Alleviating the Curse of Dimensionality in Minkowski Sum Approximations of Storage Flexibility

Emrah Öztürk ⬤, Timm Faulwasser ⬤, *Senior Member, IEEE*, Karl Worthmann ⬤, Markus Preißinger ⬤, and Klaus Rheinberger ⬤

Abstract—**Many real-world applications require the joint optimization of a large number of flexible devices over some time horizon. The flexibility of multiple batteries, thermostatically controlled loads, or electric vehicles, e.g., can be used to support grid operations and to reduce operation costs. Using piecewise constant power values, the flexibility of each device over d time periods can be described as a polytopic subset in power space. The aggregated flexibility is given by the Minkowski sum of these polytopes. As the computation of Minkowski sums is in general demanding, several approximations have been proposed in the literature. Yet, their application potential is often objective-dependent and limited by the curse of dimensionality. In this paper, we show that up to 2^d vertices of each polytope can be computed efficiently and that the convex hull of their sums provides a computationally efficient inner approximation of the Minkowski sum. Via an extensive simulation study, we illustrate that our approach outperforms ten state-of-the-art inner approximations in terms of computational complexity and accuracy for different objectives. Moreover, we propose an efficient disaggregation method applicable to any vertex-based approximation. The proposed methods provide an efficient means to aggregate and to disaggregate typical battery storages in quarter-hourly periods over an entire day with reasonable accuracy for aggregated cost and for peak power optimization.**

Index Terms—**distributed energy resources, battery storage, flexibility aggregation, Minkowski sum, vertex-based approximation, ancillary services, demand response**

I. Introduction

THE coordinated control of a large number of distributed, flexible devices offers significant potential for power grids. For example, the flexibility of shiftable loads in the distribution grid, such as batteries, refrigerators, heat pumps,

water heaters, and air conditioners, can be used to support grid operations and to reduce operation costs. Eventually, for the sake of computational tractability, the large number of devices necessitates to cluster units and their flexibilities. To this end, the concept of an aggregator is introduced in the literature, cf. [1]. The aggregator is typically an entity located between consumers, energy markets, and network operators. This entity manages contracted consumer devices, estimates the collective flexibility, and assigns power profiles to individual devices. The aggregator thus serves as an interface to a virtual power plant, see also [2]. The flexibility of each device can be described by a subset in the power space and the aggregated flexibility by the point-wise sum of these sets. However, the computation of this Minkowski sum is often prohibitive, cf. [3]. Therefore, various tailored approximations have been proposed in the literature.

Existing approximations can be roughly divided into top-down and bottom-up approaches. The former typically use machine learning, Markov chains, etc. to directly approximate the aggregated flexibility, cf. [4]–[6]. The latter start from individual flexibilities, they usually assume a certain underlying structure, and they can be further divided into inner and outer approximations. Outer approximations [7]–[12], compute supersets of the Minkowski sum and therefore they have the major drawback to potentially contain infeasible elements. Inner approximations make up the majority of Minkowski sum approximations in the literature [7], [9]–[11], [13]–[18], [23]–[25]. However, many of these have drawbacks, such as poorer optimization results compared to a setting without flexibility, high computational burden, and objective-dependent performance, cf. [19]. Indeed the computational burden limits the application potential of several approaches significantly. The objective-dependent performance is likely induced by the employed underlying set parametrizations, e.g., an ellipsoid inscribed in a polytope covers the interior rather than the vertices, resulting in poor performance in cost optimization and in good performance for peak reduction. An attempt to avoid the underlying structure is made in [20], where a recursive algorithm is proposed to compute the vertices of a polytope by computing extreme bounds. Yet, this approach suffers from combinatorial complexity as it attempts to compute all vertices with a scheme that may lead to redundant computations. However, a related idea will also be used for the method proposed in the present paper. Further aggregation strategies, such as characterizing the flexibility of

a fleet of heterogeneous storage units using the so-called E-p transform, can be found in [26]–[28]; strategies in the presence of nonlinearities with probabilistic inputs are discussed by [29]–[31]. There also exist a dynamic programming approach [32] and an exact aggregation strategy for a population of electric vehicles using permutahedra [33].

Disaggregation represents the inverse operation to aggregation, i.e., the distribution of power profiles across individual flexible devices, cf. [7], [14], [23]. Existing methods are often based on the solution of optimization problems which may induce a significant computational burden.

The novelty of the present paper is threefold. First, we propose an efficient vertex-based inner approximation that can be computed for a large number of time periods within seconds and that overcomes the weaknesses of existing approximations. Second, the proposed approximation method is benchmarked against ten state-of-the-art inner approximations from the literature. It is shown to outperform the other methods in terms of accuracy for various objectives and in terms of computational performance. Finally, we propose an efficient disaggregation method that does not require optimization and that can be combined with any vertex-based approximation.

The remainder of this paper is organized as follows: Definitions are given in Section II, where we define our approach for all polytopes satisfying two assumptions, discuss its properties, and give example polytopes for illustration purposes. Section III discusses the general results related to our approach. In Section IV we propose an efficient algorithm to compute the approximation for battery storages with unrestricted final energy and extend it to the case of restricted final energy by applying corrections. In Section V, we test our approximation against 10 state-of-the-art inner approximations in terms of accuracy for various objectives and computational complexity. Section VI is devoted to a novel disaggregation method that applies to all vertex-based approximations. Finally, conclusions are drawn in Section VII.

Notation: The set of natural and real numbers are denoted by $\mathbb{N} = \{1, 2, \ldots\}$ and $\mathbb{R}$, respectively. The Minkowski sum of sets $\mathcal{X}_i \subseteq \mathbb{R}^d$, $i \in \{1, \ldots, n\}$ is defined by $\mathcal{M} := \{x \in \mathbb{R}^d : x = \sum_{i=1}^n x_i, \ x_i \in \mathcal{X}_i\}$. For a matrix $A \in \mathbb{R}^{k \times d}$ and a vector $b \in \mathbb{R}^k$, the set $\mathcal{P}(A, b) := \{x \in \mathbb{R}^d : Ax \le b\}$ is a polyhedron, and a polytope if it is bounded. The convex hull of a set $\mathcal{X}$ is written as $\text{Conv}(\mathcal{X})$. The d-dimensional vector of zeros and ones are written as $\mathbf{0}_d$ and $\mathbf{1}_d$, respectively. For $x \in \mathbb{R}^d$ and $t \le d$, we use the notation $\text{Proj}^t(x) := (x_1, \ldots, x_t, \mathbf{0}_{d-t})^\top$ for the projection of x onto its first t components. The vector consisting of the first t components of a vector $x \in \mathbb{R}^d$ is denoted by $x_{[t]} \in \mathbb{R}^t$. A matrix with constant diagonals descending from left to right is called a Toeplitz matrix. We say $v \in \mathbb{R}^d$ is a proper convex combination of $p, q \in \mathbb{R}^d$ if $v = tp + (1-t)q$, with $p \ne q$ and $t \in (0, 1)$. The cardinality of a set $\mathcal{X}$ is denoted by $|\mathcal{X}|$.

II. PRELIMINARIES

In this section, we introduce necessary assumptions, give example polytopes that satisfy these assumptions, and we
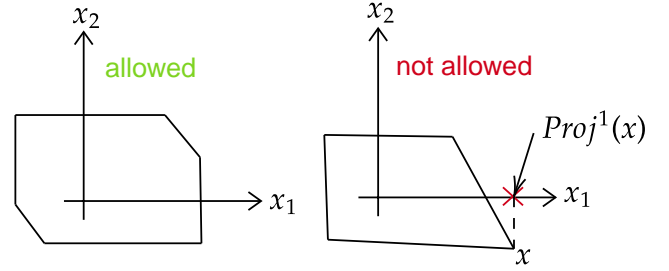

Fig. 1. Illustration of Assumption 2. While the polytope on the left satisfies the assumption, the polytope on the right does not.

define vectors of extreme actions within these polytopes. We consider the following assumptions for $\mathcal{P}(A, b) \subset \mathbb{R}^d$:

Assumption 1 (Required flexibility). *If* $\text{Proj}^t(x) \in \mathcal{P}(A, b)$ *for* $t \in \{1, \ldots, d-1\}$, *then there exists an* $\varepsilon \in \mathbb{R} \setminus \{0\}$ *such that* $(x_1, \ldots, x_t, \varepsilon, \mathbf{0}_{d-(t+1)})^\top \in \mathcal{P}(A, b)$. *Furthermore, there exists an* $\varepsilon \in \mathbb{R} \setminus \{0\}$ *such that* $(\varepsilon, \mathbf{0}_{d-1})^\top \in \mathcal{P}(A, b)$.

Assumption 2 (Projection feasibility). *If* $x \in \mathcal{P}(A, b)$, *then* $\text{Proj}^t(x) \in \mathcal{P}(A, b)$ *for all* $t \in \{1, \ldots, d-1\}$. *Furthermore,* $\mathbf{0}_d \in \mathcal{P}(A, b)$.

Assumption 1 requires a minimum flexibility in each time period, and Assumption 2 requires the feasibility of all projections of x if x is feasible, cf. Fig. 1. The inclusion of the zero vector is the extension of the projections beyond $t = 1$. It models not using the flexibility.

