Scalable Computation of 2D-Minkowski Sum of Arbitrary Non-Convex Domains: Modeling Flexibility in Energy Resources*

Soumya Kundu, Vikas Chandan and Karan Kalsi
Optimization and Control Group
Pacific Northwest National Laboratory, Richland, WA 99352, USA
{soumya.kundu, vikas.chandan, karanjit.kalsi}@pnnl.gov

Abstract

The flexibility of active (p) and reactive power (q) consumption in distributed energy resources (DERs) can be represented as a (potentially non-convex) set of points in the p-q plane. Modeling of the aggregated flexibility in a heterogeneous ensemble of DERs as a Minkowski sum (M-sum) is computationally intractable even for moderately sized populations. In this article, we propose a scalable method of computing the M-sum of the flexibility domains of a heterogeneous ensemble of DERs, which are allowed to be non-convex, non-compact. In particular, the proposed algorithm computes a guaranteed superset of the true M-sum, with desired accuracy. The worst-case complexity of the algorithm is computed. Special cases are considered, and it is shown that under certain scenarios, it is possible to achieve a complexity that is linear with the size of the ensemble. Numerical examples are provided by computing the aggregated flexibility of different mix of DERs under varying scenarios.

1. Introduction

As the penetration of low-inertia renewable generation increases, various forms of distributed energy resources (DERs), including flexible and responsive electrical loads, will be increasingly integrated into the grid operations to support ancillary services. In this context, DER refers to any load and distributed generation that can offer flexibility in net active and reactive power consumption (equivalently, generation). Real-time coordination and control of DERs requires an appropriate modeling and quantification of the loads behavior and their available flexibility. Modeling of aggregated flexibility in an ensemble of flexible loads for ancillary services (in particular, frequency regulation and ramping) have been explored in the literature in recent years [1–7]. The proposed approaches are generally applicable to ensembles of similar loads, such as a collection of residential air-conditioners, or a collection of plug-in electric vehicles. The aggregation of flexibility for a heterogeneous group of DERs, however, remains a challenging task.

In order to efficiently coordinate tens of thousands of flexible loads in distribution systems, while also satisfying line-flow and node voltage constraints, hierarchical modeling and control frameworks have been proposed [8, 9]. Hierarchical architectures alleviate the computational complexity of the optimal DER dispatch problem, by having resource aggregators to participate in the network optimization problem, instead of the individual DERs. Resource aggregators, referred to here as the *aggregate device controller* (or ADC), are tasked with the responsibility of aggregating the flexibility, in active (p) and reactive power (q), of the neighboring DERs locally at the level of a couple of service transformers (tens of residential customers). The mix of such DERs is likely to be heterogeneous, e.g. a random collection of air-conditioners, electric water-heaters, batteries, solar photovoltaic inverers, wind inverters, etc. Flexibility of each DER can be represented as a union of points or domains in the p-q plane. Aggregated (p, q)-flexibility of such an ensemble is a Minkowski sum (M-sum) of the flexibility sets of the individual DERs, and can be non-convex, non-compact. In recent work, [10], authors presented a geometric approach to approximate the aggregated flexibility domain of an ensemble of heterogeneous DERs using convex polygons. However, approximation of the true aggregated flexibility domain via computation of the exact M-sum of the individual flexibility domains still remains a computational challenge, with computational time increasing exponentially with the ensemble size.

Efficient computation of 2D M-sum of arbitrary polygons (convex/non-convex) have generated interest (primarily) in the field of computational geometry [11–15], resulting in publicly available tools [12, 16]. Convolution based methods [12,14,15] have been shown to perform better than polynomial-decomposition based methods [13]. However, applicability and performance guarantees of such algorithms for M-sum of large

HICSS

number of arbitrary non-polygon 2D domains is not obvious. In this paper we propose a computationally tractable method of approximating the aggregated flexibility for an arbitrary collection of DERs. The rest of the paper is organized as follows. In Sec. 2 we describe the problem of aggregating flexibility of heterogeneous DERs. Sec. 3 presents the key idea behind the proposed approach, while the detailed algorithmic steps along with an analysis of its computational performance is discussed in Sec. 4. Numerical examples are presented in Sec. 5, before we conclude the article in Sec. 6.

