On Modeling and Marketing the Demand Flexibility of Deferrable Loads at the Wholesale Level

Mahnoosh Alizadeh, Tsung-Hui Chang, and Anna Scaglione

Abstract—In this paper, we present a methodology to incorporate the service offered by an aggregator of longer duration deferrable loads in the Unit Commitment (UC) optimization. The description of the deferrable demand we propose is accurate, yet not computationally intensive, and can be added to any type of UC optimization so that the ISO can use time-shiftable loads as an additional dispatchable resource to minimize costs, shave demand peaks, and control the wholesale prices of electricity. Simulation results illustrate the benefits of this model compared to the so called “tank” and “detailed” models.

Index Terms—deferrable loads, quality of service, electric vehicles, renewable integration, unit commitment, demand response

I. INTRODUCTION

Demand response (DR) is a valuable asset that can be used instead of fast-ramping generation to reserve and regulation services in wholesale balancing markets. Within the class of controllable loads, the demand flexibility offered by thermostatically controlled loads, or similar types of appliances, is on shorter (sub-hourly) time scales and, thus, it is most suited to be dispatched in the ancillary services market. Therefore, there is no need to model the load shaping service they offer when solving the day-ahead Unit Commitment (UC) problem, which does not account for sub-hourly dynamics. However, if we were to move around electricity demand due to large groups of long-duration (>hourly) loads such as electric vehicles (EV), washing machines, dryers, etc., the ISO will need one of the following when solving the UC problem:

1) A regular day-ahead bid submitted by an aggregator in charge of controlling the time-shiftable loads. In this case, the aggregator acts as a price taker. It predicts the market clearing prices for the next day and internally solves for an optimized day-ahead bid to serve the population of deferrable loads it controls with minimum cost;
2) Statistics on the availability of deferrable loads, the amount of flexibility they offer, and the cost of shifting them for each hour of the next day. In this case, the ISO can directly solve for the optimized load curve to serve the population of deferrable loads jointly with the optimized generation schedules. The appliances can still be controlled by an aggregator, or the utility;
3) Complex price-quantity bids from each aggregator [1].

In the absence of a considerable amount of intermittent generation resources, the first option seems reasonable since the complexities are shifted to utilities or third party aggregators and the ISO can treat the load due to time-shiftable loads similar to other types of non-deferrable, unresponsive demand participating in today’s electricity market. However, with the presence of a considerable amount of renewables at the wholesale level, which are intermittent by nature, this approach is no longer viable since wholesale electricity prices will become much more non-transparent and volatile on a day-to-day basis, based on the availability of wind and solar energy.

The second option eliminates the need for the aggregator to predict wholesale prices in the presence of renewables since the schedules for deferrable loads and generators are solved for, concurrently, in the same optimizations (day-ahead, hour-ahead and balancing markets). However, to solve for the optimal dispatch, the ISO needs a concrete, yet simple enough, model that can capture the flexible demand offered by a large population of deferrable loads such as EVs. This flexible demand consists of the sum of load contributions from each individual appliance whose load can be deferred. Thus, one approach would be to model the contribution of each appliance by looking at customer usage behavior, e.g., when they arrive home and plug-in their vehicle or when they finish dinner and turn on the dishwasher. Other parameters include customer specified deadlines and appliance mode (e.g., EV charge level) [2]. Clearly, this approach does not scale and makes the optimization solved by the ISO overly complicated. At the wholesale level, we cannot consider individual customer behavioral parameters, including travel patterns and job deadlines. This is the reason why, missing a scalable model, most of the present work that look at integrating intermittent resources on the wholesale level with the help of deferrable loads mathematically represent the deferrable demand as a large tank that has to be filled by the end of the day [3], [4]. In this coarse approximation, all the valuable temporal information about the appliance arrival patterns, the time evolution of their energy consumption, and their time flexibility, is lost. To fix this problem, in this paper we provide a reasonably accurate model of the aggregate demand that can preserve most of the valuable temporal statistics pertaining to individual customer behavior, while reducing the associated computational complexities and communication requirements.

More specifically, we present a model and specify the associated parameters for what is arguably one of the most important types of time-shiftable loads in near future, i.e., EVs, based on real-world data on charging events and travel patterns studied in [5]. To find a scalable yet concrete model that suits the application we described, we present two different options, both of them not guaranteeing the timing of energy
delivery to each individual customer. This allows us to bundle customer into several service classes, with energy delivery to each class dispatched by the ISO under a First-In-First-Out (FIFO) discipline. Before delving into the details of the model and its benefits, one important aspect that we wish to emphasize is that this model can be used to solve the UC problem even when the actual scheduling technique used in real time is different and, possibly, more accurate in serving individual appliances. For example, the real time scheduler could incorporate deadlines (see e.g. [6]–[10]).

SYNOPSIS: In Section II, we present a methodology for providing a detailed, yet not computationally intensive, description of the electricity demand from deferrable loads. Since our description does not incorporate customer specified deadlines, we present an alternate technique to control the quality of usage delivered to the customers in Section III. In Section IV, we illustrate how the described model can be used to represent the service offered by deferrable loads in the UC optimization. As we will see, having accurate statistics of the arrival rate and the temporal evolution of the energy consumption of deferrable loads is an integral part of our model. Thus, as an example, we provide such statistics for plug-in hybrid electric vehicles (PHEV) in Section V.

II. APPLIANCE LOAD MODELING

Electricity demand from most appliances that handle long duration jobs and are deferrable (e.g., EVs, washing machines, dryers, dishwashers, etc.) can be characterized using three parameters: the time at which the energy request arrives ($t^a_i$), the laxity (amount of slack time before the job has to begin to finish by deadline) of each request offers ($\gamma_i$), and a parameter vector ($c$) that can be mapped one to one with the temporal evolution of the appliance’s electric load, once on. Specifically, assume that, if turned on time zero, the load due to each appliance can be described by the complex phasor signal $g(t; c)$ with parameter $c$. Thus, we use the tuple $(t^a_i, c_i, \gamma_i)$ to fully describe individual energy requests, indexed by $i$. As we will see later, for the special case of EVs, the vector $c_i$ consists of two elements: the charging power (level) and the duration of charging for each arriving vehicle. The pulse $g(t; c_i)$ is well approximated by a real rectangular pulse with a variable length and amplitude determined by the argument $c_i$.

