Data-Driven Modeling of Aggregate Flexibility Under Uncertain and Non-Convex Device Models

Sina Taheri, *Member, IEEE*, Vassilis Kekatos, *Senior Member, IEEE*, Sriharsha Veeramachaneni, and Baosen Zhang, *Member, IEEE*

***Abstract*—Bundling a large number of distributed energy resources through a load aggregator has been advocated as an effective means to integrate such resources into wholesale energy markets. To ease market clearing, system operators allow aggregators to submit bidding models of simple prespecified polytopic shapes. Aggregators need to carefully design and commit to a polytope that best captures their energy flexibility along a day-ahead scheduling horizon. This work puts forth a model-informed data-based optimal flexibility design for aggregators, which deals with the time-coupled, uncertain, and non-convex models of individual loads. The proposed solution first generates efficiently a labeled dataset of (in)-feasible aggregation schedules. The feasible set of the aggregator is then approximated by an ellipsoid upon training a convex quadratic classifier using the labeled dataset. The ellipsoid is subsequently inner approximated by a polytope. Using Farkas' lemma, the obtained polytope is finally inner approximated by the polytopic shape dictated by the market. Numerical tests show the effectiveness of the proposed flexibility design framework for designing the feasible sets of small- and large-sized aggregators coordinating solar photovoltaics, thermostatically-controlled loads, batteries, and electric vehicles. The tests further demonstrate that it is crucial for the aggregator to consider time-coupling and uncertainties in optimal flexibility design.**

***Index Terms*—Feasible set, aggregator, convex quadratic classifier, ellipsoids, containment of polytopes, optimal flexibility design, day-ahead markets, load disaggregation.**

I. Introduction

IN ADDITION to the existing bid models for generators and load serving entities, independent system operators (ISO) are currently accepting new bid models to facilitate the participation of load aggregators [1]. Similar to a battery model, new bidding models could consist of upper and lower limits on power and ramping, but also on energy to capture state-of-charge (SoC)-type of constraints. Such battery models capture how *flexible* the aggregator can be as a dispatchable resource for the ISO in the energy market. On one hand, if an aggregator over-estimates its flexibility and then fails to meet its schedule dispatched by the ISO, it will be penalized. On the other hand, offering more flexibility when participating in the market could increase the financial benefit of the aggregator. Evidently, the aggregator aims at offering the maximal flexibility that can be implemented. To achieve this dual goal, the aggregator needs to carefully design its feasibility set to be submitted to the ISO. This task is defined as *optimal flexibility design* (OFD).

An aggregator controls a diverse set of devices such as solar photovoltaics (PVs), batteries, electric vehicles (EVs), thermostatically controlled loads (TCLs), home appliances (such as dishwashers/dryers), and pool cleaners/pumps [2]. The OFD task is challenging for three reasons that stem from the properties of such devices or load types. First, devices such as EVs and batteries exhibit time-coupling naturally, meaning that load schedules are constrained across successive control periods and cannot be determined independently. Second, the inefficiencies of EVs and batteries and the ON/OFF characteristics of TCLs lead to non-convex models. Third, most devices operate under time-varying externalities, such as solar irradiance for PVs; the initial SoC for batteries; arrival/departure times for EVs; and ambient temperature or occupancy for TCLs. These externalities are inherently random and uncertain when the OFD problem is solved, e.g., at day-ahead. Such uncertainty further complicates OFD and calls for stochastic formulations.

The current literature in OFD can be categorized in two main groups. The first group of research uses geometric techniques to find a *polytope* that captures the aggregate flexibility [3], [4], [5], [6], [7], [8], [9]. A reduced-order model for aggregate flexibility is designed using quantization in [3], yet the approach applies only to devices described by convex linear models of the same shape. Stochastic battery models for load aggregations is put forth in [4], [5], though again they are not applicable to non-convex device models. Presuming externalities to be deterministically known and considering only specific types of devices like batteries and TCLs under averaged linear models, reference [6] models the aggregate flexibility as the Minskowski sum of the convex feasible

Manuscript received 26 January 2022; revised 22 April 2022; accepted 17 June 2022. Date of publication 22 June 2022; date of current version 21 October 2022. This work was supported in part by the NextEra Analytics and in part by the U.S. National Science Foundation under Grant 1751085. Paper no. TSG-00137-2022. *(Corresponding author: Vassilis Kekatos.)*

Sina Taheri and Vassilis Kekatos are with the Bradley Department of ECE, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: st92@vt.edu; kekatos@vt.edu).

Sriharsha Veeramachaneni is with the Applied Mathematics Group, NextEra Analytics, Saint Paul, MN 55107 USA (e-mail: sriharsha.veeramachaneni@nexteraanalytics.com).

Baosen Zhang is with the ECE Department, University of Washington, Seattle, WA 98195 USA (e-mail: zhangbao@uw.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2022.3185532.

Digital Object Identifier 10.1109/TSG.2022.3185532

sets of individual devices, which can be approximated using homothets [7], or zonotopes [8].

The second group of approaches formulates OFD as a multi-level optimization problem [10], [11], [12], [13], [14]. Reference [10] models aggregation flexibility by an ellipsoid to be found via semi-definite programming [10]. Reference [11] considers the cost of flexibility for each device and finds a flexibility cost map, which can be used to find the desired aggregate flexibility region. Nonetheless, the latter approach can handle neither uncertain externalities nor non-convex device models. OFD under linear and deterministic device models has been posed as a multi-stage optimization task also in [12], [13], [14], yet flexibility limits are presumed decoupled across time, which may yield non-implementable aggregator schedules because devices cannot ramp sufficiently fast. Reference [14] treats disaggregation as a random policy over random externalities, but also ignores time-coupling and non-convex device models.

The three aforementioned challenges with load aggregations (namely the time-coupled, non-convex, and stochastic nature of device models) have been considered while modeling the effect of aggregations in demand-response programs and chance-constrained optimal power flow formulations; see [15], [16] and references therein. Nonetheless, to the best of our knowledge, not all three aspects have been considered simultaneously while designing the feasibility set of an aggregator under the ISO specifications. We put forth a data-driven OFD framework that addresses all three challenges. In particular, the contribution of this work is threefold: *1)* Develop a data generation framework that deals with time-coupling in flexibility design, non-convex device models, and uncertain externalities through a chance-constrained formulation (Section III); *2)* Train a convex quadratic classifier to approximate the feasible set of the aggregator by an ellipsoid (Section IV); *3)* Inner approximate the obtained ellipsoid with a polytope and use the geometry of polytopes to reformulate OFD as a linear program (Section V). Section VI evaluates the performance of the proposed flexibility design solution using two aggregators of increasing modeling complexity. Numerical tests demonstrate that flexibility is well-approximated by convex sets, while capturing time-coupling and uncertainty of device models seems to be important.

Regarding *notation*, column vectors (matrices) are denoted by lowercase (uppercase) boldface letters; calligraphic symbols are reserved for sets. The n-th element of $\mathbf{x}$ is denoted by x_n. Symbol $\mathbf{1}$ denotes the all-one vector. Inequalities between vectors, such as $\mathbf{x} \geq \mathbf{y}$, apply entry-wise.

II. Problem Formulation

Consider an aggregator participating in a wholesale electricity market cleared by an independent system operator (ISO). To submit its energy bids for the day-ahead market, the aggregator has to comply with the bidding model for virtual generators and flexible loads supported by the ISO. We will henceforth refer to this bidding model as the *aggregator model*. To describe this model, consider a day-ahead market organized in T scheduling intervals indexed by t, each one of duration δ. The aggregator model consists of limits $(\underline{p}_t, \overline{p}_t)$ on the instantaneous power p_t provided by the aggregator to the grid during period t; and ramping constraints. Such characteristics are standard for conventional market participants. To better facilitate the integration of load aggregations, aggregator models may also include the initial state-of-charge (SoC) s_0 at the beginning of the first period; as well as limits $\{(\underline{s}_t, \overline{s}_t)\}_{t=1}^{T}$ on the SoC s_t at the end of period t for the aggregation. The aggregator model is essentially a battery model augmented by ramping constraints. It can be precisely expressed as:

$$\underline{p}_t \leq p_t \leq \overline{p}_t, \qquad t = 1{:}T \tag{1a}$$

$$s_t = s_{t-1} - \Delta p_t, \qquad t = 1{:}T \tag{1b}$$

$$\underline{s}_t \leq s_t \leq \overline{s}_t, \qquad t = 1{:}T \tag{1c}$$

$$\underline{\alpha}_t \leq p_{t+1} - p_t \leq \overline{\alpha}_t, \qquad t = 1{:}T-1. \tag{1d}$$

Different from the bidding model of conventional generators, the aggregator model imposes limits on SoC via the additional constraints of (1c). Moreover, for conventional thermal generators, capacity and ramping limits are known and typically remain unchanged during normal operations. On the contrary, all limits in (1) may be changing across time and day-by-day. The goal of this work is exactly to find these limits.

Let vectors $(\mathbf{p}, \underline{\mathbf{p}}, \overline{\mathbf{p}})$ collect the instantaneous power and its limits across all scheduling intervals; vectors $(\underline{\mathbf{s}}, \overline{\mathbf{s}})$ collect SoC limits; and vectors $(\underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}})$ collect the ramping rates. The aggregator model is described by *model variables* $(\underline{\mathbf{p}}, \overline{\mathbf{p}}, s_0, \underline{\mathbf{s}}, \overline{\mathbf{s}}, \underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}})$. To submit a bid to the ISO, the aggregator has to carefully select these variables. Upon collecting bids from all market participants, the ISO clears the market to satisfy demand while ensuring network and reliability constraints. The ISO subsequently informs market participants of their schedules. For the aggregator of interest, the schedule decided by the ISO is denoted by $\mathbf{p}^* \in \mathbb{R}^T$. This schedule satisfies constraints (1) by design. A key objective for the aggregator is to ensure that the scheduled $\mathbf{p}^*$ can be actually realized, i.e., there exist devices whose dispatches across the scheduling horizon sum up to $\mathbf{p}^*$. Such schedule will be henceforth termed *disaggregatable* or *feasible*.