In the following, we characterize polytopes that are typically used to model the flexibility of battery storages. These polytopes are parameterized by the vector $p = (\alpha, \underline{x}, \overline{x}, \overline{S}, \Delta t)^\top \in (0, 1] \times (-\infty, 0] \times [0, \infty)^2 \times (0, \infty)$ which denotes, respectively, the self-discharge factor, lower and upper bound on the charging rate (kW), maximum State of Charge (SoC; kWh), and time step (h). Furthermore, the number of time periods is denoted by $d \in \mathbb{N}$, initial SoC by $S_0 \in [0, \overline{S}]$ (kWh), and minimum final SoC by $S_f \in [0, \overline{S}]$ (kWh). The set of feasible power profiles given by the system dynamics

$$\underline{x} \le x(t) \le \overline{x} \quad \forall\, t = 1, \ldots, d \tag{1a}$$
$$S(t) = \alpha S(t-1) + x(t)\Delta t \quad \forall\, t = 1, \ldots, d \tag{1b}$$
$$0 \le S(t) \le \overline{S} \quad \forall\, t = 1, \ldots, d-1 \tag{1c}$$
$$S(0) = S_0 \tag{1d}$$
$$S_f \le S(d) \le \overline{S} \tag{1e}$$

results in the polytope

$$\mathcal{B}(S_0, S_f, p) := \{x \in \mathbb{R}^d : A(\alpha)x \le b(S_0, S_f, p)\} \tag{2}$$

with $A(\alpha) \in \mathbb{R}^{4d \times d}$ and $b(S_0, S_f, p) \in \mathbb{R}^{4d}$ defined by

$$A(\alpha) := \left(-I, I, \Gamma^\top, -\Gamma^\top\right)^\top \tag{3a}$$
$$b(S_0, S_f, p) :=$$
$$\left(-\underline{x}\mathbf{1}_d^\top, \overline{x}\mathbf{1}_d^\top, \frac{1}{\Delta t}(\overline{S}\mathbf{1}_d - S_0 a_d)^\top, \frac{S_0}{\Delta t}a_{d-1}^\top, \frac{\alpha^d S_0 - S_f}{\Delta t}\right)^\top . \tag{3b}$$

Moreover, we have $a_d := (\alpha, \alpha^2, \ldots, \alpha^d)^\top$, $I \in \mathbb{R}^{d \times d}$ is the identity matrix, and $\Gamma \in \mathbb{R}^{d \times d}$ is a Toeplitz matrix with first column and row defined by $(1, \alpha, \ldots, \alpha^{d-1})^\top$ and $(1, 0, \ldots, 0)$, respectively. Note that we use x as flexibility variable as in [7], [8], [13], [16], [19], [20] rather than the notation with u which is commonly used in systems and control. The polytopes $\mathcal{B}(S_0, S_{\mathrm{f}}, p)$ model a variety of real-world flexibilities such as batteries, thermostatically controlled loads, cf. [8], [9]. Moreover, if $\overline{x} > 0, \underline{x} < 0$, and $\overline{S} > 0$, then $\mathcal{B}(S_0, 0, p)$ satisfies Assumptions 1 and 2.

Our approach aims to compute certain vectors of extreme actions within the polytopes.

Definition 1 (Extreme actions). *Let polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d$, $i \in \{1, \ldots, n\}$ satisfy the Assumptions 1 and 2. Then, for $j \in \{-1, 1\}^d$ the vectors $y_i^j \in \mathbb{R}^d$ defined by*

$$y_{i,1}^j := j_1 \cdot \max\{j_1 \cdot x \in \mathbb{R} : (x, \mathbf{0}_{d-1})^\top \in \mathcal{P}(A_i, b_i)\}, \quad (4)$$

and

$$y_{i,t}^j := j_t \cdot \max\{j_t \cdot x \in \mathbb{R} : (y_{i,[t-1]}^j, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)\}. \quad (5)$$

for $t \in \{2, \ldots, d\}$ are called extreme actions.

Note that $j_t = -1$ in (4) and (5) is equivalent to replacing the maximization with a minimization. Intuitively, the vectors y_i^j are obtained by moving as far as possible in each axis in the negative direction if $j_t = -1$, and in the positive direction if $j_t = 1$, cf. Fig. 2. The vectors y_i^j exist for all polytopes fulfilling the Assumptions 1 and 2, and it holds that $y_i^j \in \mathcal{P}(A_i, b_i)$ by construction. The summation over all $i = 1, \ldots, n$ with fixed $j \in \{-1, 1\}^d$ is denoted by:

$$v^j := \sum_{i=1}^n y_i^j, \quad (6)$$

and the convex hull of the set of summed vectors leads to

$$\mathcal{A} := \mathrm{Conv}(\{v^j : j \in \{-1, 1\}^d\}). \quad (7)$$

The set $\mathcal{A}$ can be described as a deformed cuboid, cf. Fig. 2. It follows from (7) that $\mathcal{A}$ is a polytope and $\mathcal{A} \subseteq \mathcal{M}$.

Henceforth, we show that the summed extreme actions (6) are distinct vertices of the Minkowski sum, and thus the convex hull of summed vectors turns out to be an inner approximation of the Minkowski sum.

III. Main Result

Next, we discuss the properties of the summed extreme actions. Due to space limitations, standard definitions such as convex independence, vertex are not given; instead we refer to, e.g., [21], [22]. Fig. 2 illustrates the following results.

Lemma 1. *Let polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d$, $i \in \{1, \ldots, n\}$, fulfill the Assumptions 1 and 2. Further, let $v^j, v^k \in \mathbb{R}^d$, $j, k \in \{-1, 1\}^d$ satisfy (6). Then, the following hold:*

1) $v_t^j \begin{cases} \geq 0 & \text{for } j_t = 1 \\ \leq 0 & \text{for } j_t = -1 \end{cases} \quad \forall t \in \{1, \ldots, d\},$

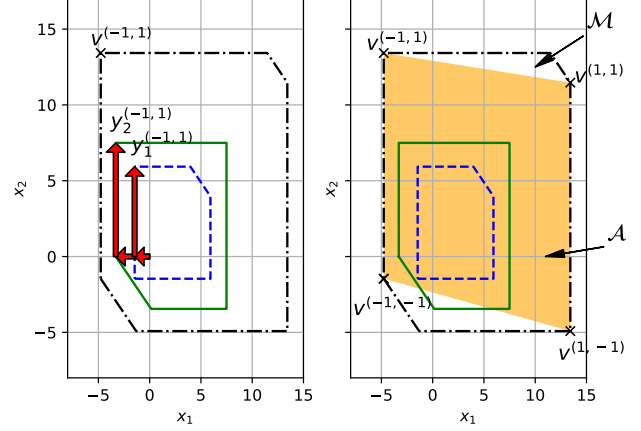2) *if $j \neq k$, then $v^j \neq v^k$.*

The proof is given in Appendix-A.

Fig. 2. Left: vectors $y_1^{(-1,1)}, y_2^{(-1,1)}$ within the polytopes shown in dashed blue and solid green, and the sum $v^{(-1,1)}$ shown in the Minkowski sum $\mathcal{M}$ in dash-dotted black. Right: all possible vectors $v^j, j \in \{-1, 1\}^2$ in the Minkowski sum with the resulting set $\mathcal{A}$ in orange.

Lemma 2. *Let polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d$, $i \in \{1, \ldots, n\}$ with Minkowski sum $\mathcal{M}$ fulfill the Assumptions 1 and 2, and $p \in \mathbb{R}^d$. Further, let $v^j \in \mathbb{R}^d, j \in \{-1, 1\}^d$ satisfy (6). For $t \in \{2, \ldots, d\}$, if $p_{[t-1]} = v_{[t-1]}^j$ and $p_t > v_t^j$ with $j_t = 1$, or $p_t < v_t^j$ with $j_t = -1$, then $p \notin \mathcal{M}$. Furthermore, if $p_1 > v_1^j$ with $j_1 = 1$ or $p_1 < v_1^j$ with $j_1 = -1$, then $p \notin \mathcal{M}$.*

The proof is given in Appendix-B. Lemma 2 states that there can be no vector in $\mathcal{M}$ that has $t - 1$ coordinates equal to v^j and a value greater than v_t^j in the t-th coordinate if $j_t = 1$. Similarly, there cannot be a vector with equal $t - 1$ coordinates in $\mathcal{M}$ that has a value less than v_t^j when $j_t = -1$. This characteristic behavior is also illustrated in Fig. 2.

Proposition 1. *Let polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d$, $i \in \{1, \ldots, n\}$, fulfill the Assumptions 1 and 2. Further, let $v^j \in \mathbb{R}^d, j \in \{-1, 1\}^d$ satisfy (6), and $\mathcal{A}$ satisfy (7). Then, v^j is a vertex of $\mathcal{A}$.*

The proof is in Appendix-C. The proposition states that v^j is a vertex of $\mathcal{A}$, and by Lemma 1, the elements of $\{v^j : j \in \{-1, 1\}^d\}$ are distinct. Thus, they are distinct vertices of $\mathcal{A}$.