We mentioned that appliances arrive to consume electricity according to some point process, with $t^a_i$ denoting the arrival time of the $i$-th job. In the scenario where each job is served immediately upon its arrival, the aggregate load is given by:

$$L_{UG}(t) = \sum_i g(t - t^a_i; c_i).$$  \hfill (II.1)

Similarly, if an aggregator is allowed to modify, either directly or by using pricing techniques, the time at which appliances start their job, the electricity consumption of the $i$-th request will start at $t^d_i > t^a_i$, thus modifying the deferrable load to be

$$L_G(t) = \sum_i g(t - t^d_i; c_i).$$  \hfill (II.2)

When there are deadline constraints, $t^a_i \leq t^d_i \leq t^a_i + \gamma_i$, $\forall i$.

In theory, the aggregator can gather statistics on the random tuples $(t^a_i, c_i, \gamma_i)$ and present those to to ISO, to be used in the UC optimization. However, the combinatorial optimization resulting from this detailed model becomes rapidly intractable. To find a compromise between complexity and performance, we simplify (II.2) using the following ingredients:

- As previously mentioned, when modeling the aggregate demand of deferrable loads, explicitly tracking customer deadlines will incur prohibitive communication and computational costs. Thus, we do not consider the laxity values $\gamma_i$ in the model outlined next. \footnote{Remember that the actual scheduling technique used by the aggregator can still be deadline-constrained. Implementing the scheduling algorithm in a distributed fashion would avoid most of the mentioned problems. However, here we need a centralized description that can be used in the dispatch optimization.}
- Instead, one option is to resort to a best effort service model used in many popular media applications such as voice, streaming services, or most famously, the protocol responsible for routing packets on the Internet;
- The vectors $c_i$ cannot be communicated and stored with full precision. The limited capacity of communication links requires us to quantize the numerical values being exchanged and stored;
- Communications as well as decisions are carried out at discrete epochs rather than continuously.

Consequently, as argued in [11], we define the quantization function $\Psi(c)$, which maps the continuous valued vectors $c$ onto the discrete codebook $C = \{ C_1, \ldots, C_q, C_{q+1}, \ldots, C_Q \}$. We associate a service queue with each of the elements in $C$. Each appliance arriving to receive service (electricity) from the aggregator belongs to one of these queues. We define the continuous arrival process in each queue (a stochastic process which counts the total number of arrivals) as

$$a_q(t) = \sum_i \delta(c_i - C_q) u(t - t^a_i), \quad q = 1, \ldots, Q,$$  \hfill (II.3)

where $\delta(.)$ is the Kronecker delta function, which is equal to 1 only if $c_i = C_q$ and zero otherwise. Note that the arrival process counts the number of arrivals in the queue since the origin of time ($t = 0$). Each arrival process is characterized by a non-homogeneous rate $\lambda_q(t)$, which describes the intensity of the arrival events at time $t$ for the $q$-th queue.

If the access of appliances to the grid is somehow controlled and we have $t^d_i > t^a_i$, appliances wait in their associated queue until they are authorized to start consuming electricity, which we denote as $\text{departing}$ the queue. Thus, we define as the continuous departure process from each queue the stochastic process which counts the total number of departures,

$$d_q(t) = \sum_i \delta(c_i - C_q) u(t - t^d_i), \quad q = 1, \ldots, Q.$$  \hfill (II.4)

As mentioned, we assume that updates are carried out only in discrete epochs. Let the same variable $t$ denote the discrete times at which communication as well as computation updates happen. Accordingly, $a_q(t)$, $d_q(t)$, $g(t; C_q)$ and $\lambda_q(t)$ now denote the discrete counterparts of the continuous functions.
defined previously. Hence, we can rewrite (II.1) as,
\[
L_{S}(t) = \sum_{q=1}^{Q} \sum_{t=0}^{t} [a_{q}(t) - a_{q}(t-1)]g(t - \ell; C_{q}). \tag{II.5}
\]
Similarly, if some type of load control policy is in place:
\[
L_{S}(t) = \sum_{q=1}^{Q} \sum_{t=0}^{t} [d_{q}(t) - d_{q}(t-1)]g(t - \ell; C_{q}). \tag{II.6}
\]
Note that, contrary to (II.2), (II.6) is linear mapping and it can be used to represent a simplified but tractable model for the demand of deferrable loads in the UC problem. Since the model does not discriminate tasks with different deadlines, the only decision variables in (II.6) are the value of the departure processes \(d_{q}(t)\)'s, with the simple linear constraints that the values of \(d_{q}(t)\) are non-negative and the causality constraints requiring \(d_{q}(t-1) \leq d_{q}(t) \leq a_{q}(t)\). Consequently, a decision-making unit using the load description in (II.6) to dispatch the deferrable loads only needs statistics on the arrival processes \(a_{q}(t)\), which describe the arrival trend of appliances in each queue. These statistics can be provided by each aggregator to the ISO. If the arrivals follow a non-stationary Poisson process (which will later show is reasonable for the case of EVs), a rate function \(\lambda_{q}(t) = E[a_{q}(t)]\) for each queue gives us full statistical information about the arrivals. The information required to choose appropriate estimates of the \(\lambda_{q}(t)\)'s is:
- the set of parameters to dynamically updated forecasts of an aggregate arrival rate \(\lambda(t)\), describing the intensity of arrivals irrespective of which queue they belong to. In other words, \(\lambda(t) = \sum_{q=1}^{Q} \lambda_{q}(t)\);
- a reliable estimate of the probability density function for the vector \(c\), which we denote as \(f_{c}(c)\)

Given these, we can write the arrival rate in each queue as:
\[
\lambda_{q}(t) = \lambda(t) \int_{c \in \Psi^{-1}(C_{q})} f_{c}(c)dc, \quad q = 1, \ldots, Q. \tag{II.7}
\]
In Section V, we will provide statistical models to provide the above-mentioned values for the case of plug-in electric vehicles. For a more detailed description of the communication and computational infrastructure required to collect the information required for the model described above, along with a scheduling optimization to determine the \(d_{q}(t)\)'s in real-time to minimize the cost of the aggregator, see [12].