In essence, the aggregator model variables define the feasible set wherein the aggregator schedule variable $\mathbf{p}$ (and hence $\mathbf{p}^*$) can lie. To express that set in a compact form, eliminate the SoC variables and rewrite (1) as the polytope

$$\mathcal{P}(\mathbf{x}) := \{\mathbf{p} : \mathbf{G}\mathbf{p} \leq \mathbf{x}\} \tag{2}$$

where vector $\mathbf{x}$ depends on the aggregator model variables as

$$\mathbf{x} := \left\{\overline{\mathbf{p}}, -\underline{\mathbf{p}}, \overline{\mathbf{s}} - s_0\mathbf{1}, -\underline{\mathbf{s}} + s_0\mathbf{1}, \overline{\boldsymbol{\alpha}}, \underline{\boldsymbol{\alpha}}\right\} \tag{3}$$

and the $(6T-2) \times T$ matrix $\mathbf{G}$ is defined as

$$\mathbf{G}^\top := \begin{bmatrix} +\mathbf{I}^\top & -\mathbf{I}^\top & +\Delta\mathbf{L}^\top & -\Delta\mathbf{L}^\top & +\mathbf{K}^\top & -\mathbf{K}^\top \end{bmatrix} \tag{4}$$

where $\mathbf{I}$ is the identity matrix of size T; matrix $\mathbf{L}$ is a $T \times T$ lower triangular matrix with all ones on its lower triangular part; and $\mathbf{K}$ is a $(T-1) \times T$ difference matrix. Matrix $\mathbf{K}$ takes the value of -1 on its main diagonal; the value of $+1$ on the first above the main diagonal; and zero, otherwise. Because $\mathbf{G}$ is known and fixed, the ISO only needs to know $\mathbf{x}$ to dispatch

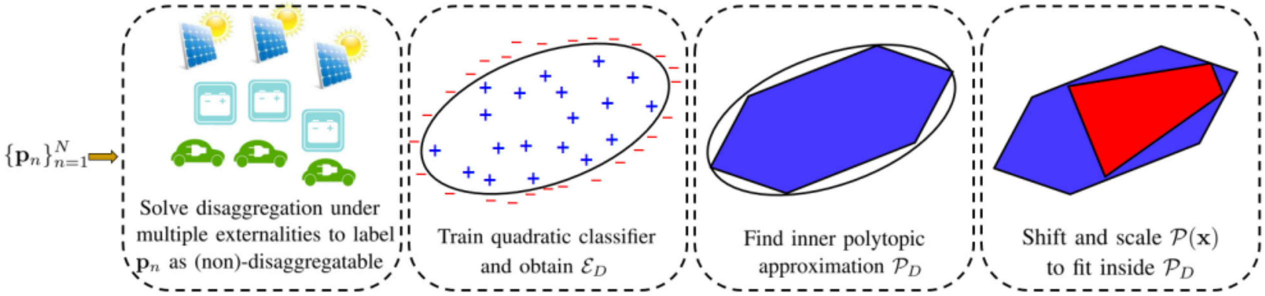

$\{\mathbf{p}_n\}_{n=1}^N$
Solve disaggregation under multiple externalities to label $\mathbf{p}_n$ as (non)-disaggregatable
Train quadratic classifier and obtain $\mathcal{E}_D$
Find inner polytopic approximation $\mathcal{P}_D$
Shift and scale $\mathcal{P}(\mathbf{x})$ to fit inside $\mathcal{P}_D$

Fig. 1. Solving the OFD task in four steps: *S1)* Generating a labeled dataset $\mathcal{D}$ of (non)-disaggregatable aggregator schedules by solving disaggregation for multiple instances of externalities $\boldsymbol{\omega}$; *S2)* Training a convex quadratic classifier to discern disaggregatable schedules and obtaining ellipsoid $\mathcal{E}_D$; *S3)* Inner approximating $\mathcal{E}_D$ with a polytope $\mathcal{P}_D$; *S4)* Finding a maximum volume polytope $\mathcal{P}(\mathbf{x})$ for a parametric form of $\mathbf{x}$ to inner approximate $\mathcal{P}_D$.

the aggregator. It is therefore fair to say that $\mathbf{x}$ captures the aggregator flexibility.

While designing $\mathbf{x}$, the aggregator targets two goals:

i) Ensure that every $\mathbf{p} \in \mathcal{P}(\mathbf{x})$ can be actually implemented by the available resources to avoid penalties; and

ii) Maximize its flexibility while participating in the electricity market.

The latter can be measured by the volume $V(\mathbf{x})$ of polytope $\mathcal{P}(\mathbf{x})$. The task of maximizing flexibility while guaranteeing feasible disaggregation is henceforth termed *optimal flexibility design* (OFD), and can be posed as the optimization

$$\max_{\mathbf{x}} \quad V(\mathbf{x}) \tag{5a}$$

$$\text{s.to} \quad \mathcal{P}(\mathbf{x}) \subseteq \mathcal{F} \tag{5b}$$

where $\mathcal{F}$ is the set of disaggregatable $\mathbf{p}$'s.

It is worth stressing that in addition to $\mathcal{P}(\mathbf{x})$, an aggregator may also wish to optimize its bids. These would be the (convex piece-wise) linear costs per scheduling period t. In fact, the costs and feasible set submitted to the ISO can be jointly optimized by pursuing a strategic investment approach; see [17] and references therein. Depending on the setup, strategic investment can be quite challenging as it is typically posed as a bilevel program over multiple market scenarios. This work does not consider strategic investment although knowing $\mathcal{P}(\mathbf{x})$ is a prerequisite for designing an optimal bidding strategy. Notice also that $\mathcal{P}(\mathbf{x})$ may not necessarily be the feasible set submitted to the ISO.

Problem (5) entails two challenges: First, constraint (5b) is abstract and it is not obvious how it can handled. Second, finding $V(\mathbf{x})$ is hard in general and bears no closed-form mathematical expression. To cope with the first challenge, we take a data-based approach involving the four steps as illustrated in Fig. 1:

S1) Construct a labeled dataset $\mathcal{D} := \{(\mathbf{p}_n, y_n)\}_{n=1}^N$, where label $y_n = -1$ when $\mathbf{p}_n$ is disaggregatable; and $y_n = +1$, otherwise (Section III and the Appendix).

S2) Use dataset $\mathcal{D}$ to approximate the set $\mathcal{F}$ of disaggregatable schedules by an ellipsoid $\mathcal{E}_D$ (Section IV).

S3) Find a polytope $\mathcal{P}_D$ that inner approximates $\mathcal{E}_D$ to arbitrary accuracy (Section V-A).

S4) Design a parametric form for $\mathbf{x}$ and design it so that $\mathcal{P}(\mathbf{x}) \subseteq \mathcal{P}_D$; see Section V-B.

Interestingly, step *S4)* introduces a variable that is proportional to the volume of $\mathcal{P}(\mathbf{x})$. By maximizing this variable, we obviate the need of handling volume $V(\mathbf{x})$ explicitly, and thus, address the second challenge. Steps *S1)–S4)* are delineated in the next sections.

III. Data Generation

We commence with step *S1)* of generating the labeled dataset $\mathcal{D}$. Note that $\mathcal{D}$ may be available to the aggregator from historical data, in which case, step *S1)* is not needed. If historical data are not available or insufficient in numbers, the aggregator can generate training examples by sampling $\mathbf{p}$'s and labeling them. This section explains how $\mathbf{p}$'s can be sampled and labeled efficiently. A given $\mathbf{p}$ is disaggregatable in a deterministic sense if the disaggregation task presented next yields a zero optimal objective value:

$$g(\mathbf{p}; \boldsymbol{\omega}) := \min_{\{\mathbf{p}^d\}} \ \|\mathbf{p} - \sum_{d=1}^{D} \mathbf{p}^d\| \tag{6a}$$

$$\text{s.to} \quad \mathbf{p}^d \in \mathcal{F}^d(\boldsymbol{\omega}^d), \quad d = 1 : D \tag{6b}$$

for a vector norm $\|\cdot\|$. Here D is the number of devices controlled by the aggregator. Each vector $\mathbf{p}^d$ collects the dispatch decisions for device d across all times. The feasible set $\mathcal{F}^d(\boldsymbol{\omega}^d)$ captures the operational constraints $\mathbf{p}^d$ should satisfy, including for example temperature limits for TCLs, apparent power constraints for PVs, and SoC and power limits for batteries. Each set $\mathcal{F}^d(\boldsymbol{\omega}^d)$ depends on a vector of external parameters $\boldsymbol{\omega}^d$, such as local ambient temperature for TCLs, solar irradiance for PVs, initial SoC for batteries and EVs, and arrival/departure times for EVs. Let vector $\boldsymbol{\omega}$ concatenate all externalities $\boldsymbol{\omega}^d$'s. The challenges are that (6) may not be convex and $\boldsymbol{\omega}$ is random.

If sets $\mathcal{F}^d(\boldsymbol{\omega}^d)$ are convex in $\mathbf{p}^d$, problem (6) can be reformulated as a convex program [18]. That is the case for inverter-interfaced devices or TCLs with average models, and energy storage devices without inefficiencies. On the other hand, the operation of TCLs and batteries with inefficiencies introduce non-convex feasible sets involving binary variables, which render (6) a mixed-integer program. Regardless convex or not, problem (6) is amenable to decomposition techniques along devices similarly to unit commitment-type of problems encountered in transmission systems.

Schedule $\mathbf{p}$ is disaggregatable for the particular $\boldsymbol{\omega}$ only when $g(\mathbf{p};\boldsymbol{\omega})=0$. Since the aggregator determines its flexibility in a day-ahead setting, vector $\boldsymbol{\omega}$ is uncertain. To account for this uncertainty, rather than solving (6) for a single realization of $\boldsymbol{\omega}$, the aggregator would like to determine if $\mathbf{p}$ is disaggregatable with high probability over random $\boldsymbol{\omega}$. Verifying this property may entail solving a problem with complex chance constraints over a possibly unknown probability distribution of $\boldsymbol{\omega}$, which can be computationally intractable.