2. Problem Description

In the context of this article, any energy (consuming or generating) resource which offers certain flexibility in active (p) and/or reactive (q) power, possibly over a reasonably short time window (such as a 5-15 min long control period), is considered as a DER. We use the notation $\mathcal{F}$ to represent the flexibility domain as a collection of (p, q)-points that are physically admissible by the DER (possibly via some local device-level control). Note that the flexibility domain ($\mathcal{F}$) could be a continuous or a discrete domain. For example, a solar photovoltaic (PV) inverter that can modulate its active and reactive power over a continuous range will have a continuous flexibility domain, while the flexibility domain for switching loads, such as an air-conditioner or electric water-heater, will be discrete. Moreover this flexibility is time-varying, and depends on exogenous parameters as well as end-user preferences.

Figure 1 shows examples of flexibility domains for certain types of DERs, with positive (negative) values of p and q denoting consumption (generation). Discrete flexibility domain of a switching load (e.g. air-conditioner) that operates in two discrete operational states ('on' and 'off') is shown in Figure 1(b), while the rest of the plots represent continuous flexibility domains. Batteries (Figure 1(a)) offer full four-quadrant flexibility, while PV (Figure 1(c)) and wind (Figure 1(d)) inverters offer flexibility only on the left half-plane (active power generation). Note that the flexibility offered by the DERs is dynamic, and change based on end-usage and exogenous influence. For example, if there is a cloudy sky, the PV inverter output might only be restricted to a small fraction of its rated generation. Similarly, the air-conditioner may be operating mostly in 'on'-state when the outside air-temperature is high.

Let us assume that there are N DERs within some heterogeneous ensemble. The flexibility domain of the i-th DER is denoted by $\mathcal{F}_i$, such that its active and reactive power consumption (with negative value

signifying net generation)

$$(p_i, q_i) \in \mathcal{F}_i, \quad i \in \{1, 2, \ldots, N\}.$$

The goal of the flexibility aggregation task is to find the net flexibility domain $\mathcal{F}_\Sigma$ in the form of a Minkowski sum of the individual DER flexibility domains ($\mathcal{F}_i$), such that,

$$\mathcal{F}_\Sigma := \biguplus_{i=1}^{N} \mathcal{F}_i = \left\{ (p, q) \; \middle| \; \begin{array}{c} p = \sum_{i=1}^{N} p_i \\ q = \sum_{i=1}^{N} q_i \\ (p_i, q_i) \in \mathcal{F}_i \; \forall i \end{array} \right\}.$$

Calculating the exact M-sum of the individual flexibility domains is computationally complex, especially as the number of DERs increases. Thus, from a practical point-of-view, a desirable approach is to construct approximations of the aggregate flexibility domain in a scalable way. In this paper, we propose a computationally tractable method to approximate, with arbitrary accuracy, the aggregate flexibility of the (heterogeneous) ensemble of DERs.

3. Flexibility Aggregation: Key Idea

In general, the flexibility domains of DERs can be represented in the form of:

$$\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}^k \tag{1a}$$

$$\forall k : \mathcal{F}^k = \left\{ (p, q) \; \middle| \; \begin{array}{c} p \in \left[\underline{p}^k, \overline{p}^k\right] \\ q \in \left[\underline{g}^k(p), \overline{g}^k(p)\right] \end{array} \right\}, \tag{1b}$$

where K is some positive integer; and $\underline{g}^k(p)$ and $\overline{g}^k(p)$ represent the flexibility in reactive power at any given active power value within the range between $\underline{p}^k$ and $\overline{p}^k$. This representation can be used to represent the continuous and discrete flexibility domains of the types depicted in Figure 1, as illustrated below.