III. ACCOUNTING FOR THE SERVICE QUALITY

We are now one step closer to our goal: using the description in (II.6) to include deferrable loads as a dispatchable resource in the UC optimization. But what is the cost of using this resource? Normally, one would think that as long as the appliance finishes it’s job by the deadline specified by the customer, e.g., the EV is charged by 7 AM, the deferrable loads can be shifted with no cost. However, as mentioned, enforcing individual deadline constraints would make the optimization computationally infeasible at the wholesale aggregation scale. To alleviate this problem, we propose one of the following:

1) The ISO can minimize the average delay experienced by the entire population rather than taking hard deadlines into account. Obtaining the aggregate delay experienced by the population receiving service from a queue in a time interval is rather simple if we know (or can forecast) \(a_{q}(t)\) and \(d_{q}(t)\). A queue polygon is obtained by super-imposing the departure and arrival processes \(a_{q}(t)\) and \(d_{q}(t)\). The area inside the polygon gives the total delay incurred by the appliances in the \(q\)-th queue for a specific time interval [13]. If we associate different costs to a unit time delay in different queues in order to provide different Quality of Service (QoS) rates, we can define the delay cost for the entire population at time \(t\) as,
\[
C_{D}(t) = \sum_{q=1}^{Q} C_{q}(t)(a_{q}(t) - d_{q}(t)), \tag{III.1}
\]
where \(C_{q}(t)\) is the cost per unit time delay for the \(q\)-th queue. To further refine the cost of modifying the load, we can take the laxity offered by individual energy requests into account when determining which queue they belong to, i.e., the quantization mapping would change to \(\Psi(c, \gamma)\). Accordingly, we can associate higher delay costs \(C_{q}(t)\), and possibly, higher billing rates, to deferrable energy requests with lower laxity values. Discriminating between customers that offer different levels of delay intolerance will allow for a more accurate description of the flexibility that the aggregate deferrable load can provide in the UC optimization.

Note that the per unit delay costs \(C_{q}(t)\) in (III.1) should be intelligently designed to avoid unacceptable delays. In our numerical results, we predicted the average savings that can result from shifting EV battery charges from peak to off-peak hours and designed the \(C_{q}(t)\)'s proportionally. More specifically, we calculated the average savings for each queue as (average savings per hour of charge shifted) \(\times\) (length of charge). To ensure an average delay of 3 hours, we divided these savings by 3 and assigned the resulting numbers to the \(C_{q}(t)\)'s. This way, it would be unlikely for any vehicle to be delayed for more than 3 hours since the resulting savings are not comparable to the delay cost.

Inclusion of a cost associated with the delay will ensure that the UC optimization will not over-estimate the flexibility of the deferrable loads or the ability of the aggregator to pay back the customers for their service. These costs may not be directly paid to the aggregator. Rather, they provide a description of how much inconvenience is caused to the customer through the demand shifting program or how much the aggregator has to pay its customers for the service they are providing: by shifting the customers demand to less congested periods, the aggregator makes a profit from receiving an optimized day-ahead bid from the ISO, which will reduce the price it has to pay for the electricity bought in the wholesale market. Thus, no double payment is necessary for this service.

2) The ISO can enforce per queue maximum delay constraints instead of individual appliance deadlines as follows. This is, again, an approximation that we envision would give an accurate enough description of the flexibility of the demand at the wholesale level. Denote the maximum delay that appliances in the \(q\)-th queue can tolerate by \(\gamma_{q}\). Then, maximum delay constraints for the \(q\)-th queue are:
\[
d_{q}(t) > a_{q}(t) - \gamma_{q}, \quad \forall q, t. \tag{III.2}
\]
Note that this would assign different laxity (maximum delay) values to different queues but not to individual appliances in the queues. As for the other quantities that are quantized, the number of constraints increases linearly with the number of queues but does not vary with population size and the accuracy can be refined by increasing the number of queues. In fact, appliances previously belonging to the same queue are in this case divided in multiple queues based on their degree of flexibility. For example, a PHEV with a charge request of 3 hours and 1 hour of flexibility would belong to a different queue from a PHEV with a charge request of 3 hours and 2 hours of flexibility.

IV. MARKET INTEGRATION OF DEFERRABLE LOADS

The UC problem is the optimization required for deciding which generation plants should be running at each hour of the next day in order to serve the time-variant demand of electricity in a wholesale electricity market. Here we look at a rather simple formulation of the UC problem in order to illustrate how one can include the service available from the deferrable loads in the set of resources that can be dispatched.

A. The Standard UC Optimization

The objective of the UC optimization is to minimize the cost paid by the retailers to the generation units over all hours of the next day. Here, our goal is not to present or improve the solution methodologies for the UC optimization in its most general form but rather, to illustrate how to easily integrate a more accurate description of the deferrable loads in the optimization, overcoming the inaccuracies of the crude tank model, while keeping computational complexities at bay. Namely, if we have \( N \) generation units, \( D \) demand buses and \( W \) transmission lines, we can write the mathematical version of a simplified UC problem for a time span of \( H \) hours as,

\[
\min_{\{G^i(t), X^i(t)\}} \sum_{i=1}^{H} \sum_{t=1}^{N} J_i(X^i(t)G^i(t)), \quad (IV.1)
\]

where \( X^i(t) \) and \( G^i(t) \) respectively denote the state (on/off) and the amount of energy generated by the \( i \)-th generation unit at time \( t \) and, the function \( J_i(\cdot) \) denotes the cost curve of the \( i \)-th unit. The above sum gives the total generation cost at time \( t \). The optimization has to be solved for all \( H \) hours of the next day. The problem is subject to the following constraints:

1) Demand/supply balance constraint:

\[
\sum_{i=1}^{N} X^i(t)G^i(t) = \sum_{j=1}^{D} L^j(t), \quad (IV.2)
\]

where \( L^j(t) \) is the demand at the \( j \)-th load bus at time \( t \).