To arrive at a practical solution, we resort to a sample approximation of the probability of $\mathbf{p}$ being disaggregatable over uncertain $\boldsymbol{\omega}$'s. In detail, suppose the aggregator has access to a collection of externality scenarios. For each $\mathbf{p}$, one randomly samples $K=25$ externality scenarios and solves (6) for each one of them. The sought probability of $\mathbf{p}$ being disaggregatable can be approximated by the sample statistic

$$c(\mathbf{p};\Omega_K)=\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(g(\mathbf{p};\boldsymbol{\omega}_k)=0) \tag{7}$$

where $\mathbb{I}$ is the indicator function returning one when $g(\mathbf{p};\boldsymbol{\omega}_k)=0$; and zero, otherwise. The reason to use only the scenarios in Ω_K rather than all scenarios in Ω_R is that solving (6) for R scenarios can be computationally demanding. Based on (7), we can now define when a schedule is disaggregatable in a probabilistic sense.

Definition 1: Schedule $\mathbf{p}$ is deemed as disaggregatable (feasible) if $c(\mathbf{p};\Omega_K)\geq 1-\epsilon$ for a given small $\epsilon\geq 0$; and non-disaggregatable (infeasible), otherwise.

To recap, data generation step *S1)* involves sampling a $\mathbf{p}_n$ and labeling it as feasible (disaggregatable) or not per Definition 1. In detail, labeling $\mathbf{p}_n$ entails drawing K externality scenarios $\boldsymbol{\omega}_k$, solving the disaggregation problem (6) independently for each $\boldsymbol{\omega}_k$ to compute $g(\mathbf{p}_n,;\boldsymbol{\omega}_k)$, and computing the statistic $c(\mathbf{p}_n;\Omega_K)$ from (7). At the end, schedule $\mathbf{p}_n$ is labeled as feasible if $c(\mathbf{p}_n;\Omega_K)\geq 1-\epsilon$ for say $\epsilon=0.05$.

For the classification purposes proposed in the next section, dataset $\mathcal{D}$ should be approximately *balanced*, that is disaggregatable (feasible) and non-disaggregatable (infeasible) examples $\mathbf{p}_n$'s should be similar in numbers. The Appendix develops a method that generates efficiently a roughly balanced dataset $\mathcal{D}$. Our dataset $\mathcal{D}$ should be approximately balanced for three reasons. First, if a classifier is trained with an unbalanced dataset, it may be biased towards the majority class. If for instance, the ratio of infeasible-to-feasible examples is 90% to 10%, a classifier predicting always infeasible would have the seemingly great prediction accuracy of 90%. Second, feasible examples are expected to be surrounded by infeasible examples. This is because the set of feasible examples is anticipated to be approximated well by a convex set as discussed in the next section. Therefore, infeasible examples lying away from the decision boundary would not really contribute to shaping the classification rule. Third, for a general classification task intentionally biasing the process of sampling examples may raise concerns of altering the true data distribution. Such concerns are waived in our case as we deal with a purely geometrical problem where one wants to sample points within and around an approximately convex set. If the location of the set is not known *a priori*, one may start sampling over a significantly wider area, but then zoom into the area of interest.

Note by solving (6) for a pair $(\mathbf{p},\boldsymbol{\omega})$, not only we decide whether $g(\mathbf{p};\boldsymbol{\omega}_k)$ is zero; we can also compute the point

$$\hat{\mathbf{p}}(\boldsymbol{\omega})=\sum_{d=1}^{D}\hat{\mathbf{p}}^d(\boldsymbol{\omega}) \tag{8}$$

where $\{\hat{\mathbf{p}}^d(\boldsymbol{\omega})\}_{d=1}^{D}$ is the minimizer of (6). Although schedule $\hat{\mathbf{p}}(\boldsymbol{\omega})$ can be disaggregated into device schedules for this specific $\boldsymbol{\omega}$, it is not necessarily disaggregatable in the strict sense used throughout this work as it may not satisfy the chance probability of $c(\hat{\mathbf{p}}(\boldsymbol{\omega});\Omega_K)\geq 1-\epsilon$. Schedule $\hat{\mathbf{p}}(\boldsymbol{\omega})$ will be useful in the sampling process described in the Appendix.

As a final remark, the uncertainty scenarios used to produce the dataset are assumed to be sufficiently representative of the actual conditions to be experienced by the device aggregation over the day-ahead horizon. Otherwise, the computed flexibility set can be subpar. That is an inherent difficulty with any day-ahead scheduling task (e.g., predicting wind farm generation in a day-ahead market), and goes beyond the scope of this work. Obviously, to account for increased uncertainty, an aggregator could draw $\boldsymbol{\omega}_k$'s from a distribution of larger variance, but that trades safety for conservativeness in determining the flexibility set. An interesting open question is endowing the feasible set $\mathcal{P}(\mathbf{x})$ with probabilistic guarantees, so that each $\mathbf{x}$ comes with a confidence interval quantifying the chances of it being infeasible. Here we consider the easier problem of finding a deterministic $\mathcal{P}(\mathbf{x})$.

IV. Learning Feasible Aggregations From Data

This section uses dataset $\mathcal{D}$ to approximate $\mathcal{F}$ using a convex quadratic classifier. Recent research shows that aggregations of large number of devices with individually non-convex feasible sets can be closely approximated by a convex set; see e.g., [9] and references therein. However, even if $\mathcal{F}$ is non-convex, an ISO would only accept polytopic descriptions for the feasible set of an aggregator to ease its scheduling operations; see [1], [9]. Therefore, designing a convex classifier as a first step to best capture $\mathcal{F}$ is reasonable. Due to the presence of uncertain externalities $\boldsymbol{\omega}$ and the possibly non-convex nature of $\mathcal{F}^d$'s, computing an explicit expression for $\mathcal{F}$ is a formidable task. Nonetheless, we can use data to approximate $\mathcal{F}$ with a convex set $\hat{\mathcal{F}}$. Set $\hat{\mathcal{F}}$ can be expressed as the 0-sublevel set of a convex function $d(\mathbf{p})$ as $\hat{\mathcal{F}}:=\{\mathbf{p}:d(\mathbf{p})\leq 0\}$. Function $d(\mathbf{p})$ acts as a *classifier* to decide if $\mathbf{p}$ is disaggregatable ($d(\mathbf{p})\leq 0$) or not ($d(\mathbf{p})>0$).

Prior to learning $d(\mathbf{p})$ from $\mathcal{D}$, the aggregator needs to choose a functional form for $d(\mathbf{p})$. A linear classifier of the form $d(\mathbf{p})=\mathbf{w}_1^\top\mathbf{p}+w_0$ is the simplest option, but may have limited representation capabilities. A more complex classifier would be a convex quadratic function. Such classifier has been shown to be effective for capturing chance constraints [19], which motivated us to chose the same option as the set $\mathcal{F}$ that we are trying to capture here is the feasible set of a chance constraint on $\mathbf{p}$ as well. While there may be more sophisticated options, such as multi-linear and neural network-based classifiers, they are left for future research. We proceed with training the convex quadratic classifier

$$d(\mathbf{p}) := \mathbf{p}^\top \mathbf{W}_2 \mathbf{p} + \mathbf{w}_1^\top \mathbf{p} + w_0 \tag{9}$$

where $\mathbf{W}_2 \succeq 0$ is a symmetric positive semidefinite matrix. The classifier is linear in $(\mathbf{W}_2, \mathbf{w}_1, w_0)$ and convex quadratic in $\mathbf{p}$. It approximates $\mathcal{F}$ by the ellipsoid

$$\mathcal{E}_D := \left\{\mathbf{p} : \mathbf{p}^\top \mathbf{W}_2 \mathbf{p} + \mathbf{w}_1^\top \mathbf{p} + w_0 \leq 0\right\}. \tag{10}$$

The classifier parameters can be learned from dataset $\mathcal{D}$ by solving the optimization

$$\min \ \frac{1}{N}\sum_{n=1}^{N}\left[1 - y_n d(\mathbf{p}_n)\right]_+ + \lambda\left(\|\mathbf{W}_2\|_F^2 + \|\mathbf{w}_1\|_2^2\right) \tag{11a}$$

$$\text{over} \ \ \mathbf{W}_2 \succeq 0, \mathbf{w}_1, w_0 \tag{11b}$$

where $[x]_+ := \max(0, x)$. An ideal classifier should yield $d(\mathbf{p}_n) < 0$ if $y_n = -1$ (feasible $\mathbf{p}_n$); and $d(\mathbf{p}_n) \geq 0$ if $y_n = +1$ (infeasible $\mathbf{p}_n$). In other words, an ideal classifier should satisfy $y_n d(\mathbf{p}_n) \geq 0$ for all n. Therefore, a classifier should be ideally trained upon minimizing the loss function $\sum_{n=1}^{N} [1 - \text{sgn}(y_n d(\mathbf{p}_n))]$, where sgn returns 1 for a positive argument; and 0, otherwise. Unfortunately, function sgn is discontinuous, and thus, hard to minimize. The standard approach in classification is to surrogate the aforesaid loss function with the so-termed hinge loss $[1 - y_n d(\mathbf{p}_n)]_+$, which is convex; see [20, Fig. 7.5] for details. To avoid overfitting, the training problem would typically penalize large values for the classifier weights in the ell_2-norm excluding the intercept w_0; see [20, Ch. 7]. Parameter $\lambda > 0$ balances the trade-off between the hinge classification cost and the regularization terms; and was tuned using cross-validation. Problem (11) can be reformulated to a semidefinite program (SDP).

Note that $\mathcal{E}_D$ serves only as a surrogate for $\mathcal{F}$, and cannot be claimed to be an inner or outer approximation of $\mathcal{F}$. This is because $\mathcal{F}$ may be non-convex, while $\mathcal{E}_D$ has been learned from data and using the sample approximation in (7). We next explain how an inner polytopic approximation of $\mathcal{E}_D$ can be used to approximate $\mathcal{P}(\mathbf{x}) \subseteq \mathcal{F}$ in (5b).