Example 1 (BATTERIES) *The flexibility domain of a battery, with a maximum charge/discharge rate of $p^{\max}$ and the apparent power rating of $s > p^{\max}$, is given by*

$$\mathcal{F} = \left\{ (p, q) \; \middle| \; p \in [-p^{\max}, p^{\max}], |q| \leq \sqrt{s^2 - p^2} \right\}.$$

Example 2 (WIND INVERTERS) *The flexibility domain of a wind inverter, with a maximum active power generation of $p^{\max}$ and the apparent power ratings $s_1 > \sqrt{\alpha} p^{\max}$ (due to rotor current limits) and $s_2 > \sqrt{\alpha} p^{\max}$*

(a) batteries   (b) switching loads   (c) PV inverters   (d) wind inverters
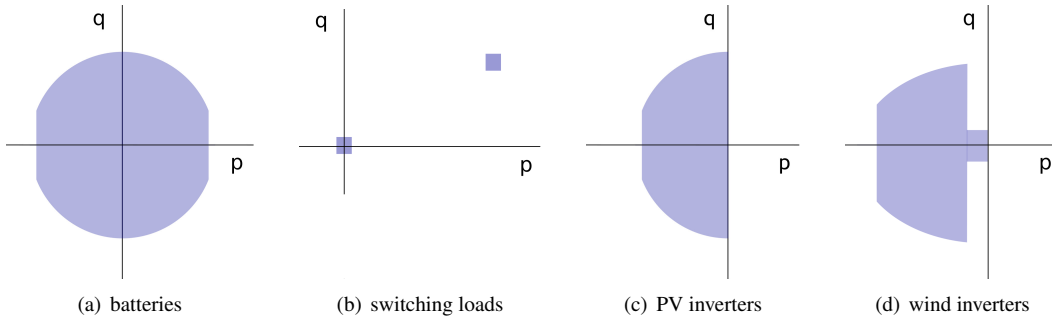
Figure 1: Representation of continuous and discrete flexibility domains for an individual DER of certain types.

(due to stator current limits), is given by [17–19]

$$\mathcal{F} = \mathcal{F}^1 \bigcup \mathcal{F}^2 \bigcup \mathcal{F}^3$$

$$\mathcal{F}^1 = \left\{ (p,q) \,\middle|\, p \in [-p^0, 0], \, q \in [-q^0, q^0] \right\}$$

$$\mathcal{F}^2 = \left\{ (p,q) \,\middle|\, \begin{array}{c} p \in [-p^{\max}, -p^0) \\ 0 \le q \le \sqrt{s_2^2 - \alpha p^2} \end{array} \right\}$$

$$\mathcal{F}^2 = \left\{ (p,q) \,\middle|\, \begin{array}{c} p \in [-p^{\max}, -p^0) \\ -\sqrt{s_1^2 - \alpha p^2} \le q \le 0 \end{array} \right\}$$

where p^0 and q^0 are much smaller than the rated capacities, and $\alpha > 0$.

Example 3 *(AIR-CONDITIONERS) The flexibility domain of a residential air-conditioner with an active power consumption rating of $p^{\max}$ (equal to the power consumed in 'on' state) is represented by*

$$\mathcal{F} = \mathcal{F}^1 \cup \mathcal{F}^2, \, \mathcal{F}^1 = \{(0,0)\}, \, \mathcal{F}^2 = \{(p^{\max}, \gamma p^{\max})\}$$

where $\gamma > 0$ is related to the power factor.

Note that the individual flexibility domains can be continuous and discrete, as well as of arbitrary shape and size. Unfortunately, computation of the exact M-sum is possible only in some specific cases. For example, the M-sum of two (semi-)circles is another (semi-)circle with a radius equal to the sum of the radii of the individual (semi-)circles and its center at the vector sum of the centers of the individual (semi-)circles. As another example, the M-sum of two discrete (on/off) loads with individual flexibility domains $\{(0,0), (p_1, q_1)\}$ and $\{(0,0), (p_2, q_2)\}$, is given by $\{(0,0), (p_1, q_1), (p_2, q_2), (p_1 + p_2, q_1 + q_2)\}$. However, there does not exist a generic algorithm with guaranteed performance for computation of M-sum of continuous sets with arbitray shape (non-polygons). Moreover, the computational time required to compute the M-sum of discrete on/off loads

increases exponentially rendering the exact computation intractable even for moderately sized populations.

