet49014-49020.pdf.

[7] 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.

[8] International Electrotechnical Commission, "Grid integration of large-capacity Renewable Energy sources and use of large-capacity Electrical Energy Storage," oct. 2012.

[9] P. P. Varaiya, F. F. Wu, and J. W. Bialek, “Smart Operation of Smart Grid: Risk-Limiting Dispatch,” *Proceedings of the IEEE*, vol. 99, no. 1, pp. 40–57, 2011.

[10] A. Shapiro and A. Nemirovski, “On complexity of stochastic programming problems,” in *Continuous optimization*, 2005, pp. 111–146.

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

[12] 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.

[13] S. Albers, “Competitive online algorithms,” in *BRICS LS-96-2*, 1996.

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

[15] 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.

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

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

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

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

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

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

[22] 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.

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

APPENDIX

A. *Proof of Lemma 1*

Proof: We would like to apply Lemma 6 of [15]. Note that [15] deals with a system without renewable energy and background load. Hence, in order to utilize Lemma 6 of [15], we first construct another system with only EV demand (which we call Π_2), by mapping the renewable energy and background load of the original system to a "fictitious" EV demand.

Let the original system be denoted by Π_1. We construct another EV-charging system Π_2 with EV demand only based on Π_1. The demand of Π_2 can be fully represented by a $T \times T$ upper-triangular matrix $\tilde{\mathbf{a}} = [\tilde{a}_{i,j}]$. For any $\tilde{\mathbf{a}} \in \tilde{\mathcal{Z}}$, there exists $Z \in \mathcal{Z}_Y$, such that

$$\tilde{a}_{i,j} = \begin{cases} a_{i,j}, \text{ if } i < j, \\ a_{i,j} + b_i, \text{ if } i = j. \end{cases} \tag{21}$$

Note that in (21), we have viewed the net non-deferrable demand b_i as a "fictitious" EV demand arriving at the beginning of the time-slot i and departing at the end of the time-slot i. As a result, the total EV demand $\tilde{a}_{i,i}$ with the same arrival time and deadline i of the system Π_2 equals the sum of the "real" EV demand $a_{i,i}$ and the "fictitious" EV demand b_i of the system Π_1. We denote the set of all possible realizations of $\tilde{\mathbf{a}}$ by $\tilde{\mathcal{Z}}$, i.e.,

$$\tilde{\mathcal{Z}} = \{\tilde{\mathbf{a}} : \text{there exists } Z \in \mathcal{Z}_Y, \text{such that (21) holds}\}.$$

We apply the same schedule of Π_2 to Π_1. Specifically, given $\tilde{\mathbf{a}} \in \tilde{\mathcal{Z}}$, we first find the corresponding $Z \in \mathcal{Z}_Y$ such that (21) holds. Then, at each time t, we can compute $E_t(Z_t, \pi)$ as if we were making decisions for the system Π_1, and then procure $E_t(Z_t, \pi)$ amount of energy to serve the EV demands in Π_2. It is easy to check that π is feasible for Π_1 if and only if for any $\tilde{\mathbf{a}} \in \tilde{\mathcal{Z}}$ in Π_2, the schedule obtained from the above procedure can finish all the EV demands in $\tilde{\mathbf{a}}$. Note that Π_2 only contains EV demand. Then, we can apply Lemma 6 in [15] to Π_2. Therefore, the schedule of π can finish all the demands in Π_2 if and only if for all $\tilde{\mathbf{a}} \in \tilde{\mathcal{Z}}$ and all $t_1 \le t_2, t_1, t_2 \in \mathbb{T}$, the following inequality holds,

$$\sum_{t=t_1}^{t_2} \sum_{s=t}^{t_2} \tilde{a}_{t,s} \le \sum_{t=t_1}^{t_2} E_t(Z_t, \pi). \tag{22}$$

Note that

$$\sum_{t=t_1}^{t_2} \sum_{s=t}^{t_2} a_{t,s} + \sum_{t=t_1}^{t_2} b_t = \sum_{t=t_1}^{t_2} \sum_{s=t}^{t_2} \tilde{a}_{t,s}.$$