V. OFD AS CONTAINMENT OF POLYTOPES

Since $\mathcal{E}_D$ surrogates set $\mathcal{F}$, the constraint $\mathcal{P}(\mathbf{x}) \subseteq \mathcal{F}$ in (5b) can be approximated by

$$\mathcal{P}(\mathbf{x}) \subseteq \mathcal{E}_D \text{ or } \max_{\mathbf{p}:\mathbf{G}\mathbf{p}\leq\mathbf{x}} d(\mathbf{p}) \leq 0. \tag{12}$$

Problem (12) involves maximizing a convex quadratic function over a polytope. To bypass this non-convexity issue, we proceed in two steps: First find a polytope $\mathcal{P}_D$ that is inscribed in $\mathcal{E}_D$, and then design $\mathbf{x}$ so that $\mathcal{P}(\mathbf{x}) \subset \mathcal{P}_D \subset \mathcal{E}_D$.

A. Polytopic Inner Approximation of an Ellipsoid

We first adopt the approach of [21], [22], according to which the ellipsoid $\mathcal{E}_T^r := \{\mathbf{y} \in \mathbb{R}^T : \|\mathbf{y}\|_2 \leq r\}$ is approximated by a polytope $\mathcal{P}_T^\delta$ within accuracy δ in the sense

$$\mathcal{E}_T^{r/(1+\delta)} \subset \mathcal{P}_T^\delta \subset \mathcal{E}_T^r. \tag{13}$$

Polytope $\mathcal{P}_T^\delta$ is defined over the original variables $\mathbf{y}$ and a vector of auxiliary variables $\mathbf{q}$ as

$$\mathcal{P}_T^\delta := \left\{\mathbf{y} : \mathbf{E}_1\mathbf{y} + \mathbf{E}_2\mathbf{q} \leq \bar{\mathbf{d}} \text{ for some } \mathbf{q}\right\}.$$

The way $(\mathbf{E}_1, \mathbf{E}_2, \bar{\mathbf{d}})$ are determined is summarized in [23]. The number of auxiliary variables in $\mathbf{q}$ and the number of linear constraints in $\mathcal{P}_T^\delta$ scale logarithmically with δ.

Ellipsoid $\mathcal{E}_D$ in (10) can be converted to form $\mathcal{E}_T^r$ by setting

$$\mathbf{y} := \mathbf{W}_2^{1/2}\mathbf{p} + \mathbf{W}_2^{-1/2}\mathbf{w}_1 \text{ and } r := \mathbf{w}_1^\top \mathbf{W}_2^{-1}\mathbf{w}_1 - w_0$$

where $\mathbf{W}_2^{1/2}$ is the matrix square root of $\mathbf{W}_2$. Then, ellipsoid $\mathcal{E}_D$ can be inner approximated by the polytope

$$\mathcal{P}_D = \left\{\mathbf{p} : \mathbf{E}_1\left(\mathbf{W}_2^{1/2}\mathbf{p} + \mathbf{W}_2^{-1/2}\mathbf{w}_1\right) + \mathbf{E}_2\mathbf{q} \leq \bar{\mathbf{d}} \text{ for a } \mathbf{q}\right\}.$$

In words, a point $\mathbf{p}$ belongs to $\mathcal{P}_D$ if there exists a $\mathbf{q}$ satisfying the aforesaid linear inequalities. In essence, polytope $\mathcal{P}_D$ is the projection of a polytope in $(\mathbf{p}, \mathbf{q})$ onto the space of the variables $\mathbf{p}$ alone. Because such representation of $\mathcal{P}_D$ is not convenient for future developments, we next eliminate $\mathbf{q}$. This projection operation is in general computationally hard given the polytope in $(\mathbf{p}, \mathbf{q})$ is described in its vertex representation. Nevertheless, for moderate lengths of $\mathbf{p}$ and $\mathbf{q}$, one can use the Fourier-Motzkin algorithm [24]. This algorithm eliminates $\mathbf{q}$ by generating additional linear constraints on $\mathbf{p}$. The projection is exact in the sense that one eventually gets the next equivalent representation of $\mathcal{P}_D$ for given $(\mathbf{E}, \mathbf{d})$:

$$\mathcal{P}_D = \{\mathbf{p} : \mathbf{E}\mathbf{p} \leq \mathbf{d}\}. \tag{14}$$

B. Reformulating OFD Using Farkas' Lemma

Having found a convenient representation for $\mathcal{P}_D$, we can now approximate the OFD problem in (5) as

$$\max_{\mathbf{x}} \ V(\mathbf{x}) \tag{15a}$$

$$\text{s.to} \ \mathcal{P}(\mathbf{x}) \subseteq \mathcal{P}_D. \tag{15b}$$

Recall $V(\mathbf{x})$ is the volume of $\mathcal{P}(\mathbf{x})$. We handle constraint (15b) upon invoking a version of Farkas' lemma on the containment of polytopes as presented in [25], [26].

Lemma 1 (Farkas' Lemma): Consider two non-empty polyhedra $\mathcal{P}(\mathbf{x}) = \{\mathbf{p} : \mathbf{G}\mathbf{p} \leq \mathbf{x}\}$ and $\mathcal{P}_D := \{\mathbf{p} : \mathbf{E}\mathbf{p} \leq \mathbf{d}\}$. It holds that $\mathcal{P}(\mathbf{x}) \subseteq \mathcal{P}_D$ if and only if there exists matrix $\mathbf{F} \geq \mathbf{0}$ satisfying $\mathbf{F}\mathbf{G} = \mathbf{E}$ and $\mathbf{F}\mathbf{x} \leq \mathbf{d}$.

Using Lemma 1, problem (15a) can be reformulated as

$$\max_{\mathbf{x},\mathbf{F}\geq 0} \ V(\mathbf{x}) \tag{16a}$$

$$\text{s.to} \ \mathbf{F}\mathbf{G} = \mathbf{E} \tag{16b}$$

$$\mathbf{F}\mathbf{x} \leq \mathbf{d}. \tag{16c}$$

Due to the product $\mathbf{F}\mathbf{x}$, problem (16) remains non-convex, while we still lack a good choice for $V(\mathbf{x})$.

To resolve these two issues, we resort to a restriction of (16). We parameterize the sought vector $\mathbf{x}$ as

$$\mathbf{x} = \frac{1}{\beta}(\bar{\mathbf{x}} - \mathbf{G}\mathbf{z}) \tag{17}$$

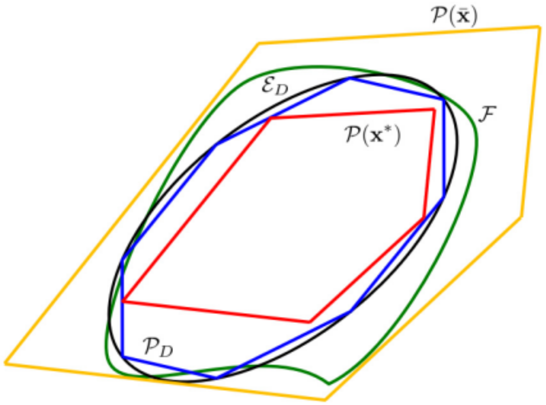


Fig. 2. The non-convex original set $\mathcal{F}$; the ellipsoid $\mathcal{E}_D$ being a data-based estimate of $\mathcal{F}$; set $\mathcal{P}_D$ is the inner polytopic approximation of $\mathcal{E}_D$; the prototype polytope $\mathcal{P}(\bar{\mathbf{x}})$; and the solution polytope $\mathcal{P}(\mathbf{x}^*)$. Notice that sets $\mathcal{P}(\mathbf{x}) \subseteq \mathcal{P}_\mathcal{D}$ and $\mathcal{P}_\mathcal{D} \subset \mathcal{E}_\mathcal{D}$ are ensured to be contained in $\mathcal{F}$, whereas $\mathcal{E}_D$ and consequently $\mathcal{P}(\bar{\mathbf{x}})$ may not. This is because the ellipsoid $\mathcal{E}_\mathcal{D}$ is constructed via learning from labeled data (classification).

where $\mathbf{z}$ and $\beta > 0$ are to be designed, whereas $\bar{\mathbf{x}}$ is given. Apparently, the parameterization would have been inconsequential had $\bar{\mathbf{x}}$ been left free. But then the inherent computational complexity would remain. For this reason, parameter $\bar{\mathbf{x}}$ (serving as the 'center' of the polytope) is fixed to some prior guess and $\mathbf{x}$ is confined to lie in the affine set $\bar{\mathbf{x}} - \mathbf{Gz}$ scaled by β for $(\mathbf{z}, \beta)$ to be found.

Thanks to this form, constraint (16c) can be written as

$$\frac{1}{\beta}\mathbf{F}(\bar{\mathbf{x}} - \mathbf{Gz}) \leq \mathbf{d} \quad \Leftrightarrow \quad \mathbf{F}\bar{\mathbf{x}} \leq \mathbf{FGz} + \beta\mathbf{d} = \mathbf{Ez} + \beta\mathbf{d}$$

using the key observation that $\mathbf{FG} = \mathbf{E}$ from (16b).

To better understand the proposed restriction, consider polytope $\mathcal{P}(\bar{\mathbf{x}}) := \{\mathbf{p} : \mathbf{Gp} \leq \bar{\mathbf{x}}\}$. Evidently under the restriction of (17), if $\bar{\mathbf{p}} \in \mathcal{P}(\bar{\mathbf{x}})$, then $\mathbf{p} := (\bar{\mathbf{p}} - \mathbf{z})/\beta \in \mathcal{P}(\mathbf{x})$. This means that if we take any point in $\mathcal{P}(\bar{\mathbf{x}})$, shift it by $-\mathbf{z}$, and scale it by $1/\beta$, we obtain $\mathcal{P}(\mathbf{x})$. In other words, the polytope $\mathcal{P}(\mathbf{x})$ is selected to be a shifted and scaled version of the 'prototype' polytope $\mathcal{P}(\bar{\mathbf{x}})$; see also Fig. 2. The shape of the prototype polytope depends on $\bar{\mathbf{x}}$, while its shifting and scaling parameters can be optimally selected.