2) Generation capacity constraints:

\[
G^i_{\min} \leq G^i(t) \leq G^i_{\max} \quad i = 1, \ldots, N, \quad (IV.3)
\]

where \( G^i_{\min} \) and \( G^i_{\max} \) denote the minimum and maximum generation capacity of the \( i \)-th generation unit.

3) Minimum up and down time constraints:

\[
\forall i = 1, \ldots, N : \begin{align*}
& \text{if } X^i(t-1) = 0 \text{ and } X^i(t) = 1 \quad (IV.4) \\
& \quad \rightarrow \forall k < ST_i X^i(t+k) = 1, \\
& \text{if } X^i(t-1) = 1 \text{ and } X^i(t) = 0 \quad (IV.5) \\
& \quad \rightarrow \forall k < SD_i X^i(t+k) = 0,
\end{align*}
\]

where \( ST_i \) and \( SD_i \) respectively denote the minimum up and down times of the \( i \)-th generation unit.

4) Maximum line flow constraint:

\[
\left| \mathbf{H}[G^1(t) \ldots G^N(t) L^1(t) \ldots L^D(t)]^T \right| \leq F, \quad (IV.6)
\]

where the \( W \times (N+D) \) matrix \( \mathbf{H} \) relates nodal generation and load injections to line flows on the \( W \) transmission lines. The \( W \times 1 \) vector \( F \) captures the maximum flow on each line and \( \leq \) denotes element by element inequality.

This is the DC model but the constraints can be replaced by the AC model. Additional elements that can be added to the above optimization include start-up and shut-down costs, maximum generation ramp rates, reserve generation capacity, and \((N-1)\) contingency constraints. These aspects have inconspicuous consequences on the model we are describing. More interesting are variations of the UC problem that include intermittent resources the day ahead [2], where the rich statistical testing are variations of the UC problem that include intermittent resources the day ahead [2], where the rich statistical model we describe next for deferrable loads could be used more proficiently to assess risks. However, this presents such complexities that it is beyond the scope of this work.

B. UC Optimization with Deferrable Loads

With the addition of aggregators that offer deferrable loads as a service, the load at a subset of nodes \( j \in A \) changes from a deterministic value to:

\[
L^j(t) = L_{\text{abc}}^j(t) + L_{\text{def}}^j(t), \quad (IV.7)
\]

where \( L_{\text{abc}}^j(t) \) denotes the non-controllable load for the aggregator at the \( j \)-th load bus. From this point on, we use the superscript \( j \) to distinguish to all the previously defined quantities for the \( j \)-th aggregator. Consequently, there will be new decision variables added to the UC problem: the expected rate of departures \( d_j(t) \) of appliances from each queue on the next day, which we will denote as \( \nu_j(t) \). We propose that, if using an aggregate delay cost, (IV.1) can be modified as,

\[
\min_{\{G^i(t), X^i(t)\}} \sum_{i=1}^{H} \sum_{t=1}^{N} J_i(X^i(t)G^i(t)) + \sum_{j \in A} C_D^j(t), \quad (IV.8)
\]

where \( C_D^j(t) \) is the delay cost for the \( j \)-th aggregator, given by (III.1). Note that \( C_D^j(t) \) is a linear term. The optimization subject to the following constraints:

1) constraints (IV.2) to (IV.6) with:

\[
L^j(t) = L_{\text{abc}}^j(t) + \sum_{q=1}^{Q} \sum_{\ell=1}^{t} [\nu_q^j(\ell) - \nu_q^j(\ell-1)]g^j(t - \ell; C_q)
\]

for \( j \in A \).
2) The causality constraints for each hour:

\[
\forall j \in \mathcal{A}, q = 1, \ldots, Q \\
\text{if } t < H : \quad \mu_q^j(t-1) \leq \mu_q^j(t) \leq \sum_{\ell=1}^H \lambda_q^\ell(t), \\
\text{if } t = H : \quad \mu_q^j(H) = \sum_{\ell=1}^H \lambda_q^\ell(t), \quad \text{(IV.9)}
\]

with the constraint on the last hour of the day ensuring the deferrable loads are all served by the end of the day. Note that this would allow loads to be served on the next day since they are only departing their associated queue at \( t = H \). A more strict constraint would be to ensure that all loads are served by \( t = H \). If we denote by \( \tau_q \) the length of the jobs associated to the \( q \)-th queue, this would mean that \( \forall q \):

\[
\mu_q^j(H - k) = \sum_{\ell=1}^{H-k} \lambda_q^\ell(t), \quad \forall k < \tau_q. \quad \text{(IV.10)}
\]

As it is easily seen, some constraints like (IV.4), (IV.5) and (IV.9) include the decision variables for several hours. Consequently, the minimization of cost has to be carried out over the entire day. Several solution approaches exist, which include Dynamic Programming, Lagrangian Relaxation, Mixed Integer Programming, etc.

As mentioned before, another option is to eliminate the delay cost and enforce the constraints in (III.2) to ensure acceptable delays.

C. Computation Reduction by Decomposition

One of the important issues of (IV.8) is how to efficiently solve it when there are a large number of generators, aggregators and deferrable loads. The main difficulty is that the model of deferrable loads requires solving an expanded 24 hours UC optimization jointly rather that on an hour by hour basis. Moreover, the complexity of solving (IV.8) exponentially increases with \( N \) and \( |A| \). One possible approach to facilitate the scalability issue is to consider the dual decomposition method [14], for which the complexity linearly increases with \( N \) and \( |A| \) instead. In particular, by looking at the Lagrangian dual of (IV.8), we are able to decompose (IV.8) into several subproblems that have considerably smaller problem sizes.