Proposition 2. *Let polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d$, $i \in \{1, \ldots, n\}$, with Minkowski sum $\mathcal{M}$ fulfill the Assumptions 1 and 2. Further, let $v^j \in \mathbb{R}^d, j \in \{-1, 1\}^d$ satisfy (6), $\mathcal{A}$ satisfy (7), and $p, q \in \mathcal{M}$ with $v^j = tp + (1 - t)q, t \in (0, 1)$, then, $p, q \in \mathcal{A}$.*

The proposition gives that if v^j is a proper convex combination of elements $p, q \in \mathcal{M}$, then p, q must be in $\mathcal{A}$, see Appendix-D for the proof. We can now state our first main result which shows that the readily computable 2^d vectors v^j are indeed vertices of the Minkowski sum, and, thus, their convex hull constitutes an inner approximation.

Theorem 1 (Extreme actions define vertices). *Let polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d$, $i \in \{1, \ldots, n\}$, with Minkowski sum $\mathcal{M}$ fulfill the Assumptions 1 and 2. Then, any $v^j \in \mathbb{R}^d, j \in \{-1, 1\}^d$,*

satisfying (6) *is a vertex of* $\mathcal{M}$.

Proof. Suppose that v^j is not a vertex of $\mathcal{M}$, then $v^j = tp + (1-t)q$ with $p, q \in \mathcal{M}$, $p \neq q$ and $t \in (0, 1)$. From Proposition 2 it follows that $p, q \in \mathcal{A}$, which gives v^j as a proper convex combination of elements in $\mathcal{A}$. Thus v^j cannot be a vertex of $\mathcal{A}$, which contradicts Proposition 1. Hence, the assumption that v^j is not a vertex of $\mathcal{M}$ must be false. $\square$

Theorem 1 combined with Lemma 1 states that the sums of extreme actions are distinct vertices of $\mathcal{M}$, providing a novel method to compute a subset of Minkowski sum vertices. Note that $\mathcal{A}$ is exact for cuboids since they have 2^d vertices in d-dimensional space and $|\{v^j : j \in \{-1, 1\}^d\}| = 2^d$. Moreover, for any set $\mathcal{V} \subseteq \{v^j : j \in \{-1, 1\}^d\}$ holds $\mathrm{Conv}(\mathcal{V}) \subseteq \mathcal{M}$, thus $\mathrm{Conv}(\mathcal{V})$ is an inner approximation of $\mathcal{M}$.

IV. Application to Battery Storage

Next, we present an efficient algorithm for computing the $y^j, j \in \{-1, 1\}^d$ within the set $\mathcal{B}(S_0, 0, p)$. This approach is then further extended by a corrective algorithm to compute a subset of vertices of $\mathcal{B}(S_0, S_f, p)$. Note that this is necessary because $\mathcal{B}(S_0, S_f, p)$ may violate Assumption 2 for arbitrary $S_f > 0$. Finally, the complete algorithm for polytopes $\mathcal{B}(S_{0,i}, S_{f,i}, p_i), i = 1, \ldots, n$ is presented.

Algorithm 1 computes the y^j for given $S_0, p, j \in \{-1, 1\}^d$, and $S_f = 0$ without invoking any numerical optimization problems. The procedure iterates through the components j_t. If $j_t = 1$, then y_t^j is determined by charging to the limit without violating the constraints, and by discharging to the limit for $j_t = -1$. To this end, Line 5 checks whether the upper energy constraint for $y_t^j = \overline{x}$ is violated. If so, y_t^j in Line 6 is chosen to fully charge the battery. Similarly, Line 10 checks whether the lower energy constraint is violated with $y_t^j = \underline{x}$. If so, y_t^j in Line 11 is chosen to fully discharge the battery. The application of Algorithm 1 to all $j \in \{-1, 1\}^d$ yields the set of vectors $\{y^j : j \in \{-1, 1\}^d\}$. The aggregated vectors are then obtained by storing these vectors in matrices V_i and further calculating $\sum_{i=1}^{n} V_i$.

Algorithm 1 (Vertex)

Input $S_0, p, j \in \{-1, 1\}^d$

1: $y^j \leftarrow \mathbf{0}_d$
2: **for** $t = 1$ to d **do**
3:     **if** $j_t = 1$ **then**
4:         $y_t^j \leftarrow \overline{x}$
5:         **if** $\alpha^t S_0 + \sum_{\tau=1}^{t} \alpha^{t-\tau} y_\tau^j \Delta t > \overline{S}$ **then**
6:             $y_t^j \leftarrow \frac{\overline{S} - (\alpha^t S_0 + \sum_{\tau=1}^{t-1} \alpha^{t-\tau} y_\tau^j \Delta t)}{\Delta t}$
7:         **end if**
8:     **else if** $j_t = -1$ **then**
9:         $y_t^j \leftarrow \underline{x}$
10:         **if** $\alpha^t S_0 + \sum_{\tau=1}^{t} \alpha^{t-\tau} y_\tau^j \Delta t < 0$ **then**
11:             $y_t^j \leftarrow \frac{-(\alpha^t S_0 + \sum_{\tau=1}^{t-1} \alpha^{t-\tau} y_\tau^j \Delta t)}{\Delta t}$
12:         **end if**
13:     **end if**
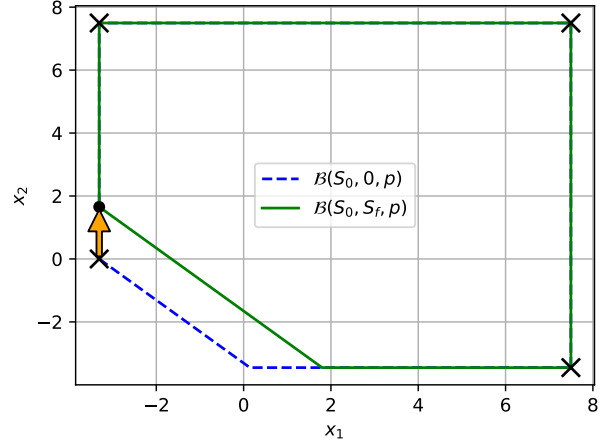14: **end for**

Output y^j

Fig. 3. The set $\mathcal{B}(S_0, S_f, p)$ in solid green and the set $\mathcal{B}(S_0, 0, p)$ in dashed blue. The crosses on $\mathcal{B}(S_0, 0, p)$ indicate the y^j, and the arrow with dot visualizes the correction process.

The previous approach can be extended to compute a subset of vertices of $\mathcal{B}(S_0, S_f, p)$, which may violate Assumption 2 for arbitrary $S_f > 0$, cf. Fig. 3. Suppose y^j satisfying Definition 1 is obtained for the set $\mathcal{B}(S_0, 0, p)$. The set $\{x \in \mathbb{R} : (y_{[d-1]}^j, x)^\top \in \mathcal{B}(S_0, S_f, p)\}$ is equivalent to $\{x \in \mathbb{R} : \underline{x} \leq x \leq \overline{x}, S_f \leq \alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} y_\tau^j \Delta t + x \Delta t \leq \overline{S}\}$. Hence, if $y^j \notin \mathcal{B}(S_0, S_f, p)$, then $S_f > \alpha^d S_0 + \sum_{\tau=1}^{d} \alpha^{d-\tau} y_\tau^j \Delta t$ since $y^j \in \mathcal{B}(S_0, 0, p)$ and the remaining inequalities are identical for both sets. Thus, increasing the values in y^j without violating the power constraints so that the inequality associated with S_f is satisfied yields $y^j \in \mathcal{B}(S_0, S_f, p)$. The set $\mathcal{B}(S_0, S_f, p)$ models a battery with a minimum final energy constraint. The charging and discharging in $\mathcal{B}(S_0, 0, p)$ may result in a final energy less than S_f. Thus, by correcting—i.e., increasing the values in y^j—one can achieve the given final energy S_f, and the corrected vectors $\tilde{y}^j$ fulfill $\tilde{y}^j \in \mathcal{B}(S_0, S_f, p)$, cf. Lemma 3. The correction process starts at the last period d by checking in Line 2 of Algorithm 2 whether S_f can be reached without violating the power constraints. If this is possible, the d-th coordinate of y^j is changed in Line 3 and the algorithm terminates, otherwise, the coordinate $d-1$ is changed to the highest possible value $\overline{x}$ and it is checked again whether the final SoC can be reached without violating the power constraints. If possible, the d-th coordinate of y^j is changed in Line 9 and the algorithm terminates, otherwise, it is continued with the coordinate $d-2$ and so forth until S_f is reached. This correction process is visualized in Fig. 3. First, the y^j are computed within $\mathcal{B}(S_0, 0, p)$, i.e., the four crosses in Fig. 3, then the coordinates of the crosses not contained in $\mathcal{B}(S_0, S_f, p)$ are increased, so that S_f is reached, indicated by the arrow and dot in Fig. 3.

Lemma 3. *Let* $\mathcal{B}(S_0, S_f, p), \mathcal{B}(S_0, 0, p) \subset \mathbb{R}^d$ *with parameter vector* $p = (\alpha, \underline{x}, \overline{x}, \overline{S}, \Delta t)^\top$ *be given. Further, let* $y^j \in \mathbb{R}^d$, $j \in \{-1, 1\}^d$ *satisfy Definition 1 for* $\mathcal{B}(S_0, 0, p)$. *Then,* $\tilde{y}^j$ *defined by Algorithm 2 is in* $\mathcal{B}(S_0, S_f, p)$ *if* $\mathcal{B}(S_0, S_f, p) \neq \emptyset$.