Adopting the restriction of (17) simplifies the task of dealing with volume $V(\mathbf{x})$. Because $\mathcal{P}(\mathbf{x}) \subset \mathbb{R}^T$ is a shifted and scaled replica of $\mathcal{P}(\bar{\mathbf{x}}) \subset \mathbb{R}^T$, it holds that

$$V(\mathbf{x}) = V(\bar{\mathbf{x}})/\beta^T \tag{18}$$

where $V(\bar{\mathbf{x}})$ is the fixed volume of $\mathcal{P}(\bar{\mathbf{x}})$. Consequently, under (17), problem (16) simplifies to the linear program

$$\left(\mathbf{F}^*, \mathbf{z}^*, \beta^*\right) \in \arg \min_{\mathbf{F}\geq\mathbf{0},\mathbf{z},\beta\geq 0} \ \beta \tag{19a}$$
$$\text{s.to} \ \ \mathbf{FG} = \mathbf{E} \tag{19b}$$
$$\mathbf{F}\bar{\mathbf{x}} \leq \mathbf{Ez} + \beta\mathbf{d}. \tag{19c}$$

The final answer to the OFD problem is the aggregator model $\mathbf{Gp} \leq \mathbf{x}^*$ with

$$\mathbf{x}^* = \frac{1}{\beta^*}\left(\bar{\mathbf{x}} - \mathbf{Gz}^*\right) \tag{20}$$

TABLE I
TRAINING RESULTS

	Probability of infeasible aggregator schedule ϵ			
metric	0	0.04	0.08	0.12
convexity metric M_1	0.028%	0.145%	0.183%	0.224%
convexity metric M_2	0.0%	0.0%	0.0%	0.0%
training accuracy	97.5%	95.2%	92.5%	91.6%
validation accuracy	98.6%	94.6%	92.0%	91.1%
cond. number of $\mathbf{W}_2$	5.91	5.27	5.05	4.66

Finally, for the parameterization in (17), we need a suitable choice for $\bar{\mathbf{x}}$. One heuristic would be to find $\bar{\mathbf{x}}$ as the minimum-norm $\mathbf{x}$ for which $\mathcal{P}(\bar{\mathbf{x}})$ contains all disaggregatable schedules in dataset $\mathcal{D}$ by solving

$$\bar{\mathbf{x}} = \arg \min_{\mathbf{x}} \ \ \|\mathbf{x}\|_2^2 \tag{21a}$$
$$\text{s.to} \ \ \mathbf{Gp}_n \leq \mathbf{x}, \quad \forall \ \mathbf{p}_n \in \mathcal{D} \text{ with } y_n = -1. \tag{21b}$$

Heed that $\mathcal{P}(\bar{\mathbf{x}})$ for the aforesaid choice of $\bar{\mathbf{x}}$ does contain all feasible points in $\mathcal{D}$, but it may not lie inside $\mathcal{P}_D$ and may contain infeasible points too as illustrated in Fig. 2. Table I summarizes the steps of the proposed OFD approach.

VI. Numerical Tests

A. Simple Illustrative Example

The proposed methodology was first evaluated under a synthetic setup, which has been oversimplified for the purposes of visualization and drawing intuition. This setup involves only $T = 2$ control periods, while externalities are deterministic. Moreover, the actual feasible set $\mathcal{F}$ of disaggregatable schedules is convex. It in fact matches the aggregator model $\mathcal{P}(\mathbf{x}) := \{\mathbf{p} : \mathbf{Gp} \leq \mathbf{x}\}$ postulated by the ISO in (2). The actual $\mathbf{x}$ can be computed from (3) by setting

$$\overline{\mathbf{p}} = -\underline{\mathbf{p}} = \overline{\mathbf{s}} = \mathbf{1}_2, \ \ \underline{\mathbf{s}} = \mathbf{0}_2, \ \ s_0 = 0.5, \ \ \text{and} \ \overline{\alpha} = -\underline{\alpha} = 1.$$

The feasible set $\mathcal{F}$ is shown in Fig. 3a. Note that out of the $6T - 2 = 10$ linear inequalities in $\mathcal{P}(\mathbf{x})$, only 6 are binding for the particular $\mathbf{x}$. Of course, the aggregator does not know that $\mathcal{F}$ is actually $\mathcal{P}(\mathbf{x})$. It can only draw points $\mathbf{p} \in \mathbb{R}^2$ and test whether they are disaggregatable. The goal for the aggregator is to fit $\mathcal{P}(\mathbf{x})$ onto data by finding $\mathbf{x}$ to maximize its flexibility.

We first followed the process described in Section III and the Appendix to generate a dataset $\mathcal{D}$ of $N = 500$ points shown in Fig. 3b. We set parameter $\kappa = 0.2$; see the Appendix for details. Observe there are pairs of (in-)feasible points lying close to the boundary of $\mathcal{F}$; infeasible points away from the boundary of $\mathcal{F}$; and feasible points on the interior of $\mathcal{F}$. We then trained a classifier per (11) with $\lambda = 10^{-5}$. The obtained ellipsoid $\mathcal{E}_D$ is also shown on Fig. 3b. Ellipsoid $\mathcal{E}_D$ closely approximates $\mathcal{F}$ as it achieves 99.68% accuracy on the training data. We next followed the steps of Section V-A with $\delta = 0.1$ and first obtained the polytope $\mathcal{P}_D \subset \mathcal{E}_D$ with $\mathbf{q} \in \mathbb{R}^2$ and then used the Fourier-Motzkin algorithm to convert $\mathcal{P}_D$ to the form of (14). We subsequently ran Algorithm 1 and obtained the solution polytope $\mathcal{P}(\mathbf{x}^*)$. The containment of $\mathcal{P}(\mathbf{x}^*)$ in $\mathcal{P}_D$ is depicted in Fig. 3c. This figure shows that the recovered polytope $\mathcal{P}(\mathbf{x}^*)$ is very close to the original feasible set $\mathcal{F}$.

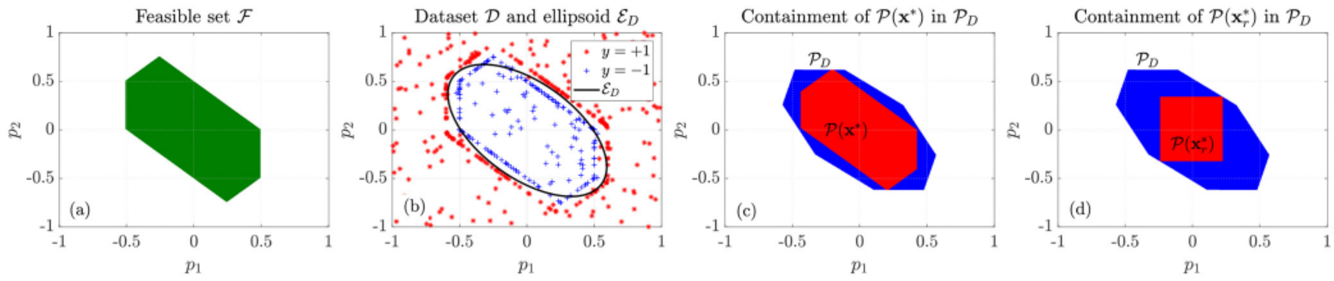


Fig. 3. *(a)* The feasible set $\mathcal{F}$ of an aggregator matches with the model $\mathbf{Gp} \leq \mathbf{x}$ of (2) postulated by the ISO. *(b)* We sampled $N = 500$ labeled datapoints (schedules) to train a convex quadratic classifier and obtained ellipsoid $\mathcal{E}_D$. *(c)* Inner polytopic approximation $\mathcal{P}_D$ of $\mathcal{E}_D$, and the final solution $\mathcal{P}(\mathbf{x}^*)$. *(d)* Optimal solution $\mathcal{P}(\mathbf{x}_r^*)$ if the ISO simplifies the model $\mathbf{Gp} \leq \mathbf{x}$ to account only for box constraints and ignore any time coupling. Notice that $\mathcal{P}(\mathbf{x}^*)$ has a much larger volume than $\mathcal{P}(\mathbf{x}_r^*)$, which indicates the importance of more elaborate aggregator models in OFD.

Algorithm 1 Optimal Flexibility Design (OFD)

```
Input: Dataset of (dis)-aggregatable schedules $\mathcal{D}$.
Output: Maximum-volume polytope $\mathcal{P}(\mathbf{x}^*)$.
Train quadratic classifier $d(\mathbf{p})$ to find ellipsoid $\mathcal{E}_D$ by solving (11).
Find polytope $\mathcal{P}_D \subset \mathcal{E}_D$ in the $(\mathbf{p}, \mathbf{q})$ representation as described in [23].
Convert $\mathcal{P}_D$ in the form of (14) via the Fourier-Motzkin algorithm.
Compute $\bar{\mathbf{x}}$ by solving (21).
Solve (19) and compute the optimal $\mathbf{x}^*$ from (17).
```

Previous works considered only the lower/upper power limits in (1a) [18]. By ignoring ramping and SoC limits, they have postulated a simpler aggregator model. To demonstrate the effect of simpler aggregator models, we solved (19) again but this time with tried to find a polytope defined by the simpler $\mathbf{G}^\top = [\mathbf{I}^\top \ -\mathbf{I}^\top]$ yielding a hyper-rectangle. Figure 3d illustrates the results for this test. The volume of $\mathcal{P}(\mathbf{x}_r^*)$ is much smaller than the volume of the polytope $\mathcal{P}(\mathbf{x}^*)$ depicted in Fig. 3c. This demonstrates the importance of considering time-coupling and its effect on the total volume recovered.