The result of Lemma 1 then follows. ■

B. *Proof of Lemma 5*

Proof: We first show that $M_2 \ge M_1$. Recall that M_1 is the smallest upper bound of the optimization problem in (10). Then for any $\epsilon > 0$, there exists $(\vec{x}_\epsilon, y_\epsilon)$ satisfying the constraints of (10), such that $(c^T \vec{x}_\epsilon + \alpha)/y_\epsilon > M_1 - \epsilon$. Note that $y_\epsilon = f(\vec{x}_\epsilon) > 0$ (by condition (a) of the lemma). Let

$$\vec{x}' = \frac{\vec{x}_\epsilon}{y_\epsilon}, u = \frac{1}{y_\epsilon}.$$

Then, $(\vec{x}', u)$ satisfies the constraints in (11), and $c^T \vec{x}' + \alpha u = (c^T \vec{x}_\epsilon + \alpha)/y_\epsilon > M_1 - \epsilon$. The optimal value M_2 of (11) must be no smaller than the achievable value. Therefore, $M_2 > M_1 - \epsilon$. Let $\epsilon \to 0$, we then have $M_2 \ge M_1$. Further, according to the condition (b), it is easy to check that the optimal solution of (10) must be positive, i.e., $M_1 > 0$. Therefore, $M_2 \ge M_1 > 0$.

We next show that $M_2 \le M_1$. According to the definition of M_2, for any $0 < \epsilon < M_2$, there exists $(\vec{x}'_\epsilon, u_\epsilon)$ satisfying the constraints of (11), such that $c^T \vec{x}'_\epsilon + \alpha u_\epsilon > M_2 - \epsilon > 0$. Note that $u_\epsilon > 0$. Let

$$\vec{x} = \frac{\vec{x}'_\epsilon}{u_\epsilon}, y = \frac{1}{u_\epsilon}.$$

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

$$\frac{c^T \vec{x} + \alpha}{y_0} \ge \frac{c^T \vec{x} + \alpha}{y} = c^T \vec{x}'_\epsilon + \alpha u_\epsilon > M_2 - \epsilon.$$

The optimal value M_1 of (10) must be no smaller than the achievable value. Therefore, $M_1 > M_2 - \epsilon$. Let $\epsilon \to 0$, we then have $M_1 \ge M_2$.

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

C. *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 $0 < \lambda < 1$,

$$g(x_0) \le \lambda g(x_1) + (1-\lambda) g(x_2). \tag{23}$$

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,

$$\begin{aligned} & \lambda g(x_1) + (1-\lambda) g(x_2) \\ > \ & \lambda f(x_1, y_1) + (1-\lambda) f(x_2, y_2) - \epsilon \\ \ge \ & f(x_0, y_0) - \epsilon, \end{aligned} \tag{24}$$

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

$$g(x_0) = \inf_{y \in D_{x_0}} f(x_0, y) \le f(x_0, y_0). \tag{25}$$

Combining Eqn. (24) and (25), we then have

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

Let $\epsilon \to 0$, we then obtain (23). Thus, $g(x)$ is a convex function. ■

D. *Proof of Lemma 8*

Proof: Recall that $\max_{t_1 \ge t}\{r_{t,t_1} + R^*_{\eta^*_Y}(Z_t, t_1)\}$ is the smallest value of $E_t(Z_t, \pi)$ that satisfies (13) for all possible t_1's and all possible future realizations. Therefore, in order to prove (17), it suffices to show that $\eta^*_Y E^{pe}_t(Z_t)$ also satisfies

(13), i.e.,

$$\eta_Y^* \sum_{s=t}^{t_1} E_s^{pe}(Z_s) \geq r_{t,t_1} + \sum_{s=t+1}^{t_1} \left(\sum_{w=s}^{t_1} a_{s,w} + b_s \right), \qquad (26)$$

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 (26) 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), (26) holds trivially for all $t_1 \geq t$ when $t = 1$.