Algorithm 2 (Correction)

Input $S_0, S_f, p, y^j \in \mathcal{B}(S_0, 0, p)$

1: **if** $y^j \notin \mathcal{B}(S_0, S_f, p)$ **then**
2:      **if** $\underline{x} \leq \frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} y_\tau^j \Delta t)}{\Delta t} \leq \overline{x}$ **then**
3:          $y_d^j \leftarrow \frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} y_\tau^j \Delta t)}{\Delta t}$
4:      **end if**
5:      $t \leftarrow d - 1$
6:      **while** $S_f \neq \alpha^d S_0 + \sum_{\tau=1}^{d} \alpha^{d-\tau} y_\tau^j \Delta t$ and $t > 0$ **do**
7:          $y_t^j \leftarrow \overline{x}$
8:          **if** $\underline{x} \leq \frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} y_\tau^j \Delta t)}{\Delta t} \leq \overline{x}$ **then**
9:              $y_d^j \leftarrow \frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} y_\tau^j \Delta t)}{\Delta t}$
10:          **end if**
11:          $t \leftarrow t - 1$
12:      **end while**
13: **end if**
14: $\tilde{y}^j \leftarrow y^j$

Output $\tilde{y}^j$

For the proof see Appendix-E. The next result shows that the corrected vector $\tilde{y}^j$ is a vertex of $\mathcal{B}(S_0, S_f, p)$; its proof is given in Appendix-F.

Theorem 2. *Let nonempty sets $\mathcal{B}(S_0, S_f, p), \mathcal{B}(S_0, 0, p) \subset \mathbb{R}^d$ with parameter vector $p = (\alpha, \underline{x}, \overline{x}, \overline{S}, \Delta t)^\top$ be given. Further, let $y^j \in \mathbb{R}^d$, $j \in \{-1, 1\}^d$ satisfy Definition 1 for $\mathcal{B}(S_0, 0, p)$. Then, $\tilde{y}^j$ defined by Algorithm 2 is a vertex of $\mathcal{B}(S_0, S_f, p)$.*

Algorithm 3 now uses the polytopes $\mathcal{B}(S_{0,i}, 0, p_i), i = 1, \ldots, n$ to compute a subset of their vertices, and then corrects these y_i^j with Algorithm 2 such that vertices of polytopes $\mathcal{B}(S_{0,i}, S_{f,i}, p_i), i = 1, \ldots, n$ are obtained. The parameter g in Algorithm 3 allows to consider a subset of $\{v^j : j \in \{-1, 1\}^d\}$. It allows to adjust accuracy and computational complexity. Line 1 guarantees that the maximum amount is limited to 2^d. If $g < 2^d$, we propose to stochastically select the $j \in \{1, -1\}^d$ using a uniform distribution. In Line 7, the vector of zeros modeling the non-use of flexibility is inserted into the $g + 1$-th column of V. This is not necessary when $g = 2^d$ because then all vectors are computed, and it can be shown that in this case $\mathbf{0}_d \in \mathcal{A}$. Note that the vector is to be appended only if $S_{f,i} \leq S_{0,i} \alpha_i^d \ \forall i \in \{1, \ldots, n\}$, otherwise $\mathbf{0}_d$ is not included in $\mathcal{B}(S_{0,i}, S_{f,i}, p_i)$. In Lines 15 and 16, the Algorithms 1 and 2 are invoked, respectively.

V. BENCHMARK RESULTS

Now we compare the proposed method using the benchmark for Minkowski sum approximations previously published in [19]. The considered scenario models households with real demand curves and stationary batteries modeled by $\mathcal{B}(S_{0,i}, \frac{1}{2}S_{0,i}, p_i), p_i = (1, \overline{x}_i, \underline{x}_i, \overline{S}_i, \frac{1}{4})$. The battery parameters are sampled from intervals: $\overline{S}_i \in [10.5, 13.5]$ (KWh), $S_{0,i} \in [0, 10.5]$ (kWh), $\overline{x}_i \in [4, 6]$ (kW), and $\underline{x}_i \in [-6, -4]$ (kW) $\forall i \in \{1, \ldots, n\}$, cf. [19] for details and for an indepth discussion of the bnechmark.

Algorithm 3 (Complete Algorithm)

Input $S_{0,i}, S_{f,i}, p_i, i = 1, \ldots, n, g$

1: **if** $g < 2^d$ **then**
2:      init $\mathcal{J}$ ▷ chose g distinct elements in $\{-1, 1\}^d$
3: **else**
4:      $\mathcal{J} \leftarrow \{-1, 1\}^d$
5: **end if**
6: **if** $S_{f,i} \leq S_{0,i} \alpha_i^d \ \forall i \in \{1, \ldots, n\}$ **then**
7:      $V \leftarrow \mathbf{0}_{d \times (g+1)}$ ▷ $d \times (g+1)$ matrix of zeros
8: **else**
9:      $V \leftarrow \mathbf{0}_{d \times (g)}$ ▷ $d \times (g)$ matrix of zeros
10: **end if**
11: **for** $i = 1$ **to** n **do**
12:      $V_i \leftarrow \mathbf{0}_{d \times g}$
13:      $k \leftarrow 1$
14:      **for** $j \in \mathcal{J}$ **do**
15:          $y_i^j \leftarrow \text{Vertex}(S_{0,i}, p_i, j)$
16:          $\tilde{y}_i^j \leftarrow \text{Correction}(S_{0,i}, S_{f,i}, p_i, y_i^j)$
17:          $V_i[:, k] \leftarrow \tilde{y}_i^j$
18:          $k \leftarrow k + 1$
19:      **end for**
20: **end for**
21: $V[:, 1 : g] \leftarrow \sum_{i=1}^{n} V_i$

Output V

We assess the quality of inner approximations via the Unused Potential Ratio (UPR) defined as

$$\text{UPR} := \frac{z_{\text{approx}} - z_{\text{exact}}}{z_{\text{no flex}} - z_{\text{exact}}} \cdot 100. \tag{8}$$

Here z_{approx} represents the solution of an optimization problem, e.g., the minimal cost or peak power, based on the approximation, z_{exact} the solution of the same optimization using the exact feasible region, and $z_{\text{no flex}}$ the solution in a setting without flexibility. If the UPR is close to 0 %, then the approximation and the Minkowski sum yield almost the same result, otherwise, if the UPR is close to 100 %, then there is a large (unused) improvement available in the approximation. There is also the possibility that the UPR value is greater than 100 %, in which case the solution without flexibility, i.e., $\mathbf{0}_d$, gives better results than using the approximation.

We consider the objectives $c^\top \left(x + \sum_{i=1}^{N} d_i \right) \Delta t$ for economic cost and $\left\| x + \sum_{i=1}^{N} d_i \right\|_\infty$ for peak power, where c is the associated cost and d_i the household demand.

To account for uncertainties, UPR values are calculated for each month of a year along with 5 random villages, i.e., sets of households with stationary batteries, the median of which is used for further analysis.

We use Algorithm 3 and a uniformly sampled subset $\mathcal{J} \subseteq \{-1, 1\}^d$ with d^2 distinct vectors to calculate the y^j, i.e., $g = d^2$. Note that g needs to be a function of d, as the number of vertices increases with increasing dimension, e.g., a hypercube has 2^d vertices in d dimensions. To motivate the quadratic dependence, we conducted an experiment with 100 batteries for $12, 24, \ldots 96$ time periods. For each tuple $(n, d) \in \{100\} \times \{12, 24, \ldots, 96\}$, the approximation and
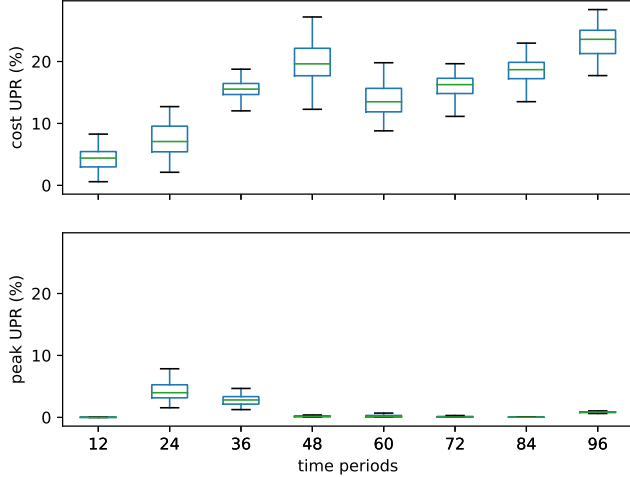
Fig. 4. Boxplot for UPR values with 100 batteries, $d = 12, 14, \ldots, 96$ time periods, and $g = d^2$. For each time period, the approximation is calculated 50 times.