B. A More Realistic Aggregator Model

We also tested a more realistic setting of an aggregator controlling 200 loads described by non-convex models subject to random externalities. This aggregator controls PVs, batteries, EVs, and TCLs, 50 of each type. Although the market is ran on an hourly basis indexed by $t = 1, \ldots, T$, loads are controlled at a finer timescale of 15-min intervals indexed by $\tau = 1, \ldots, 4T$. This is practical for devices such as TCLs and PVs. If ℓ_τ^d denotes the kW load consumed by device d during the 15-min interval τ, the kW load for this device over hour t is

$$p_t^d = \frac{1}{4}\sum_{k=1}^{4} \ell_{4(t-1)+k}^d.$$

We next describe the models loads ℓ_τ^d should satisfy for each type of loads. These models determine the device feasible sets $\mathcal{F}^d(\omega^d)$ in (6). To keep the notation uncluttered, we will drop superscript d.

Photovoltaics were modeled as negative loads as

$$-r_\tau \bar{p} \leq \ell_\tau \leq 0, \qquad \tau = 1:4T$$

where $\bar{p}$ is the PV capacity and r_τ the expected solar irradiance at interval τ. PV capacities were randomly sampled from real data from Southern California Edison (SCE) [27]. Solar irradiance profiles r_τ are the random externalities for PVs. To generate them, we used 1-min irradiance data from [28], perturbed them by ±10% with a normal distribution, and averaged them over 15-min intervals.

Batteries: We used a model with charging/discharging inefficiencies described below for $\tau = 1\ldots, 4T$:

$$\ell_\tau = \ell_\tau^+ - \ell_\tau^-, \quad 0 \leq \ell_\tau^+ \leq b_\tau \bar{p}, \quad 0 \leq \ell_\tau^- \leq (1 - b_\tau)\bar{p}$$

$$s_\tau = s_0 + \frac{1}{4}\sum_{k=1}^{\tau}\left(0.9\ell_k^+ - 1.1\ell_k^-\right), \quad 0 \leq s_\tau \leq \bar{s}$$

where $(\ell_\tau^+, \ell_\tau^-)$ are the (dis)-charging powers during interval τ, and binary variable b_τ indicates whether this battery is charging or discharging. The binary variable is needed to properly capture inefficiencies. Quantities $(\bar{p}, \bar{s})$ are the limits on power and SoC obtained also from SCE data [27]. The initial SoC s_0 is the random externality for batteries and was drawn independently and uniformly at random within $[0, \bar{s}]$. The $1/4$ term in front of the sum accounts for the 15-min intervals.

Electric vehicles were modeled similarly to batteries with the additional constraints that $(\ell_\tau, \ell_\tau^+, \ell_\tau^-)$ are zero for intervals τ the EV is not available. The assumption here is that EVs come and park at business locations. Upon arrival, each EV specifies a departure time and the needed SoC upon departure. Based on the most common EV models in the U.S market, we chose kW capacities from the set $\bar{\mathbf{p}} \in \{11, 16.5, 18, 19.2, 20, 21.1, 22\}$ kW, and kWh capacities from the set $\bar{s} \in \{42, 60, 70, 75, 85, 90, 100\}$ kWh. Arrival/departure times are uncertainties and were generated by sampling independently from a truncated Gaussian distribution between 9-10 AM and 4-5 PM, respectively. The requested SoC upon departure is also another random externality and was drawn uniformly between the SoC upon arrival and the SoC capacity of each vehicle.

Thermostatically controlled loads obeyed the discrete-time model assuming operation under the cooling mode [6]:

$$0 \leq \ell_\tau \leq b_\tau \bar{p}, \qquad \theta_s - 0.5 \leq \theta_\tau \leq \theta_s + 0.5$$

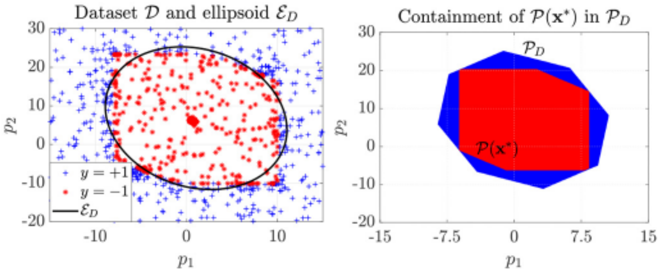


Fig. 4. *Left:* Sampled labeled points of the realistic aggregator and the trained convex quadratic classifier and resulting ellipsoid $\mathcal{E}_D$ for $T = 2$. *Right:* Inner polytopic approximation $\mathcal{P}_D$ of $\mathcal{E}_D$, and the final solution $\mathcal{P}(\mathbf{x}^*)$.

$$\theta_{\tau+1} = \theta_\tau - \frac{1}{4C}\sum_{l=1}^{\tau}\left(P\ell_l - \frac{\theta_l^a}{R}\right).$$

Here binary variable b_τ indicates whether the TCL is on, $\bar{p}$ is the fixed power consumed when the TCL is on, while θ_τ and θ_τ^a are respectively the house and predicted ambient temperatures in Celcius at time τ. Parameters (C, P, R) are the thermal capacitance, resistance, and power transfer of the house. They were drawn uniformly at random from $[1.5, 2.5]$, $[3, 5]$, and $[15, 30]$, respectively. The temperature setpoint θ_s was drawn uniformly at random from $[24, 26]$.

Data generation: To generate the training dataset $\mathcal{D}$, we solved the disaggregation problem in (6) for $T = 8$ control periods using the ℓ_1-norm in the objective to arrive at an MILP. We followed the steps explained in the Appendix with $K = 25$ scenarios of random externalities and $\kappa = 0.2$. The value of $K = 25$ may seem relatively small granted the dimension of $\boldsymbol{\omega}_k$ increases with the number of devices D. Nonetheless, several entries of $\boldsymbol{\omega}_k$ may be correlated due to patterns in device and solar profiles. Even for the uncorrelated entries of $\boldsymbol{\omega}_k$ (e.g., arrival/departure times and SoC for electric vehicles), one may not need to sample a large number of uncertainty scenarios K thanks to an *ergodicity* argument: Suppose we are interested in a particular uncertainty scenario $\boldsymbol{\omega}^d = \bar{\boldsymbol{\omega}}$ for some device d. Problem (6) is invariant on whether scenario $\bar{\boldsymbol{\omega}}$ appears for device d or some other device d'. As D increases, the chances of experiencing a scenario $\bar{\boldsymbol{\omega}}$ increase, and thus, there is no need for sampling many $\boldsymbol{\omega}^k$'s in (7). Note also that our method is independent of the data generation step, which in fact could be skipped if an aggregator has historical data. If not, the aggregator could handle (6)-(7) more efficiently than we did here, e.g., by parallelizing the K MILPs needed in (7), using specialized algorithms exploiting the unit commitment-type nature of (6), and moving computations on the cloud.

We first tested the case of $T = 2$ to visualize how the proposed OFD performs with realistic data. However, $T = 2$ is limiting for control of EVs and for this reason EVs were removed from this particular test. We tried solving the OFD for probability of infeasibility $\epsilon = 0.04$, which resulted in a dataset with $N = 500$ points (200 feasible and 300 infeasible ones). This dataset and the obtained ellipsoid $\mathcal{E}_D$ are shown in Figure 4 (left). The obtained classifier had a training accuracy of 94.68% and a validation accuracy of 93.76%. Figure 4 (right) shows the inner polytopic approximation $\mathcal{P}_D$ of $\mathcal{E}_D$, and the final solution $\mathcal{P}(\mathbf{x}^*)$, which demonstrates that the obtained $\mathcal{P}(\mathbf{x}^*)$ closely models the aggregate flexibility.

TABLE II
RESULTS ON OPTIMAL FLEXIBILITY DESIGN (OFD)

	Probability of infeasible aggregator schedule ϵ			
metric	0	0.04	0.08	0.12
% infeasible $\in \mathcal{P}(\bar{\mathbf{x}})$	13.64%	19.02%	21.72%	24.39%
$\hat{V}(\mathbf{x}_{\text{inner}})$	3.27	5.53	5.84	6.88
$\hat{V}(\mathbf{x}_{\text{outer}})$	7.23	11.86	12.52	21.09
$\hat{V}(\mathbf{x}_r)$	0.35	0.35	0.35	2.68
$\hat{V}(\mathbf{x}_m)$	53.9	-	-	-

We continued with the case of $T = 8$ and included all 4 device types. We tried solving the OFD for different values of $\epsilon \in \{0, 0.04, 0.08, 0.12\}$, which resulted in four datasets. Each dataset consisted of $N = 7,244$ points, equally split between feasible and infeasible ones.

We first wanted to study whether set $\mathcal{F}$ is convex by examining dataset $\mathcal{D}$ being a snapshot of $\mathcal{F}$. If $\mathcal{D}^-$ is the subset of $\mathcal{D}$ of all feasible points, we evaluated two metrics:

- M_1: % of infeasible points in $\mathcal{D}$ falling in conv($\mathcal{D}^-$).
- M_2: % of infeasible points falling in conv($\mathcal{D}^-$).

Metric M_1 can be computed easily by solving an LP for each of the infeasible points in $\mathcal{D}$. Nonetheless, this metric can be biased if $\mathcal{D}$ has been generated per the process of the Appendix. This is because several infeasible points in $\mathcal{D}$ have been intentionally selected to lie close to the boundary of $\mathcal{F}$, so they have higher chances of falling in conv($\mathcal{D}^-$). Metric M_2 alleviates this issue but is harder to compute. To approximate M_2, we sampled $I = 100$ points uniformly at random from conv($\mathcal{D}^-$) and checked what percentage of them were infeasible. However labeling each one of these points entails solving (6) for K times. Table I shows the two metrics for the four datasets. Metric M_2 is smaller than M_1 (in fact zero), as expected. The two metrics indicate that $\mathcal{F}$ is described quite accurately by a convex set, which resonates with the analytical findings in [9]. Nonetheless, electricity markets may want to surrogate $\mathcal{F}$ with polytope $\mathbf{Gp} \leq \mathbf{x}$ for a $\mathbf{G}$ that is convenient for market clearing processes. We next evaluated our OFD design under this requirement.