Assume that (26) holds for a given t and all $t_1 \geq t$. We will show that (26) 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 (26) 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 (14). 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, (26) holds for $t+1$ and all $t_1 \geq t+1$. By induction, (26) holds for all t's and $t_1 \geq t$. Thus, Lemma 8 holds. ■

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 [21] 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 [21]. 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. 7 (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 [23]. 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. 6. From Fig. 6, we can see that the peak of t_{start} occurs at around 6pm.

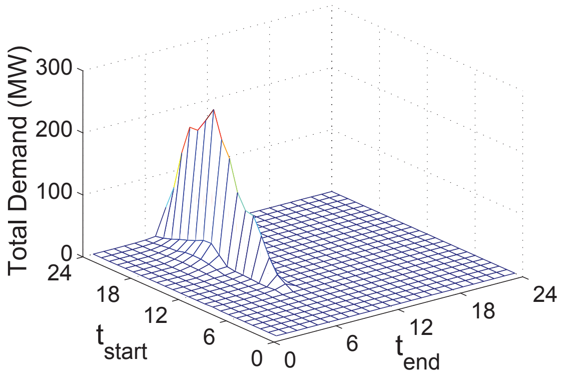


Fig. 6. EV demand.

2) Background Demand: We obtain the background demand and its day-ahead forecast from Elia [20], Belgium's electricity transmission system operator. We plot the revealed background demand and the day-ahead predicted background demand of one month in Fig. 7 (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[2] (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. 7 (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.,

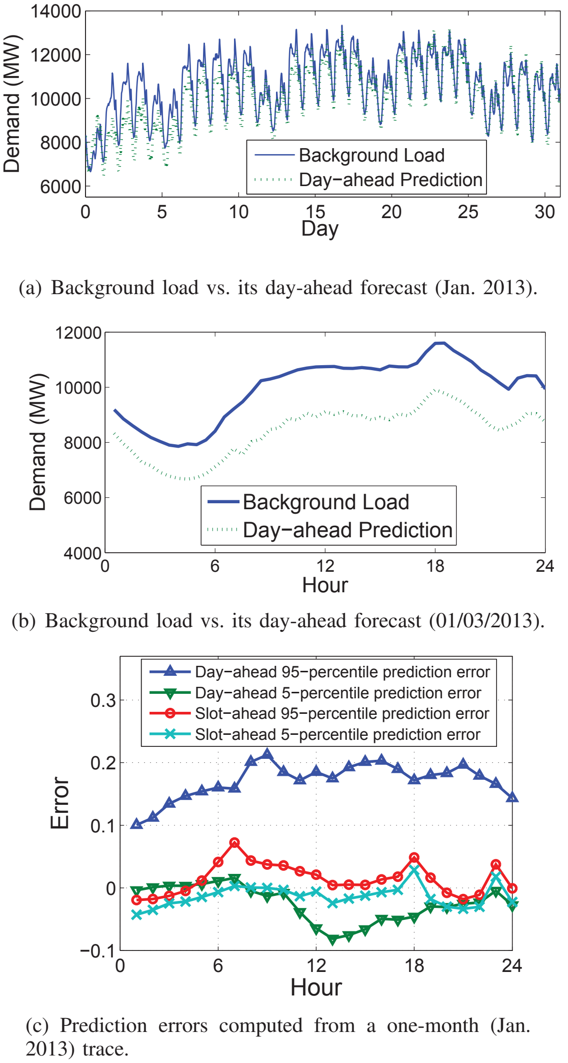
$$e^{\text{day-ahead}}_B(t, k) = \frac{B^{\text{real}}_t(k) - B^{\text{day-ahead}}_t(k)}{B^{\text{day-ahead}}_t(k)}.$$

For each fixed time-slot t, we have different $e^{\text{day-ahead}}_B(t, k)$'s for different days. In Fig. 7 (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. 7 (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 the intra-day prediction as required by our system model (see Appendix F).



(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. 7. Background demand and its forecast.