TABLE I

MAX UPR VALUES AND MAX CALCULATION TIME FOR 2, 5, 10, 15 TIME PERIODS AND 5, 10, 15, 20 BATTERIES FOR DIFFERENT INNER APPROXIMATION METHODS

Algorithm	Ref.	Time (s)	UPR (%)	
			Peak	Cost
Cuboid Homothets Stage 0	[15]	0.25	111.92	17.44
Battery Homothets	[9]	37.65	27.25	39.66
Battery Homothet Projection with LDR	[17]	-	-	-
Zonotopes l_∞	[13]	24.64	134.23	21.10
Zonotopes l_1	[13]	27.10	199.32	16.24
Zonotopes l_2	[13]	9.45	121.87	17.34
Zonotopes weighted	[14]	9.23	180.08	14.02
Cuboid Homothets Stage 1	[15]	7.52	111.92	16.21
Ellipsoid Projection with LDR	[16]	8.73	25.07	54.34
Ellipsoid Projection	[7]	-	-	-
Proposed Vertex Generation		**0.07**	**5.38**	**4.19**

UPR values are calculated 50 times to measure the variation in UPR values with different choices of $\mathcal{J} \subseteq \{1, -1\}^d$. The results are shown in Fig. 4. The maximum range of the UPR values for the peak power and cost objectives is 6.3 % and 14.9 %, respectively, indicating a good degree of robustness with different choices of $\mathcal{J} \subseteq \{1, -1\}^d$. Since Algorithm 3 computes $n \cdot d^2$ vertices for n devices, the total number of vectors to compute is nd^2. Thus the computational complexity is quadratic in the number of time periods and linear in the number of devices.

Table I shows the evaluation of 11 approximation algorithms for 5, 10, 15, and 20 batteries and 2, 5, 10, and 15 time periods. For each tuple $(n, d) \in \{5, 10, 15, 20\} \times \{2, 5, 10, 15\}$, the UPR values and calculation times are computed similar to [19]. The maximum computation time and the maximum UPR value for cost and peak objectives across all tuples are listed in the columns for each algorithm. The reason for empty entries is a limitation in the benchmark, which skips algorithms that take longer than 5 minutes for a tuple (n, d) and are thus not calculated for further tuple combinations. Table I shows that

TABLE II

MAXIMUM UPR VALUES AND MAX CALCULATION TIME FOR 12, 24, ..., 96 TIME PERIODS AND 10, 20, ..., 100 BATTERIES.

Algorithm	Time (s)	UPR (%)	
		Peak	Cost
Proposed Vertex Generation	**173.33**	**6.70**	**37.60**

our proposed vertex generation achieves the lowest computation time and UPR values. This solves one of the problems in [19], namely the objective dependent performance, e.g., the algorithm "Ellipsoid Projection with LDR" achieves the second best results at Peak, but the worst results at Cost. Another problem identified in [19] is that the inner approximations may have worse performance than the setting without flexibility, in which case UPR > 100 %. This behavior is observable in the Peak column of Table I for all algorithms except for our approach as well as the algorithms "Battery Homothets" and "Ellipsoid Projection with LDR". This problem is tackled in Line 7 of Algorithm 3, where the vector $\mathbf{0}_d$ is implicitly added in the $g + 1$-th column of V. The last problem mentioned in [19] is the computational complexity. Indeed most algorithms listed in Table I have unrealistic runtimes for day ahead predictions, cf. [19]. However, as the results of Table II show our approach enable efficient day-ahead predictions with 96 time periods. Indeed for a set of 100 batteries the maximal computation time and UPR values are calculated for tuples $(n, d) \in \{10, 20, \ldots, 100\} \times \{12, 24, \ldots, 96\}$ and listed in Table II. It can be seen that the maximum computation time is about 3 minutes. Furthermore, Table II shows that our method exhibits a maximum UPR value for the peak power objective of 6.7 %, which occurs at $(80, 12)$ and is lower than that of the other algorithms, considering only 20 batteries and 15 time periods, cf. Table I. The maximum cost UPR value is reported to be 37.6%, which occurs at $(10, 60)$ and is again better than two of the other algorithms in settings with up to 20 devices and 15 time periods, cf. Table I, and can be further improved by considering more than d^2 vectors.

VI. DISAGGREGATION FOR VERTEX-BASED APPROXIMATIONS

Next, we describe a novel disaggregation method for vertex-based approximations. Once an estimate of the collective flexibility is available, a grid operator, for example, can select a power profile that needs to be distributed (disaggregated) by the aggregator to the individual flexible devices. Mathematically, disaggregation is the inverse operation of aggregation. Aggregation can be described as a mapping $f : \mathcal{P}(A_1, b_1) \times \mathcal{P}(A_2, b_2) \times \cdots \times \mathcal{P}(A_n, b_n) \mapsto \mathbb{R}^d$, $f(x_1, \ldots, x_n) \coloneqq \sum_{i=1}^n x_i$ for polytopes $\mathcal{P}(A_i, b_i) \subset \mathbb{R}^d, i = 1, \ldots, n$. However, this mapping is in general not injective and therefore not invertible, since there may be different sets of vectors with equal sum. Usually, optimization problems are formulated and solved in the literature to obtain feasible vectors whose sum is the aggregated vector, cf. [7], [14]. However, in high-dimensional spaces, this is time-consuming and, indeed, for vertex-based

---

Algorithm 4 (Disaggregation)

Input $\alpha_j, y_i^j, i = 1, \ldots, n, \mathcal{J}$

1: $D \leftarrow \mathbf{0}_{d \times n}$
2: **for** $i = 1$ **to** n **do**
3:   $D[:, i] \leftarrow \sum_{j \in \mathcal{J}} \alpha_j y_i^j$
4: **end for**

Output D

---

approximations it is not necessary. Each aggregate vector x can be described as a convex combination of its vertices, i.e.,

$$x = \sum_{j \in \mathcal{J}} \alpha_j v^j, \sum_{j \in \mathcal{J}} \alpha_j = 1, \alpha_j \geq 0 \ \forall j \in \mathcal{J} \quad (9)$$

for some index set $\mathcal{J}$. Each v^j in the aggregation is a summation of vertices in $\mathcal{P}(A_i, b_i)$, hence $v^j = \sum_{i=1}^n y_i^j$ with $y_i^j \in \mathcal{P}(A_i, b_i)$. Inserting this equality in (9) yields:

$$x = \sum_{j \in \mathcal{J}} \alpha_j \sum_{i=1}^n y_i^j = \sum_{i=1}^n \sum_{j \in \mathcal{J}} \alpha_j y_i^j \quad (10)$$

Thus, the contribution of flexibility i is the sum $\sum_{j \in \mathcal{J}} \alpha_j y_i^j$. Note that the α_j are fixed by the chosen vector x, e.g., from a previously performed optimization by the grid operator, the y_i^j are known from Algorithm 1, and thus the inner sum can be calculated. Therefore, the disaggregation reduces to the calculation of the sum for each flexibility, cf. Algorithm 4.

VII. CONCLUSIONS

This paper proposed a novel vertex-based inner approximation for the collective flexibility of multiple flexible devices. Our method is applicable to polytopes satisfying two rather mild assumptions and we provide an extension to polytopes modeling battery storages. In a benchmark against ten state-of-the-art inner approximations from the literature, our proposed approach outperforms in terms of computational complexity and in terms of accuracy for different objectives. In addition, an efficient disaggregation method is proposed which is applicable to any vertex-based approximation. In combination, the presented methods are to the best of the authors' knowledge the first ones to provide a computationally efficient mean to (dis-)aggregate typical battery storages in quarter-hourly periods over an entire day with reasonable accuracy for aggregate cost *and* peak power optimization objectives.

Future research will focus on the consideration of active and reactive power, and it will attempt to relax the assumptions to other polytopes. In particular, storages with limited availability and (dis-)charging efficiencies, such as electric vehicles, should be considered in this context. Furthermore, it will be investigated if and how cases that violate the assumptions, like storages with terminal energy constraints, can be handled via suitable problem reformulations. Finally, in order to increase the optimization performance at the aggregated level the optimal choice for the vertices subset $\mathcal{J} \subseteq \{1, -1\}^d$ is to be investigated.

A. Proof of Lemma 1

Proof. (Property 1) For $t = 2, \ldots, d$ we have $v_t^j = \sum_{i=1}^n j_t \cdot \max\{j_t \cdot x \in \mathbb{R} : (y_{i,[t-1]}^j, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)\}$. By construction $y_i^j \in \mathcal{P}(A_i, b_i)$, and by Assumption 2 $\mathrm{Proj}^{t-1}(y_i^j) \in \mathcal{P}(A_i, b_i)$, hence $(y_{i,[t-1]}^j, 0, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)$. Therefore, if $j_t = 1$, then $\max\{x \in \mathbb{R} : (y_{i,[t-1]}^j, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)\} \geq 0 \ \forall i$, hence $v_t^j \geq 0$. Otherwise, if $j_t = -1$, then $-\max\{-x \in \mathbb{R} : (y_{i,[t-1]}^j, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)\} \leq 0 \ \forall \ i$, thus $v_t^j \leq 0$.