For ease of illustration, let us define

\[
H = [h_1, \ldots, h_N, \tilde{h}_1, \ldots, \tilde{h}_D],
\]

and write the constraint in (IV.6) as

\[
\sum_{i=1}^N h_i X_i(t) G_i(t) + \sum_{j \in A} \tilde{h}_j L_j(t) \leq F, 
\]

\[
- \sum_{i=1}^N h_i X_i(t) G_i(t) - \sum_{j \in A} \tilde{h}_j L_j(t) \leq F. \quad \text{(IV.11)}
\]

Denote by \( \xi(t) \in \mathbb{R}, \lambda(t) \geq 0 \) and \( \eta(t) \geq 0 \) the dual variables associated with the constraint in (IV.2) and the two constraints in (IV.11), respectively. Then the dual of (IV.8) is

\[
\max_{\xi(t) \in \mathbb{R}, \lambda(t) \geq 0, \eta(t) \geq 0} \Phi(\{\xi(t), \lambda(t), \eta(t)\}_k) \quad \text{(IV.12)}
\]

where \( \Phi(\{\xi(t), \lambda(t), \eta(t)\}_k) \) is the dual function which can be decomposed as follows

\[
\Phi(\{\xi(t), \lambda(t), \eta(t)\}_k) = \sum_{i=1}^N \Psi_i(\xi(t), \lambda(t), \eta(t))
\]

+ \( \sum_{j \in A} \sum_{q=1}^Q T_{j,q}(\{\xi(t), \lambda(t), \eta(t)\}_k) + \Omega(\{\xi(t), \lambda(t), \eta(t)\}_k). \quad \text{(IV.13)}
\]

In the above equation, \( \Psi_i(\xi(t), \lambda(t), \eta(t)) \) involves only the optimization of generator \( i \):

\[
\min_{i \in 1, \ldots, H} \sum_{t=1}^H \left( J_i(X_i^t(G_i^t)) + (\xi(t) + (\lambda(t) - \eta(t))^T h_i) X_i^t G_i^t(t) \right)
\]

s.t. constraints in (IV.3), (IV.4) and (IV.5). \quad \text{(IV.14)}

The term \( T_{j,q}(\{\xi(t), \lambda(t), \eta(t)\}_k) \) involves only the scheduling of the \( q \)-th queue in aggregator \( j \in A \):

\[
\min_{t \in [\mu_q^j(t)]_{H+1}} \sum_{t=1}^H \left( -C_q^j(t)d_q^j(t) + (\xi(t) + \lambda(t) - \eta(t))^T h_i \right) X_i^t G_i^t(t)
\]

s.t. constraints in (IV.9) and (IV.10), \( t = 1, \ldots, H \). \quad \text{(IV.15)}

Finally, \( \Omega(\{\xi(t), \lambda(t), \eta(t)\}_k) \) is a constant independent of \( G_i^t(t) \) and \( \mu_q^j(t) \).

Therefore, given the dual variables \( \{\xi(t), \lambda(t), \eta(t)\}_k \), one can evaluate the dual function in (IV.13) by solving the problems in (IV.14) for all \( i \), and the problems in (IV.15) for all \( j \in A \) and \( q \), in a fully parallel manner. The dual variables \( \{\xi(t), \lambda(t), \eta(t)\}_k \) can be iteratively updated by the subgradient method [15]. Specifically, at iteration \( n \), one can update \( \xi(t), \lambda(t), \eta(t) \) as follows:

\[
\xi^{(n+1)}(t) = \xi^{(n)}(t) + c_n \left( \sum_{i=1}^N X_i^t G_i^t(t) - \sum_{j=1}^D L_j^t(t) \right),
\]

\[
\lambda^{(n+1)}(t) = \left[ \lambda^{(n)}(t) + c_n \left( \sum_{i=1}^N X_i^t G_i^t(t) - \sum_{j=1}^D L_j^t(t) - F \right) \right]^+, 
\]

\[
\eta^{(n+1)}(t) = \left[ \eta^{(n)}(t) + c_n \left( - \sum_{i=1}^N X_i^t G_i^t(t) - \sum_{j=1}^D L_j^t(t) - F \right) \right]^+, 
\]

for \( k = 1, \ldots, H \). The decomposition algorithm then iteratively updates the dual variables above and the inner minimization problems in (IV.14) and (IV.15) until predefined stopping criterion is satisfied [15].
V. ESTIMATING $\lambda_q(t)$’S FOR THE CASE OF PHEVS

It is apparent from previous discussions that having accurate statistics on the arrival rate of the appliances in the service queues, i.e., $\lambda_q(t)$, is an integral part of our model. As an example, here we provide such statistics for the specific case of plug-in hybrid electric vehicles (PHEV). The reason behind focusing on PHEVs is that we have access to real-world PHEV charging data that allows us to avoid numerous speculative assumptions that were unavoidable otherwise. For a complete description of this dataset, along with a more detailed statistical model and analysis, please see [16].

Many authors have looked at the statistics of electric vehicle charging load before (see e.g. [17]–[20]). However, none of the mentioned papers provides an analytical description of the individual charging request statistics. Either a very simple load model is assumed (such as constant arrivals) or the load is directly extracted from data using Monte-Carlo simulation techniques. Instead, the detailed analytical model presented next can be used to dynamically update the forecasts of $\lambda_q(t)$.

A. Arrival Process

We propose a flexible model to capture the random nature of charging request arrivals. Our model emulates the procedures used to determine optimized staffing for call centers with stochastic inhomogeneous demand. The most popular model for the call arrivals is an inhomogeneous (or piece-wise constant) Poisson process with a random arrival rate [21]. We argue this model is also appropriate for PHEV charging events in [16] by constructing a test of null hypothesis that shows that an inhomogeneous Poisson arrival model is a reasonable model for PHEVs.