Following Alg. 1, we first trained the quadratic classifier by solving (11) using MOSEK and YAMIP in MATLAB. To avoid overfitting, dataset $\mathcal{D}$ was partitioned randomly into a training and a validation dataset in a proportion of 80% to 20%. Parameter λ was tuned to $\lambda = 10^{-6}$. Training took consistently less than 1 min. All tests were performed on an Intel Core i7 @ 3.4 GHz (16 GB RAM) computer using MATLAB on a single CPU without any parallelization. Table I reports the accuracy attained during training and validation by counting the percentage of points correctly labeled by the classifier. Such high accuracy suggests that $\mathcal{F}$ can be well approximated by the learned ellipsoid $\mathcal{E}_D$. It could also be argued that the two sets have roughly similar volumes. Table I reports the condition number of $\mathbf{W}_2$. Having a well-conditioned $\mathbf{W}_2$ is important in the transformations of Section V-A. The volume of $\mathcal{E}_D$ can be computed in closed form and is shown in Table II.

TABLE III
RANDOM $\bar{\mathbf{x}}$ FOR $\epsilon = 0.08$

	index				
Volume	1	2	3	4	5
$\hat{V}(\mathbf{x}_{\text{inner}})$	1.09	1.99	1.36	2.09	2.47

Proceeding with the algorithm, we approximated $\mathcal{E}_D$ by an inscribed polytope. In Step 2, setting $\delta = 0.1$ yielded 23 extra variables in $\mathbf{q}$. In Step 3, polytope $\mathcal{P}_T^{\delta}$ was converted to its $\mathcal{P}_D$ form using the Fourier-Motzkin algorithm as outlined in Section V-A. The obtained matrix $\mathbf{E}$ of (14) had roughly $4.2 \cdot 10^5$ rows. Steps 2–3 together took no more than 5 min per dataset.

For Step 4, we solved (21) to obtain the related $\bar{\mathbf{x}}$ per dataset, which took less than 1 min. For Step 5, we solved the LP in (19) to obtain the final polytope $\mathcal{P}(\mathbf{x})$. Given the large size of $\mathbf{E}$, solving this LP took roughly 4 hours per dataset. As there is no formula for the volume of a general polytope, the volume of $\mathcal{P}(\mathbf{x})$ was numerically estimated as detailed next. We first constructed a hyper-rectangle $\mathcal{H} = \{\mathbf{p} : \underline{\mathbf{p}} \leq \mathbf{p} \leq \bar{\mathbf{p}}\}$ containing $\mathcal{P}(\mathbf{x})$. The t-th entry of $\bar{\mathbf{p}}$ ($\underline{\mathbf{p}}$) can be computed by maximizing (minimizing) p_t subject to $\mathbf{p} \in \mathcal{P}(\mathbf{x})$. The volume of $\mathcal{H}$ is $\prod_{t=1}^{T}(\bar{p}_t - \underline{p}_t)$. We then sampled 10^6 points from $\mathcal{H}$ and found the ratio of these points falling in $\mathcal{P}(\mathbf{x})$. The estimated volume $\hat{V}(\mathcal{P}(\mathbf{x}))$ was computed by multiplying the ratio times the volume of $\mathcal{H}$. The results are presented in Table II. Note that $\mathbf{x}_{\text{inner}}$ and $\mathbf{x}_{\text{outer}}$ denote the flexibility parameters corresponding to $\mathcal{P}_{\mathcal{D}}$ being an inner and outer approximation of $\mathcal{E}_{\mathcal{D}}$, respectively.

We next studied the effect of ignoring time-coupling by using the reduced matrix $\mathbf{G}_r^\top = [\mathbf{I}^\top \, -\mathbf{I}^\top]$. We ran Alg. 1 again and obtained a polytope $\mathbf{G}_r\mathbf{p} \leq \mathbf{x}_r$ whose volume is reported in Table II. Oversimplifying the aggregator model apparently led to much reduced flexibility.

To study the effect of uncertainty, we conducted OFD with no chance probability ($\epsilon = 0$) assuming externalities in $\boldsymbol{\omega}$ are set to their mean values. Table II lists the volume of the computed polytope $\mathbf{G}\mathbf{p} \leq \mathbf{x}_m$, which is significantly larger than that of the polytope computed considering uncertainty. Nevertheless, such flexibility quantification may be misleading as it may not be realizable during operation.

To examine whether the heuristic of (21) for choosing $\bar{\mathbf{x}}$ is effective, we generated 5 random values for $\bar{\mathbf{x}}$ as

$$\bar{\mathbf{x}} = \left|\bar{\mathbf{x}}'\right| + \mathbf{G}\hat{\mathbf{p}}$$

where $|\cdot|$ is the element-wise absolute value; the entries of $\bar{\mathbf{x}}'$ are drawn independently from a standard normal distribution; and $\hat{\mathbf{p}}$ is the center of $\mathcal{P}_D$. Notice that irrespective of the value of $\bar{\mathbf{x}}'$, the computed $\bar{\mathbf{x}}$ always results in a non-empty polytope $\mathcal{P}(\bar{\mathbf{x}})$ as it contains at least $\hat{\mathbf{p}}$. We then ran our OFD algorithm for $\epsilon = 0.08$ for each of five randomly drawn values of $\bar{\mathbf{x}}$. Table III reports the resulted volumes. Recall that our choice of $\bar{\mathbf{x}}$ resulted in a volume of $\hat{V}(\mathbf{x}_{\text{inner}}) = 5.84$ for $\epsilon = 0.08$, which more than twice the volume obtained by the choices of Table III. Optimizing over $\bar{\mathbf{x}}$ is an interesting, yet challenging task that goes beyond the scope of this work.

VII. Conclusion

In this work, we have used disaggregation data to capture the time-coupled flexibility of aggregators in presence of random externalities and non-convexity in individual device models. The feasible set of an aggregator has been approximated by the ellipsoid described by a convex quadratic classifier trained over labeled (non)-disaggregatable schedules. As obtaining such labels can be computationally intensive as it involves solving multiple large-scale mixed-integer disaggregation programs, we have put forth an efficient data generation scheme. To arrive from the ellipsoid to the polytopic shape dictated by the market, the ellipsoid is first safely (inner) approximated by a general polytope. Thanks to Farkas' lemma and a volume argument, the general polytope is later approximated by a polytope having the form dictated by the market. Numerical tests corroborate the effectiveness of the novel OFD approach as aggregator's flexibility seems to be well surrogated by a convex set, more detailed polytopes can better inner approximate this feasible set as expected, while taking into account the time-coupled and uncertain nature of device models seems to be important. The proposed work forms the solid foundation for exploring open problems related to the participation of aggregators in electricity markets, such as computing flexibility sets endowed with probabilistic guarantees (confidence intervals) and designing optimal bidding strategies consisting of a feasibility set along with bidding costs.

Appendix

If points $\mathbf{p}_n$'s are sampled uniformly at random, it is much more likely to sample infeasible than feasible ones. Then to create a balanced dataset $\mathcal{D}$, one has to sample a large number of $\mathbf{p}_n$'s. Albeit sampling points is easy, labeling them can be time consuming. *Labeling a point* entails solving (6) for K samples of externalities $\boldsymbol{\omega}_k$'s to compute the chance statistic in (7). We next propose an efficient way for generating a balanced $\mathcal{D}$. First, estimate a hyper-rectangle $\mathcal{H}$ wherein all feasible schedules could lie. To obtain $\mathcal{H}$, one can compute the minimum/maximum values the schedule for each device $\mathbf{p}^d \in \mathcal{F}^d$ can take per control interval. Summing up these values across all devices yields bounds on the aggregator schedule $\mathbf{p}$. While sampling in $\mathcal{H}$ is more efficient than sampling in $\mathbb{R}^T$, obtaining a balanced $\mathcal{D}$ can still be challenging as the bounds can be loose. Moreover, infeasible points too far from the boundary of $\mathcal{F}$ may not be very informative for classification purposes. In light of these, our goals are: *i)* to sample infeasible points close to the boundary of the feasible set $\mathcal{F}$; and *ii)* for each infeasible point we sample, we also identify a feasible point. These goals can be achieved through the following steps:

D1) Sample a point $\mathbf{p}_1$ uniformly at random within hyper-rectangle $\mathcal{H}$. Most likely $\mathbf{p}_1$ is infeasible, so it is assigned label $y_1 = +1$ and pair $(\mathbf{p}_1, y_1)$ is appended to $\mathcal{D}$.

D2) In the process of labeling $\mathbf{p}_1$, we have also obtained points $\{\hat{\mathbf{p}}(\omega_k)\}_{k=1}^K$ per (8). Among these points, select the one that lies the furthest away from $\mathbf{p}_1$, that is the one with the largest $g(\hat{\mathbf{p}}(\omega_k); \omega_k)$. Call this point $\mathbf{p}_2$ and

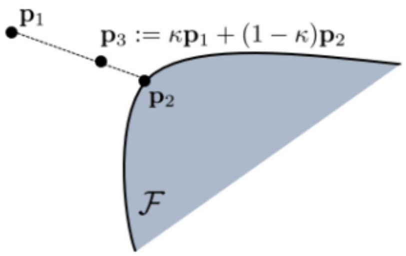

$\mathbf{p}_1$
$\mathbf{p}_3 := \kappa\mathbf{p}_1 + (1-\kappa)\mathbf{p}_2$
$\mathbf{p}_2$
$\mathcal{F}$

Fig. 5. To generate a nearly balanced dataset $\mathcal{D}$ in a computationally efficient manner, we draw a triplet of points $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ at a time. Point $\mathbf{p}_1$ is drawn uniformly at random and is most likely infeasible. Along its labeling process however, we find point $\mathbf{p}_2$ that is the approximate projection of $\mathbf{p}_1$ on feasible set $\mathcal{F}$. Point $\mathbf{p}_3$ is constructed as a convex combination of $(\mathbf{p}_1, \mathbf{p}_2)$ lying closer to $\mathbf{p}_2$. Albeit infeasible, point $\mathbf{p}_3$ lies close to the boundary of $\mathcal{F}$. Sampling such triplets yields a dataset having twice as many infeasible points as feasible ones. Upon taking convex combinations of existing $\mathbf{p}_2$'s, additional points lying in the interior of $\mathcal{F}$ can be constructed.

label it. If $\mathbf{p}_2$ is feasible (most likely it is), set $y_2 = -1$ and append pair $(\mathbf{p}_2, y_2)$ to $\mathcal{D}$.