For $t = 1$ we have $v_1^j = \sum_{i=1}^n j_1 \cdot \max\{j_1 \cdot x \in \mathbb{R} : (x, \mathbf{0}_{d-1})^\top \in \mathcal{P}(A_i, b_i)\}$. Assumption 2 gives $\mathbf{0}_d \in \mathcal{P}(A_i, b_i)$, and by the same reasoning it follows that $v_1^j \geq 0$ if $j_1 = 1$ and $v_1^j \leq 0$ if $j_1 = -1$. $\square$

Proof. (Property 2) Suppose that $v_t^k = v_t^j$ for $k_t \neq j_t$. Without loss of generality let $j_t = 1$ and $k_t = -1$. Then, $v_t^j = \sum_{i=1}^n y_t^j = \sum_{i=1}^n y_t^k = v_t^k$. Since $y_t^j \geq 0$ and $y_t^k \leq 0$ (Property 1), we have $y_t^j = y_t^k = 0 \ \forall \ i$. This is however impossible as by Assumption 2 $\mathrm{Proj}^{t-1}(y_i^j) \in \mathcal{P}(A_i, b_i)$, which implies that $(y_{i,[t-1]}^j, 0, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)$. Furthermore, Assumption 1 ensures the existence of an $\varepsilon \in \mathbb{R} \setminus \{0\}$ with $(y_{i,[t-1]}^j, \varepsilon, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)$, which gives at least two elements, and therefore $\max\{x \in \mathbb{R} : (y_{i,[t-1]}^j, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)\} \neq -\max\{-x \in \mathbb{R} : (y_{i,[t-1]}^k, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_i, b_i)\}$. Thus, the assumption that $v_t^k = v_t^j$ must be false, which yields $v_t^k \neq v_t^j$.

For $j \neq k$, there exists an index t with $j_t \neq k_t$ and by the above reasoning holds $v_t^k \neq v_t^j$, hence $v^j \neq v^k$. $\square$

B. Proof of Lemma 2

Proof. Assume that $p \in \mathcal{M}$ and $t > 1$, then there are $p_i \in \mathcal{P}(A_i, b_i)$ with $p = \sum_{i=1}^n p_i$, and $p_{[t-1]} = \sum_{i=1}^n p_{i,[t-1]} = \sum_{i=1}^n y_{i,[t-1]}^j = v_{[t-1]}^j$. We distinguish two cases:

Case 1: Let $p_{i,[t-1]} = y_{i,[t-1]}^j \ \forall \ i$. If $p_t > v_t^j$ and $j_t = 1$, then there is a $k \in \{1, \ldots, n\}$ with $p_{k,t} > y_{k,t}^j$. Since $p_{k,[t-1]} = y_{k,[t-1]}^j$ and $y_{k,t}^j = \max\{x \in \mathbb{R} : (y_{k,[t-1]}^j, x, \mathbf{0}_{d-t})^\top \in \mathcal{P}(A_k, b_k)\}$ it follows that $(p_{k,[t-1]}, p_{k,t}, \mathbf{0}_{d-t})^\top \notin \mathcal{P}(A_k, b_k)$, thus $\mathrm{Proj}^t(p_k) \notin \mathcal{P}(A_i, b_i)$. Assumption 2 yields $p_k \notin \mathcal{P}(A_k, b_k)$, which contradicts the assumption $p \in \mathcal{M}$. If $p_t < v_t^j$ and $j_t = -1$, then by similar reasoning it follows that $p \notin \mathcal{M}$.

Case 2: $\exists \ l, k \in \{1, \ldots, n\}$ with $p_{l,[t-1]} \neq y_{l,[t-1]}^j$ and $p_{k,[t-1]} \neq y_{k,[t-1]}^j$. Note that the negation of Case 1 yields at least two indices $l, k \in \{1, \ldots, n\}$, and a minimum index $m \in \{1, \ldots, d\}$ with $p_{l,m} < y_{l,m}^j$ and $p_{k,m} > y_{k,m}^j$. For $m \neq 1$ we have $p_{l,[m-1]} = y_{l,[m-1]}^j$ and $y_{l,[m-1]}^j = p_{k,[m-1]}$. If $j_m = -1$ then $\mathrm{Proj}^m(p_l) \notin \mathcal{P}(A_l, b_l)$ and by Assumption 2 $p_l \notin \mathcal{P}(A_l, b_l)$. Otherwise, if $j_m = 1$, then $\mathrm{Proj}^m(p_k) \notin \mathcal{P}(A_k, b_k)$ and by Assumption 2 $p_k \notin \mathcal{P}(A_k, b_k)$. For $m = 1$ we have that $p_{l,1} < y_{l,1}^j$ and $p_{k,1} > y_{k,1}^j$. If $j_1 = 1$, then $p_k \notin \mathcal{P}(A_k, b_k)$. Otherwise, if $j_1 = -1$, then $p_l \notin \mathcal{P}(A_l, b_l)$.

For $t = 1$ holds that if $p_1 > v_1^j$ and $j_1 = 1$, then there is an index $k \in \{1, \ldots, n\}$ with $p_{k,1} > y_{k,1}^j$ therefore $\mathrm{Proj}^1(p_k) \notin \mathcal{P}(A_k, b_k)$ and by Assumption 2 $p_k \notin \mathcal{P}(A_k, b_k)$.

Otherwise, if $p_1 < v_1^j$ and $j_1 = -1$, then by similar reasoning $p_k \notin \mathcal{P}(A_k, b_k)$. Hence, all cases lead to contradictions and therefore $p \notin \mathcal{M}$. □

C. Proof of Proposition 1

Proof. We prove the convex independence of the set of vectors $\{v^j : j \in \{-1,1\}^d\}$ by induction over d. It then follows that v^j is a vertex in $\mathcal{A}$.

Base case: ($d = 1$) In one-dimensional space two distinct numbers $v^{(0)}, v^{(1)}$ are computed cf. Lemma 1, which are convex independent by definition.

Induction hypothesis: Let the set of vectors $\{v^j : j \in \{-1,1\}^d\}$ be convex independent for a $d \in \mathbb{N}$.

Induction step: ($d \to d+1$) The $d+1$ dimensional vectors are constructed by $\{v^{(j,-1)}, v^{(j,1)} : j \in \{-1,1\}^d\}$. Assume the set of vectors $\{v^{(j,-1)}, v^{(j,1)} : j \in \{-1,1\}^d\}$ is convex dependent, then for some $k \in \{-1,1\}^d$ we have without loss of generality that

$$v^{(k,-1)} = \sum_{j \in \{-1,1\}^d, j \neq k} \alpha_j v^{(j,-1)} + \sum_{j \in \{-1,1\}^d} \beta_j v^{(j,1)} \quad (11)$$

$$\sum_{j \in \{-1,1\}^d, j \neq k} \alpha_j + \sum_{j \{-1,1\}^d} \beta_j = 1 \quad (12)$$

$$\alpha_j, \beta_j \geq 0 \quad (13)$$

Projecting (11) to the first d coordinates gives:

$$v^k = \beta_k v^k + \sum_{j \in \{-1,1\}^d, j \neq k} (\alpha_j + \beta_j) v^j$$

$$(1 - \beta_k) v^k = \sum_{j \in \{-1,1\}^d, j \neq k} (\alpha_j + \beta_j) v^j$$

where $0 \leq \beta_k \leq 1$. We distinguish two cases:
Case 1: If $\beta_k < 1$, then $1 - \beta_k > 0$ and we have:

$$v^k = \sum_{j \in \{-1,1\}^d, j \neq k} \frac{(\alpha_j + \beta_j)}{(1 - \beta_k)} v^j$$

where $\frac{(\alpha_j + \beta_j)}{(1 - \beta_k)} \geq 0$ and $\sum_{j \in \{-1,1\}^d, j \neq k} \frac{(\alpha_j + \beta_j)}{(1 - \beta_k)} = 1$, hence v^k is convex combination of vectors in $\{v^j : j \in \{-1,1\}^d\} \setminus \{v^k\}$, which contradicts the induction hypothesis.
Case 2: If $\beta_k = 1$, then $\alpha_j = \beta_j = 0 \ \forall j \in \{-1,1\}^d \setminus \{k\}$, and it follows from (11) that $v^{(k,-1)} = v^{(k,1)}$, which is impossible as the vectors are distinct by Lemma 1. These contradictions show the convex independence of the vectors.