Note that for a Poisson process, all we need towards a full statistical description of the random arrival process is the mean arrival rate $\lambda(t)$. Thus, parameter fitting is particularly simple and likely to apply for residential charging, as there is no reason to believe that the inter-arrival times would be correlated.

B. Modeling and predicting the time series $\lambda(t)$

To specify the statistics of the EV arrival process for the next day, we need to indicate the inhomogeneous rate function. Hence, we need a mathematical model for the underlying dynamics of $\lambda(t)$. We know that this arrival rate is not deterministic and also, it’s future values are correlated with previously collected historical data. Next, we develop a stochastic model for the time series $\lambda(t)$ which we will use to predict the next day’s arrival rates from arrival data.

Assuming that $\lambda(t)$ is piecewise constant in $t$, we can divide each day into $K$ segments to fully describe the rate for the entire day, with each segment having a constant value. For ease of notation, we divide the time series of the arrival rates into a sequence of daily arrival vectors, which we denote by $\Lambda_s = [\lambda((s-1) \ast K + \ell)]_{\ell=1,\ldots,K}$ for the $s$-th day. However, note that the historical data includes arrival counts and not arrival rates, since the rates are unobservable. Thus, we define $m_s = \lfloor m((s-1) \ast K + \ell) \rfloor \sim \text{Pois}\left(\sum_{\ell=1}^{K} \lambda((s-1) \ast K + \ell)\right)$ as the associated count vector for the $s$-th day.

We need to develop a time series model that can model and predict future arrival rates from the set of count vectors $[m_s]_{s=1,\ldots,n}$, with $n$ being the total number of days of data available the records.

Since the count values $m(t)$ are realizations of Poisson random variables with in-homogeneous variable means $\lambda(t)$, their associated variances, which are identical to the mean values, are time-varying as well. In order to solve numerical issues that arise from this fact, we use a simple variance-stabilizing transformation suggested in [22] and use a slightly modified version of the Anscombe square-root transform [23], which transforms Poisson distributed data like $m(t)$ to approximately normally distributed data with a constant variance of $1/4$.

The transformation is as follows:

$$T : m(t) \sim \text{Pois}(\lambda(t)) \rightarrow \sqrt{\frac{m(t) + \frac{1}{4}}{\frac{3}{4}}} \sim N\left(\sqrt{\lambda(t)}, \frac{1}{4}\right) \quad (V.1)$$

After applying (V.1) to the data, we present the transformed vector associated with day $s$ as $m^{tr}_s$. To train the model, we store our data in a matrix $M^{tr} = [m^{tr}_s]_{s=1,\ldots,n}$. Arrival counts will have multiple seasonalities that we need to model when we predict this time-series. Normally, one would use multivariate time series models. However, the large dimensions of the dataset will pose as a barrier. To address this issue, we will try to reduce the dimensions required to represent each daily arrival vector (currently, this is equal to $K$) as follows. We use Singular Value Decomposition (SVD), carried out on the data matrix $M^{tr}$. Recall the SVD of a $K \times n$ matrix $M^{tr}$. The SVD returns two unitary matrices $U_{K\times K} = [u_1, \ldots, u_K]$ and $V_{n\times K} = [v_1, \ldots, v_K]$ such that:

$$M^{tr} = U \Sigma V^T, \quad (V.2)$$

where $\Sigma$ is a diagonal matrix with the singular values $[\sigma_1, \ldots, \sigma_{\min(K,n)}]$ on its diagonal. Given $U$, $V$ and $\Sigma$, each of the count vectors $m^{tr}_s$ can be approximated as a linear combination of $\kappa \leq K$ most significant left singular vectors:

$$m^{tr}_s \approx \sum_{i=1}^{\kappa} \alpha_i(s)u_i = \sum_{i=1}^{\kappa}(\sigma_i v_i(s))u_i, \quad s = 1, \ldots, n. \quad (V.3)$$

where $v_i(s)$ is the $s$-th element in the $i$-th right singular vector $v_i$. If $\kappa = \text{rank}(M^{tr})$, which is equal to the number of non-zero singular values, (V.3) is exact. However, with a high-dimensional dataset expected in our case, it is best to choose the most significant left singular vectors. More specifically, Fig. 1 displays the first six principal component coefficients for the NHTS database, which capture 96% of the variance.

The next step is to forecast the arrival counts on the next day $m_{n+1}$ from the available count profiles $m_s$, $s = 1, \ldots, n$. Since $V$ is an orthogonal matrix, $(\alpha_1(1), \ldots, \alpha_\kappa(n))$ is orthogonal to $(\alpha_k(1), \ldots, \alpha_k(n))$ for $k \neq i$. Thus, it is reasonable to assume that a prediction of the series $\alpha_i(s)$ can be done using separate univariate forecasts for different values of $i$ [24]. With this, the model is reduced to forecasting $\kappa$ separate univariate time series using classical time-series analysis.
Since each of the time series $\alpha_i(s)$ follows a weekly periodic pattern, we choose to apply a periodic AR (PAR) model to accurately predict future $\alpha_i(s)$’s. PAR models allow the coefficients in an autoregressive (AR) model to change with various seasons, as time varying filters. This means that we would have different AR coefficients for each day of the week. A PAR($p$) model with period $P$ (which for a weekly period would be equal to 7) for $\alpha_i(s)$ is described by:

$$\sum_{\ell' = 0}^{P} \phi_{i,\ell} w(kP + \ell - \ell') = w(kP + \ell),$$ (V.4)

with $w(.)$ as white noise. The coefficients $\phi_{i,\ell}$ can be calculated using the periodic Yule Walker equation for each principal component coefficient. They should be dynamically updated as new data becomes available.

Note that the eigenstructure of the data itself may also change as we include more count vectors in our database. Thus, the vectors $u_1, \ldots, u_s$ may need to be updated as well. Recursive subspace tracking methods can be used to track these changes in an online fashion [25].