D3) Construct point $\mathbf{p}_3 = \kappa\mathbf{p}_1 + (1-\kappa)\mathbf{p}_2$ for say $\kappa = 0.2$, and label it. Most likely, point $\mathbf{p}_3$ is infeasible but lies closer to the boundary of $\mathcal{F}$ than $\mathbf{p}_1$. This is because the feasible point $\mathbf{p}_2$ is expected to be close to the boundary of $\mathcal{F}$. Append pair $(\mathbf{p}_3, y_3)$ to $\mathcal{D}$.

D4) Repeat steps *D1)–D3)* until $\mathcal{D}$ reaches a desirable size.

Figure 5 depicts the triplet $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ presuming $\mathbf{p}_1$ lies far away from the boundary of $\mathcal{F}$; point $\mathbf{p}_2$ lies on or close to the boundary; and $\mathbf{p}_3$ is infeasible but close to the boundary of $\mathcal{F}$. When κ is selected closer to zero, point $\mathbf{p}_3$ gets closer to $\mathbf{p}_2$. This provides a more refined charting for the boundary of set $\mathcal{F}$ assuming labels are correct. Nevertheless, as labeling involves the stochastic approximation of (7), one may want to leave some additional space between $\mathbf{p}_2$ and $\mathbf{p}_3$, by selecting slightly larger values of κ such as $\kappa = 0.2$. We next cover the remaining special cases for completeness. If $\mathbf{p}_1$ is feasible, we can simply add $(\mathbf{p}_1, y_1)$ to $\mathcal{D}$ with $y_1 = -1$, and repeat *D1)*. If $\mathbf{p}_2$ is infeasible, we can replace it with the point $\hat{\mathbf{p}}(\omega_k)$ obtained during its labeling that lies the furthest away from $\mathbf{p}_2$. Although there is no analytical guarantee, replacing $\mathbf{p}_2$ with its projection seems to be reaching a feasible point within one or two iterations. Finally, if $\mathbf{p}_3$ is feasible, repeat *D3)* and use $\mathbf{p}_3$ instead of $\mathbf{p}_2$ in the convex combination to find the new $\mathbf{p}_3$.

The previous process is expected to generate double the number of infeasible over feasible points. Moreover, all feasible points lie on or close to the boundary of $\mathcal{F}$, which can be problematic when training a classifier. Both of these issues can be easily resolved by generating some feasible points that are strictly in the interior of $\mathcal{F}$. To this end, we also sample random points as the convex combination of already identified feasible points and label them using the standard process. While such convex combinations are not guaranteed to be feasible, our experiments in Section VI show that they are very likely to be feasible.

References

[1] "Electric Storage Participation in Markets Operated by Regional Transmission Organizations and Independent System Operators." Feb. 2018. [Online]. Available: https://www.ferc.gov/media/order-no-841 (Accessed: Jun. 27, 2022).

[2] S.-J. Kim and G. B. Giannakis, "Scalable and robust demand response with mixed-integer constraints," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2089–2099, Dec. 2013.

[3] M. Alizadeh, A. Scaglione, A. Goldsmith, and G. Kesidis, "Capturing aggregate flexibility in demand response," in *Proc. IEEE Conf. Decis. Control*, Los Angeles, CA, USA, Jul. 2014, pp. 6439–6445.

[4] A. Nayyar, J. Taylor, A. Subramanian, K. Poolla, and P. Varaiya, "Aggregate flexibility of a collection of loads," in *Proc. IEEE Conf. Decis. Control*, Florence, Italy, Mar. 2013, pp. 5600–5607.

[5] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Aggregate flexibility of thermostatically controlled loads," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 189–198, Jan. 2015.

[6] L. Zhao, W. Zhang, H. Hao, and K. Kalsi, "A geometric approach to aggregate flexibility modeling of thermostatically controlled loads," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4721–4731, Nov. 2017.

[7] M. S. Nazir, I. A. Hiskens, A. Bernstein, and E. Dall'Anese, "Inner approximation of Minkowski sums: A union-based approach and applications to aggregated energy resources," in *Proc. IEEE Conf. Decis. Control*, Miami, FL, USA, Dec. 2018, pp. 1–10.

[8] F. L. Müller, J. Szabó, O. Sundström, and J. Lygeros, "Aggregation and disaggregation of energetic flexibility from distributed energy resources," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1205–1214, Mar. 2019.

[9] K. Hreinsson, A. Scaglione, M. Alizadeh, and Y. Chen, "New insights from the Shapley-Folkman lemma on dispatchable demand in energy markets," *IEEE Trans. Power Syst.*, vol. 36, no. 5, pp. 4028–4041, Sep. 2021.

[10] E. Polymeneas and S. Meliopoulos, "Aggregate modeling of distribution systems for multi-period OPF," in *Proc. Power Syst. Comput. Conf.*, Genoa, Italy, Jun. 2016, pp. 1–8.

[11] J. Silva *et al.*, "Estimating the active and reactive power flexibility area at the TSO-DSO interface," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4741–4750, Sep. 2018.

[12] X. Chen and N. Li, "Leveraging two-stage adaptive robust optimization for power flexibility aggregation," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 3954–3965, Sep. 2021.

[13] X. Chen, E. Dall'Anese, C. Zhao, and N. Li, "Aggregate power flexibility in unbalanced distribution systems," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 258–269, Jan. 2020.

[14] B. Cui, A. Zamzam, and A. Bernstein, "Network-cognizant time-coupled aggregate flexibility of distribution systems under uncertainties," *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1723–1728, Nov. 2021.

[15] Y. Zhang, S. Shen, and J. L. Mathieu, "Distributionally robust chance-constrained optimal power flow with uncertain renewables and uncertain reserves provided by loads," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1378–1388, Mar. 2017.

[16] J. Mathieu, M. G. Vaya, and G. Andersson, "Uncertainty in the flexibility of aggregations of demand response resources," in *Proc. Ind. Electron. Conf. (IECON)*, Nov. 2013, pp. 8052–8057.

[17] S. Taheri, V. Kekatos, and H. Veeramachaneni, "Strategic investment in energy markets: A multiparametric programming approach," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2590–2600, Jul. 2022.

[18] T. Chen, N. Li, and G. B. Giannakis, "Aggregating flexibility of heterogeneous energy resources in distribution networks," in *Proc. Amer. Control Conf.*, Milwaukee, WI, USA, Aug. 2018, pp. 4604–4609.

[19] P. Li, B. Jin, R. Xiong, D. Wang, A. Sangiovanni-Vincentelli, and B. Zhang, "A tractable ellipsoidal approximation for voltage regulation problems," in *Proc. Amer. Control Conf.*, Philadelphia, PA, USA, Aug. 2019, pp. 1301–1306.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[21] A. Ben-Tal and A. Nemirovski, "On polyhedral approximations of the second-order cone," *Math. Oper. Res.*, vol. 26, no. 2, pp. 193–205, May 2001.

[22] F. Glineur, "Computational experiments with a linear approximation of second-order cone optimization," Faculte Polytechnique de Mons, Mons, Belgium, Rep., Nov. 2000.

[23] S.-J. Kim, N. Y. Soltani, and G. B. Giannakis, "Resource allocation for OFDMA cognitive radios under channel uncertainty," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 3188–3191.

[24] A. Schrijver, *Theory of Linear and Integer Programming*. New York, NY, USA: Wiley, 1986.

[25] B. Eaves and R. Freund, "Optimal scaling of balls and polyhedra," *Math. Program.*, vol. 23, no. 1, pp. 138–147, 1982.

[26] O. Mangasarian, "Set containment characterization," *J. Global Optim.*, vol. 24, no. 1, pp. 473–480, 2002.

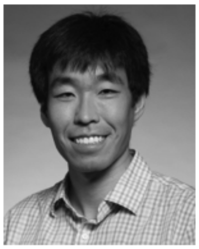
[27] A. Bernstein and E. Dall'Anese, "Real-time feedback-based optimization of distribution grids: A unified approach," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 3, pp. 1197–1209, Sep. 2019.

[28] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," in *Proc. Workshop Data Min. Appl. Sustain.*, 2012, pp. 1–6.

Sina Taheri (Member, IEEE) received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2016, and the M.Sc. and Ph.D. degrees in electrical engineering from Virginia Tech, Blacksburg, VA, USA, in 2019 and 2021, respectively. He is currently with Ibotta Inc. His research interests are focused on the application of optimization and machine learning to develop algorithmic solutions for monitoring and operation of smart power systems.

Vassilis Kekatos (Senior Member, IEEE) received the Diploma, M.Sc., and Ph.D. degrees in computer science and engineering from the University of Patras, Greece, in 2001, 2003, and 2007, respectively. He is an Associate Professor with the Bradley Department of ECE, Virginia Tech. In 2014, he stayed with the University of Texas at Austin and The Ohio State University as a Visiting Researcher. He has been a Research Associate with the ECE Department, University of Minnesota, where he received the Postdoctoral Career Development Award (honorable mention). His research focus is on optimization and learning for future energy systems. He is a recipient of the NSF Career Award in 2018 and the Marie Curie Fellowship. He is currently serving on the editorial board of the IEEE TRANSACTIONS ON SMART GRID.

Sriharsha Veeramachaneni received the Ph.D. degree in computer engineering from Rensselaer Polytechnic Institute. He is a Senior Fellow with the Applied Mathematics Group, NextEra Analytics Inc., a subsidiary of NextEra Energy Inc., specializing in statistics, machine learning, and optimization.



Baosen Zhang (Member, IEEE) received the Bachelor of Applied Science degree in engineering science from the University of Toronto in 2008, and the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley in 2013. He was a Postdoctoral Scholar with Stanford University. He is currently the Keith and Nancy Endowed Career Development Professor of Electrical and Computer Engineering with the University of Washington, Seattle, WA, USA. His research interests are in control, optimization, and learning applied to power systems and other cyber–physical systems. He received the NSF CAREER Award as well as several best paper awards.