Since $\mathcal{A} = \text{Conv}(\{v^j : j \in \{-1,1\}^d\})$, and v^k for any $k \in \{-1,1\}^d$ is not a convex combination of vectors in $\{v^j : j \in \{-1,1\}^d\} \setminus \{v^k\}$, it follows that v^k is not a convex combination of vectors in $\mathcal{A} \setminus \{v^k\}$, which proofs that v^k is a vertex of $\mathcal{A}$. □

D. Proof of Proposition 2

Proof. This statement is obvious if $\mathcal{M} \setminus \mathcal{A} = \emptyset$, i.e., $\mathcal{M} = \mathcal{A}$. Therefore, we temporarily suppose that $\mathcal{M} \setminus \mathcal{A} \neq \emptyset$ and we distinguish two cases:
Case 1: Let $p, q \in \mathcal{M} \setminus \mathcal{A}$. Assume that $v^j = tp + (1-t)q$ with $t \in (0,1)$. Since $v^j \in \mathcal{A}$ and $p, q \notin \mathcal{A}$, it follows that $p \neq v^j$ and $q \neq v^j$. Since $t \in (0,1)$, it follows that $p \neq q$,

hence there are indices in $\{1, \ldots, d\}$ where the entries in p and q are different. Let m be the minimum of these indices. For this index holds $v_m^j = tq_m + (1-t)p_m$, $t \in (0,1)$ and $p_t \neq q_t$. Without loss of generality, suppose that $p_m < q_m$, then $p_m < v_m^j < q_m$. Since m is the minimum index, we have equality in v, p and q for the indices $\{1, \ldots, m-1\}$. If $j_m = 1$, then by Lemma 2 we have $q \notin \mathcal{M}$. Otherwise, if $j_m = -1$, then by Lemma 2 we see that $p \notin \mathcal{M}$. Therefore we have a contradiction in both cases and the assumption must be false. Hence there are no $p, q \in \mathcal{M} \setminus \mathcal{A}$ with $v^j = tp + (1-t)q$ and $t \in (0,1)$.
Case 2: Let $p \in \mathcal{M} \setminus \mathcal{A}$ and $q \in \mathcal{A}$. The proof for this case is almost a copy of the previous one. Assume that $v^j = tp + (1-t)q$ with $t \in (0,1)$. Since $p \notin \mathcal{A}$, $q \in \mathcal{A}$ and $t \in (0,1)$ it follows that $v^j \neq p$ and $p \neq q$. Since $p \neq q$, there is a minimum index m where the components of p and q are different. For this index holds $v_m^j = tp_m + (1-t)q_m$ and $q_m \neq p_m$. Without loss of generality assume that $p_m < q_m$. Since m is the minimum index, we have equality in the indices $\{1, \ldots, m-1\}$. If $j_m = 1$, then it follows by Lemma 2 that $q \notin \mathcal{M}$ otherwise, if $j_m = -1$, then by the same reasoning it follows that $p \notin \mathcal{M}$. This shows that there are no $p \in \mathcal{M} \setminus \mathcal{A}$ and $q \in \mathcal{A}$ with $v^j = tp + (1-t)q$ and $t \in (0,1)$.

In conclusion, we see that the only possible case is $p, q \in \mathcal{A}$. This concludes the proof. □

E. Proof of Lemma 3

Proof. If $y^j \in \mathcal{B}(S_0, S_f, p)$, then there is nothing to show. Hence we assume that $y^j \notin \mathcal{B}(S_0, S_f, p)$. If the assignments in Lines 3 or 9 are applied, then

$$\alpha^d S_0 + \sum_{\tau=1}^{d} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t = \alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t$$
$$+ \frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t)}{\Delta t} \Delta t = S_f. \quad (14)$$

Therefore, $\tilde{y}^j \in \mathcal{B}(S_0, S_f, p)$ if there is a correction index $k \in \{1, \ldots, d\}$ such that one of the assignments are applied. If $k \in \{1, \ldots, d-1\}$ it holds that $y_\tau^j = \overline{x} \ \forall \tau \in \{k, \ldots, d-1\}$, and $k = d$ implies correction in Line 3 only.

Assume that $\frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t)}{\Delta t} > \overline{x}$ for all correction indices $k = 1, \ldots, d$, then also for $k = 1$. We have $\overline{x} < \frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \overline{x} \Delta t)}{\Delta t}$. Therefore, $S_f > \alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \overline{x} \Delta t + \overline{x} \Delta t$. Since $\mathcal{B}(S_0, S_f, p) \neq \emptyset$, there exists an $x \in \mathcal{B}(S_0, S_f, p)$ with $S_f \leq \alpha^d S_0 + \sum_{\tau=1}^{d} \alpha^{d-\tau} x_\tau \Delta t$. This gives $\alpha^d S_0 + \sum_{\tau=1}^{d} \alpha^{d-\tau} \overline{x} \Delta t < S_f \leq \alpha^d S_0 + \sum_{\tau=1}^{d} \alpha^{d-\tau} x_\tau \Delta t$ hence $\sum_{\tau=1}^{d} \alpha^{d-\tau} \overline{x} < \sum_{\tau=1}^{d} \alpha^{d-\tau} x_\tau$, which implies that there is an index m with $\overline{x} < x_m$ and therefore $x \notin \mathcal{B}(S_0, S_f, p)$, contradicting $x \in \mathcal{B}(S_0, S_f, p)$.

From the above, we have that there are indices k such that $\frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t)}{\Delta t} \leq \overline{x}$. Hence we use the maximum correction index l with $S_f \leq \alpha^d S_0 + \sum_{\tau=1}^{l-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t + \sum_{\tau=l}^{d} \alpha^{d-\tau} \overline{x} \Delta t$ and $S_f > \alpha^d S_0 + \sum_{\tau=1}^{l} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t + \sum_{\tau=l+1}^{d} \alpha^{d-\tau} \overline{x} \Delta t$. Note that this index exists since we assumed that $y^j \notin \mathcal{B}(S_0, S_f, p)$. Suppose that $\frac{S_f - (\alpha^d S_0 + \sum_{\tau=1}^{d-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t)}{\Delta t} < \underline{x}$ for this index, then $S_f <$

$\alpha^d S_0 + \sum_{\tau=1}^{l-1} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t + \sum_{\tau=l}^{d-1} \alpha^{d-\tau} \overline{x} \Delta t + \underline{x} \Delta t$. This gives with $-S_f < -\alpha^d S_0 - \sum_{\tau=1}^{l} \alpha^{d-\tau} \tilde{y}_\tau^j \Delta t - \sum_{\tau=l+1}^{d} \alpha^{d-\tau} \overline{x} \Delta t$ that $0 < -\alpha^{d-l} \tilde{y}_l^j \Delta t + \alpha^{d-l} \overline{x} \Delta t - \overline{x} \Delta t + \underline{x} \Delta t$. Thus $\alpha^{d-l} \tilde{y}_l^j < (\alpha^{d-l} - 1)\overline{x} + \underline{x} \leq \underline{x}$ and hence $\tilde{y}_l^j < 0$. Since $\alpha^{d-l} \in (0,1]$ we have that $\alpha^{d-l} \tilde{y}_l^j \geq \tilde{y}_l^j$. Therefore, $\tilde{y}_l^j \leq \alpha^{d-l} \tilde{y}_l^j < \underline{x}$, hence $\tilde{y}_l^j < \underline{x}$, which is impossible as $y^j \in \mathcal{B}(S_0, 0, p)$ an all corrections were within $[\underline{x}, \overline{x}]$. We conclude that there exists an index such that the assignments in Line 3 or 9 are applied, and $\tilde{y}^j \in \mathcal{B}(S_0, S_f, p)$. $\square$

F. Proof of Theorem 2

Proof. Assume that $\tilde{y}^j$ is not a vertex of $\mathcal{B}(S_0, S_f, p)$, then there are $p, q \in \mathcal{B}(S_0, S_f, p)$ with $\tilde{y}^j = pt + q(1-t)$, $p \neq q$ and $t \in (0,1)$. We distinguish two cases.

Case 1: $\tilde{y}^j = y^j$ and therefore y^j is not changed by Algorithm 2. Since $p, q \in \mathcal{B}(S_0, S_f, p)$, and $\mathcal{B}(S_0, S_f, p) \subseteq \mathcal{B}(S_0, 0, p)$ we have that $p, q \in \mathcal{B}(S_0, 0, p)$. Therefore we have $y^j = \tilde{y}^j = pt + q(1-t)$, $p \neq q$ and $t \in (0,1)$. Hence y^j is not a vertex of $\mathcal{B}(S_0, 0, p)$, contradicting Theorem 1.

Case 2: $\tilde{y}^j \neq y^j$ and therefore y^j is changed by Algorithm 2. Since $\tilde{y}^j \neq y^j$, there exists a maximum correction index f with $\tilde{y}_{[f-1]}^j = y_{[f-1]}^j$ and $\tilde{y}_t^j = \overline{x} \ \forall t \in \{f, \ldots, d-1\}$. Since $p \neq q$, there is a minimum index m with $p_{[m-1]} = p_{[m-1]}$ and $p_m \neq q_m$. Without loss of generality let $p_m > q_m$. If $f > m$, then $\tilde{y}_{[m]}^j = y_{[m]}^j$, hence $p_{[m-1]} = y_{[m-1]}^j = q_{[m-1]}$ and $p_m > y_m^j > q_m$. From this it follows that $\text{Proj}^m(p) \notin \mathcal{B}(S_0, 0, p)$ or $\text{Proj}^m(q) \notin \mathcal{B}(S_0, 0, p)$. With Assumption 2 we have that $p \notin \mathcal{B}(S_0, 0, p)$ or $q \notin \mathcal{B}(S_0, 0, p)$, which contradicts $p, q \in \mathcal{B}(S_0, S_f, p)$. Hence, it holds that $f \leq m$. Moreover, $\tilde{y}_t^j = \overline{x}$, $\forall t \in \{f, \ldots, d-1\}$ holds. For $m \neq d$, the inequality $\tilde{y}_m^j < q_m$ implies that $q \notin \mathcal{B}(S_0, S_f, p)$. Hence $m = d$, which gives $q_{[d-1]} = p_{[d-1]} = \tilde{y}_{[d-1]}^j$ and $p_d < \tilde{y}_d^j < q_d$. This is, however, impossible as by Algorithm 2 $\alpha^d S_{0,k} + \sum_{\tau=1}^{d} \alpha^{d-\tau} \tilde{y}_k^j \Delta t = S_{f,k}$. Using p instead leads to $\alpha^d S_{0,k} + \sum_{\tau=1}^{d} \alpha^{d-\tau} p_k \Delta t < S_{f,k}$. Hence we conclude $p \notin \mathcal{B}(S_{0,k}, S_{f,k}, p_k)$. $\square$

REFERENCES

[1] L. Gkatzikis, I. Koutsopoulos and T. Salonidis, "The Role of Aggregators in Smart Grid Demand Response Markets", in IEEE Journal on Selected Areas in Communications, vol. 31, no. 7, pp. 1247-1257, July 2013, doi: 10.1109/JSAC.2013.130708.