3) Renewable Energy: We obtain the renewable energy and its day-ahead forecast from Elia [20]. The renewable energy

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

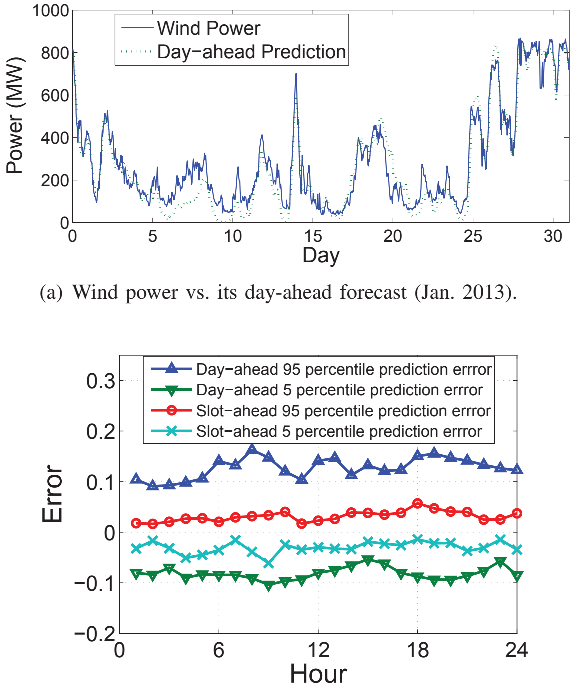
is from wind power. The total installed capacity of the wind power is $R_{\text{capacity}} = 930.65\text{MW}$.

We plot the realized renewable energy and the day-ahead predicted value of one month in Fig. 8 (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. 8 (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).



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

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

Fig. 8. Wind power and its forecasts.

F. Simulation Parameter Setting

We have synthesized an EV demand pattern from [21], and have obtained the day-ahead predicted values and the real values of both the background demand and the renewable energy from [20]. 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 [21] and [20].

We first focus on the EV demand. We use the synthesized EV demand (see Fig. 1), 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:

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

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:

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

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

G. An Example for the RHC Algorithm

We construct a simple example to demonstrate that the performance of the RHC (Receding Horizontal Control) algorithm can be very poor. To see this, consider the example in Fig. 9. Suppose that we have 48 time-slots. All EV charging jobs arrives at the beginning of time-slot 1, departs at the end of time-slot 48, and the total demand is 50. (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 the time-slots. However, their real values turn out to be 11 in real time (note that there is no intra-day prediction in this example). The offline optimal peak for this example is approximately 12. If we apply the RHC algorithm to this example, the resulting peak will be 15.5. On the other hand, like in 2-IPM, we know in advance that the error of day-ahead prediction is no more than 2. In other words, based on a day-ahead predicted value of 10 for the net non-deferrable demand, their real values will be between 8 and 12. Note that the real value of 11 specified earlier still fall into this range. We can then apply the EPS algorithm to the same setting. The EPS algorithm achieves an optimal competitive ratio 1.06 (computed from (9)). In contrast, the empirical competitive ratio of the RHC algorithm for this input is $15.5/12 = 1.29$, which is already much larger than 1.06. (The worst-case competitive ratio of the RHC algorithm can be even larger.) Hence, this example shows that the worst-case performance of the RHC algorithm can be much poorer than the EPS algorithm.

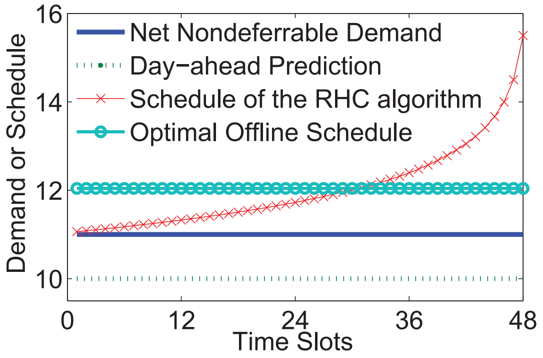


Fig. 9. Poor performance of the RHC algorithm.