After finding the parameters of the model, we summarize the steps required to forecast arrival rates on the $s$-th day from the arrival counts on the previous days:

1. First, we calculate $\alpha_i(s - \ell), \ldots, \alpha_i(s - 1)$ for all $i = 1, \ldots, K$.
2. Next, we use estimated $\tilde{\phi}_{i,\ell}$ and (V.4) to calculate $E[\alpha_i(s)]$ for all $i = 1, \ldots, \kappa$. If we have $\ell = s - [s / P]$,

$$E[\alpha_i(s)] = -\frac{1}{\tilde{\phi}_{i,\ell}} \sum_{\ell' = 1}^{P} \phi_{i,\ell} \alpha_i(s - \ell');$$ (V.5)

3. Using (V.3), we obtain

$$E[m_i^s] \approx \sum_{i = 1}^{\kappa} E[\alpha_i(s)] \alpha_i;$$ (V.6)

4. Finally, we have from (V.1) that $\Lambda_s = E^2[m_i^s]$ (element by element).

### C. Modeling charge requests for PHEVs

Now that we have seen how we can predict the arrival rates, we look at the second element in the model, which is the calculation of the queue arrival rates $\lambda_q(t)$ for PHEVs. Because of (II.7) this means that we need to model the PDF $f_c(c)$ of charge requests. First of all, we have to determine what the parameter vectors $c$ would include for PHEVs. If we approximate the power consumption profile with a real rectangular pulse, the width and the amplitude of the pulse would fully describe it. This is equivalent to having the charging length (which we denote by $S_i$ for the $i$-th charging event), as well as the charging power/rate (which we denote by $R_i$). Due to the limitations of available real-world data, we refrain from making any statements on the distribution of the charging rate $R_i$ and assume that all vehicles use level-1 home charging (1.1 kW). Thus, we only need to learn the PDF of the charge duration $S$ to complete our model.

The 620 charging samples we have access to give noisy estimates of the PDF of charge durations $f_S(s)$. Hence, we emulate [19], [20] and map the travel patterns recorded in the NHTS database into charging amounts, with the important difference that we use real-world charging data to train and check the accuracy of the derived PDF. Table I provides the values of kWh of energy we assumed the vehicles would require per traveled mile. For each vehicle type, we assumed the kWh per mile, which we denote by $\epsilon_i$, is uniformly distributed between the given values.

If we denote the miles traveled by $i$-th arriving vehicle as $M_i$, its battery capacity by $T_i$, the amount of time the vehicle will be parked at home and can charge as $\rho_i$, and assume that the owner will plug their car in with a probability $P_{\text{plug}}(M_i)$ after returning home having driven $M_i$ miles (binary random variable with success probability being a function of $M_i$), we can write the duration of charge $S_i$ for the vehicle as

$$S_i = \min\left\{\frac{T_i}{R_i} \epsilon_i, M_i, \rho_i\right\}.$$ (V.7)

and zero otherwise.

In training our model, we assume that the battery capacities $T_i$ will be uniformly distributed between 4.75-5.25 kWhs (consistent with the charging data we have access to). As mentioned, $R_i$ is assumed to be constant and $\epsilon_i$ follows Table I. $M_i$ and $\rho_i$ can be taken from the NHTS database. However, the one parameter we have not yet talked about is the plug-in probability, which as we will see plays an important role in the accuracy of the model. Accounting for $P_{\text{plug}}(M_i)$ is the major difference of our mapping from that of [19], [20].

In order to find an appropriate estimate of $P_{\text{plug}}(M_i)$, we use the 620 samples of real charging data. However, we

---

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>kWh per mile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sedans</td>
<td>0.18 - 0.3</td>
</tr>
<tr>
<td>Vans</td>
<td>0.3 - 0.4</td>
</tr>
<tr>
<td>SUVs</td>
<td>0.4 - 0.5</td>
</tr>
<tr>
<td>Trucks</td>
<td>0.5 - 0.7</td>
</tr>
</tbody>
</table>

---

3Note that we are considering $f_c(c)$ independent of time. This simplification is not essential, and using an $f_c(c; t)$ instead does not change fundamentally our model.
refrain from using a continuous mapping since that would inherently make the distribution of $S_i$ derived from the NHTS database identical to that of the real charging events. Instead, we choose to have a rough piecewise constant description. After mapping NHTS entries into charging events, we see that for the following probability distribution for vehicle plug-ins:

$$ P_{plug}(M_i) = \begin{cases} 0.05, & M_i < 1.5 \\ 0.1, & 1.5 < M_i < 8 \\ 0.5, & 8 < M_i < 16 \\ 1, & 16 < M_i \end{cases}, \quad (V.8) $$

a two-sample Kolmogorov-Smirnov test does not reject the hypothesis that the 600 real-world samples and the 150,000 charging amounts derived from the NHTS database come from the same probability distribution (at a 5% significance level). The distribution of the two datasets (real charging data and the samples derived from NHTS) is shown in Fig. 2.

Finally, after testing several distributions, we saw that a log-normal distribution $\ln \mathcal{N}(5.03, 0.78^2)$ best fits the samples:

$$ F_{S_i}^{PHEV}(s) = \frac{1}{0.78 \sqrt{2\pi s}} \exp\left(\frac{(\ln s - 5.03)^2}{2 \times 0.78^2}\right), \quad (V.9) $$

where the unit of $s$ is minutes. The mean of this random variable is 207 minutes. Another step to further improve this fit would be to clip the distribution at a maximum duration of 5.204 hours.

D. Laxity of Charge Requests

Finally, we look at the laxity (slack time) of the charging requests gathered by UC Davis [5] and derived from NHTS data. The PDFs have a common characteristic: two distinct peaks, one at 1-2 hours and the next at 8-10 hours. We can confirm from real-world data that daytime requests contribute mostly to the first peak, while nighttime requests mostly belong to the second peak. Consequently, we fit two different probability density functions to represent the laxity offered by daytime and nighttime charge requests gathered by [5]. We found that an exponential distribution with a mean of 1.089 best fits the daytime data, while the nighttime laxity is best represented by a lognormal distribution $\ln \mathcal{N}(2.25, 0.4^2)$.