[2] K. Worthmann, C.M. Kellett, L. Grüne, S.R. Weller, "Distributed control of residential energy systems using a market maker", in IFAC Proceedings Volumes, vol. 47, no. 3, 11641–11646, 2014, doi: 10.3182/20140824-6-ZA-1003.01785.

[3] H. R. Tiwary, "On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes," Discrete Comput Geom, vol. 40, no. 3, pp. 469–479, Oct. 2008, doi: 10.1007/s00454-008-9097-3.

[4] S. Taheri, V. Kekatos, H. Veeramachaneni, and B. Zhang, "Data-Driven Modeling of Aggregate Flexibility under Uncertain and Non-Convex Device Models," IEEE Trans. Smart Grid, pp. 1–1, 2022, doi: 10.1109/TSG.2022.3185532.

[5] I. A. Sajjad, G. Chicco, and R. Napoli, "Definitions of Demand Flexibility for Aggregate Residential Loads," IEEE Transactions on Smart Grid, vol. 7, no. 6, pp. 2633–2643, Nov. 2016, doi: 10.1109/TSG.2016.2522961.

[6] S. Koch, J. L. Mathieu, and D. S. Callaway, "Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services," in in Proc. Power Systems Computation Conf, 2011.

[7] S. Barot, "Aggregate load modeling for Demand Response via the Minkowski sum," Thesis, University of Toronto, 2017. Accessed: Oct. 19, 2020. [Online]. Available: https://tspace.library.utoronto.ca/handle/1807/78943

[8] S. Barot and J. A. Taylor, "A concise, approximate representation of a collection of loads described by polytopes," International Journal of Electrical Power & Energy Systems, vol. 84, pp. 55–63, Jan. 2017, doi: 10.1016/j.ijepes.2016.05.001.

[9] L. Zhao, W. Zhang, H. Hao, and K. Kalsi, "A Geometric Approach to Aggregate Flexibility Modeling of Thermostatically Controlled Loads," IEEE Transactions on Power Systems, vol. 32, no. 6, pp. 4721–4731, Nov. 2017, doi: 10.1109/TPWRS.2017.2674699.

[10] 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, doi: 10.1109/TPWRS.2014.2328865.

[11] He Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "A generalized battery model of a collection of Thermostatically Controlled Loads for providing ancillary service," in 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, Oct. 2013, pp. 551–558. doi: 10.1109/Allerton.2013.6736573.

[12] Y. Wen, Z. Hu, S. You, and X. Duan, "Aggregate Feasible Region of DERs: Exact Formulation and Approximate Models," IEEE Transactions on Smart Grid, vol. 13, no. 6, pp. 4405–4423, Nov. 2022, doi: 10.1109/TSG.2022.3179998.

[13] F. L. Müller, O. Sundström, J. Szabó, and J. Lygeros, "Aggregation of energetic flexibility using zonotopes," in 2015 54th IEEE Conference on Decision and Control (CDC), Dec. 2015, pp. 6564–6569. doi: 10.1109/CDC.2015.7403253.

[14] F. L. Müller, J. Szabó, O. Sundström, and J. Lygeros, "Aggregation and Disaggregation of Energetic Flexibility From Distributed Energy Resources," IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 1205–1214, Mar. 2019, doi: 10.1109/TSG.2017.2761439.

[15] 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 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, Dec. 2018, pp. 5708–5715. doi: 10.1109/CDC.2018.8618731.

[16] J. Zhen and D. den Hertog, "Computing the Maximum Volume Inscribed Ellipsoid of a Polytopic Projection," INFORMS Journal on Computing, vol. 30, no. 1, pp. 31–42, Jan. 2018, doi: 10.1287/ijoc.2017.0763.

[17] L. Zhao, H. Hao, and W. Zhang, "Extracting flexibility of heterogeneous deferrable loads via polytopic projection approximation," in 2016 IEEE 55th Conference on Decision and Control (CDC), Dec. 2016, pp. 6651–6656. doi: 10.1109/CDC.2016.7799293.

[18] R. R. Appino, V. Hagenmeyer and T. Faulwasser, "Towards Optimality Preserving Aggregation for Scheduling Distributed Energy Resources," in IEEE Transactions on Control of Network Systems, vol. 8, no. 3, pp. 1477-1488, Sept. 2021, doi: 10.1109/TCNS.2021.3070664.

[19] E. Öztürk, K. Rheinberger, T. Faulwasser, K. Worthmann, and M. Preißinger, "Aggregation of Demand-Side Flexibilities: A Comparative Study of Approximation Algorithms," Energies, vol. 15, no. 7, p. 2501, Mar. 2022, doi: 10.3390/en15072501.

[20] K. Trangbaek, M. Petersen, J. Bendtsen, and J. Stoustrup, "Exact power constraints in smart grid control," in IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, Dec. 2011, pp. 6907–6912. doi: 10.1109/CDC.2011.6160539.

[21] M. Conforti, G. Cornuéjols, and G. Zambelli, Integer Programming, vol. 271. in Graduate Texts in Mathematics, vol. 271. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-11008-0.

[22] J. Matoušek, Ed., Lectures on Discrete Geometry, vol. 212. New York, NY: Springer New York, 2002. doi: 10.1007/978-1-4613-0039-7.

[23] F. A. Taha, T. Vincent, and E. Bitar, "A Multi-Battery Model for the Aggregate Flexibility of Heterogeneous Electric Vehicles." arXiv, Apr. 13, 2023. Accessed: Apr. 24, 2023. [Online]. Available: http://arxiv.org/abs/2304.06769

[24] M. Zhang, Y. Xu, X. Shi and Q. Guo, "A Fast Polytope-based Approach for Aggregating Large-Scale Electric Vehicles in the Joint Market Under Uncertainty," in IEEE Transactions on Smart Grid, doi: 10.1109/TSG.2023.3274198.

[25] J. Jian, M. Zhang, Y. Xu, W. Tang and S. He, "An Analytical Polytope Approximation Aggregation of Electric Vehicles Considering Uncertainty for the Day-Ahead Distribution Network Dispatching," in IEEE Transactions on Sustainable Energy, doi: 10.1109/TSTE.2023.3275566.

[26] M. P. Evans, S. H. Tindemans and D. Angeli, "Minimizing Unserved Energy Using Heterogeneous Storage Units," in IEEE Transactions on Power Systems, vol. 34, no. 5, pp. 3647-3656, Sept. 2019, doi: 10.1109/TPWRS.2019.2910388.

[27] M. P. Evans, S. H. Tindemans and D. Angeli, "A Graphical Measure of Aggregate Flexibility for Energy-Constrained Distributed Resources," in

IEEE Transactions on Smart Grid, vol. 11, no. 1, pp. 106-117, Jan. 2020, doi: 10.1109/TSG.2019.2918058.

[28] M. P. Evans, D. Angeli, G. Strbac and S. H. Tindemans, "Chance-Constrained Ancillary Service Specification for Heterogeneous Storage Devices," 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Bucharest, Romania, 2019, pp. 1-5, doi: 10.1109/ISGTEurope.2019.8905723.

[29] A. R. Fioravanti, J. Mareček, R. N. Shorten, M. Souza, and F. R. Wirth, "On the ergodic control of ensembles," Automatica, vol. 108, p. 108483, Oct. 2019, doi: 10.1016/j.automatica.2019.06.035.

[30] J. Mareček, M. Roubalik, R. Ghosh, R. N. Shorten, and F. R. Wirth, "Predictability and fairness in load aggregation and operations of virtual power plants," Automatica, vol. 147, p. 110743, Jan. 2023, doi: 10.1016/j.automatica.2022.110743.

[31] V. Kungurtsev, J. Marecek, R. Ghosh, and R. Shorten, "On the ergodic control of ensembles in the presence of non-linear filters," Automatica, vol. 152, p. 110946, Jun. 2023, doi: 10.1016/j.automatica.2023.110946.

[32] A. Engelmann, M. B. Bandeira, and T. Faulwasser, "Approximate Dynamic Programming with Feasibility Guarantees." arXiv, Jun. 09, 2023. Accessed: Jul. 12, 2023. [Online]. Available: http://arxiv.org/abs/2306.06201

[33] K. Mukhi and A. Abate, "An Exact Characterisation of Flexibility in Populations of Electric Vehicles," 2023, doi: 10.48550/ARXIV.2306.16824.