VI. Numerical Results

In this section, we use the statistical models presented in Section V to demonstrate the benefits of the framework presented in Section IV for integrating deferrable loads in the UC problem. More specifically, we simulate the electricity demand of an aggregator responsible for managing the recharging of a population of 15000 PHEVs in a single substation. We assume that all vehicles use 1.1 kW level 1 home charging.

Algorithm 1 illustrates how we use the NHTS data jointly with our model to predict realistic arrival rates of charging requests that can be presented by the aggregator to the ISO. We choose to divide the charging requests in 5 separate service queues, each representing a charge length of 1 to 5 hours. The described aggregator in charge of the 15000 electric vehicles is connected to a grid modeled by the IEEE 9 bus test case. We have 3 generators and 3 loads placed on the nodes of a grid with a total of 9 buses and 9 lines. Generators are assumed to have quadratic generation costs, maximum ramping rates and minimum and maximum capacities specified by the test case data. We ignored minimum up and down time constraints, for simplicity. Since the expected demand modification cost curve (second term in the objective function of (IV.6)) and all the imposed constraints are also linear functions, the optimization can be formulated as a quadratic program. We used the function quadprog from the Matlab optimization toolbox to solve our UC optimization problem. Bus numbers 5, 7 and 9 are the load buses. To EV aggregator is connected to node 5, also serving a base load with a peak of 25 MWs.

Algorithm 1 Simulation of PHEVs electrical load: Day-ahead arrival rate prediction

1: Find the principal components of arrival events for NHTS. Normalize by population (Fig. 1)
2: Divide the arrival events randomly across 10 weeks
3: Find principal component coefficients for each day
4: Remove weekly/monthly mean of each of the principal component coefficient series
5: Estimate the parameters of a PAR(3) model in (V.4) for each principal component coefficient series based on the first 9 weeks of data
6: Forecast principal component coefficients for the last Monday (the tenth week) using the models in step 5
7: Forecast $\lambda(t)$ for the last Monday using (V.3)
8: Define 5 service queues, with charge lengths $S_q = 1, \ldots, 5$ hours
9: $\lambda_q(t) = \lambda(t) \int_{s \in \Psi^{-1}(S_q)} f_s^{PHEV}(s) ds, \quad q = 1, \ldots, Q$.

Algorithm 2 describes the scheduling scheme used by the aggregator to shape the aggregate EV charging demand to follow the day-ahead dispatch as closely as possible in real-time. The upward and downward balancing prices were assumed to be constant. See [12] for a full description of the real-time scheduling technique, modified to incorporate hard deadlines. Note that our scheduler is ignoring distribution feeder capacities. However, evidence shows that they will play a major role in determining the charging scheduling of EVs.
In Fig. 3, we look at the simplified case where the uncontrollable electricity demand of all the load buses is assumed to be constant, in order to showcase the load flattening effects of the addition of a reservoir of deferrable loads. In the tank model, the only constraints on the deferrable demand are causality and that all the demand has to be served by the next morning. We can see that using a tank model, the day-ahead dispatch of bus 5 is completely flattened. However, due to the hard deadlines imposed on the real-time battery recharge scheduling problem by Algorithm 2, the real-time load cannot follow the flat day-ahead dispatch. The day-ahead dispatch that resulted from our model approximately accounts for these constraints and is thus much more closely followed by the real-time load profile.

Next, we use realistic uncontrollable load curves and compare the performance of aggregator when its day-ahead dispatch is derived from each of the following: the tank model, our model, and the dispatch resulting from having a detailed model for each individual appliance with hard deadlines. Fig. 4 shows all the day-ahead dispatches and how close they are followed in real-time. The deficiencies of the tank model are better illustrated in this scenario. By shifting almost all of the charging load of PHEVs to the late night hours, the tank model makes it impossible for the demand to follow its day-ahead dispatch closely in real-time. The performance of the detailed model and our model is very close, indicating that the approximation made by our model are valid.

Lastly, we want to conclude by looking at how different levels of penetration of EVs can help or cause troubles for the grid operator or the aggregator in charge of managing their charge. In this scenario, our aggregator, located at node 5 of the 9 bus grid, now owns a 5 MW solar PV farm. Fig. 5 looks at the average balancing requirements of the aggregator when it serves different populations of electric vehicles. One can see that in the absence of the flexible charging load of the EVs, the aggregator incurs high balancing costs due to the solar PVs. As the population of EVs increases, their flexibility can be used to compensate for the intermittency of renewables to a certain extent. However, as the number of EVs increases beyond a certain point, their random nature of charge request

---

**Algorithm 2** Simulation of PHEVs electrical load: scheduling real-time arrivals

1: Set $L(t) = 0$ for all hours $t$
2: Define a set of $5N_D$ queues, each representing a specific charge duration $S_q \in \{1, \ldots, 5\}$ (in hours) and laxity $\gamma_q \in \{1, \ldots, N_D\}$ (in hours)
3: Pick population size (15k vehicles in our case). Pick a subset of NHTS travel pattern data to match population
4: for each hour $t$ of the day do
5: Use the NHTS data to compute the charge duration $S_i$ by 
6: Decide in which queue to place arrivals and update $a_q(t)$
7: Use the DDLS algorithm in [12] to select the optimum $d_q(t)$ for aggregate load $L(t)$ to follow day-ahead bid
8: Additional constraint $d_q(t) > a_q(t - \gamma_q)$ added to the algorithm in [12] ensures deadlines are met
end for
arrivals now begins to present a balancing problem itself.

VII. CONCLUSIONS

In this paper, we illustrated the benefits of having a detailed yet computationally tractable model to incorporate the service offered by an aggregator of deferrable loads in the UC optimization. In the future, this work can be further advanced through several extensions such as incorporating intermittent resources in the UC optimization and defining appropriately the payments to the aggregator for its service, if there should be in fact any extra payments, and the locational marginal prices in the presence of deferrable loads.

REFERENCES