Therefore it is important to devise a scalable approach that computes a *sufficiently close approximation* of the M-sum of diverse/heterogeneous set of individual domains, including both continuous and discrete domains. Let us first explain the key idea before going into the details of the algorithm in the next section. Note that it is easy to compute the exact M-sum of two domains each of which is modeled as a rectangle. Consider

$$\mathcal{F}_1 = \left\{ (p,q) \,\middle|\, p \in [\underline{p}_1, \overline{p}_1], \, q \in [\underline{q}_1, \overline{q}_1] \right\}$$

$$\mathcal{F}_2 = \left\{ (p,q) \,\middle|\, p \in [\underline{p}_2, \overline{p}_2], \, q \in [\underline{q}_2, \overline{q}_2] \right\}.$$

Note that we allow the boundaries of the rectangle to be non-unique, i.e. it is possible to have $\underline{p}_i = \overline{p}_i$ and/or $\underline{q}_i = \overline{q}_i$. Their M-sum is simply given by

$$\mathcal{F}_\Sigma = \mathcal{F}_1 \cup \mathcal{F}_2$$

$$= \left\{ (p,q) \,\middle|\, \begin{array}{c} p \in [\underline{p}_1 + \underline{p}_2, \, \overline{p}_1 + \overline{p}_2] \\ q \in [\underline{q}_1 + \underline{q}_2, \, \overline{q}_1 + \overline{q}_2] \end{array} \right\}$$

Extension of this to aggregation of domains which are unions of multiple (≥ 1) rectangles is possible as follows. Suppose

$$\mathcal{F}_1 = \bigcup_{i=1}^{m_1} \left\{ (p,q) \,\middle|\, p \in [\underline{p}_1^i, \overline{p}_1^i], \, q \in [\underline{q}_1^i, \overline{q}_1^i] \right\}$$

$$\mathcal{F}_2 = \bigcup_{i=1}^{m_2} \left\{ (p,q) \,\middle|\, p \in [\underline{p}_2^i, \overline{p}_2^i], \, q \in [\underline{q}_2^i, \overline{q}_2^i] \right\},$$

then

$$\mathcal{F}_\Sigma = \mathcal{F}_1 \cup \mathcal{F}_2$$

$$= \bigcup_{i=1}^{m_1} \bigcup_{j=1}^{m_2} \left\{ (p,q) \,\middle|\, \begin{array}{c} p \in [\underline{p}_1^i + \underline{p}_2^j, \, \overline{p}_1^i + \overline{p}_2^j] \\ q \in [\underline{q}_1^i + \underline{q}_2^j, \, \overline{q}_1^i + \overline{q}_2^j] \end{array} \right\}$$

The computational complexity of the above M-sum is $\mathcal{O}(m_1 m_2)$. This forms the basis of our proposed algorithm for approximating M-sum of N (≥ 2) flexibility domains of arbitrary size and shape.

Remark 1 *The above can be extended to shapes other than rectangles too, such as circles. However, in this paper, we will focus our attention to rectangles only.*

4. Minkowski Sum: Scalable Algorithm

Suppose that we have an ensemble of N DERs with flexibility domains given by

$$\mathcal{F}_i = \bigcup_{k=1}^{K_i} \mathcal{F}_i^k \quad \forall i \in \{1, \ldots, N\} \tag{2a}$$

$$\forall k: \ \mathcal{F}_i^k = \left\{ (p, q) \left| \begin{array}{c} p \in \left[\underline{p}_i^k, \overline{p}_i^k\right] \\ q \in \left[\underline{g}_i^k(p), \overline{g}_i^k(p)\right] \end{array} \right. \right\}, \tag{2b}$$

Let us denote by p_i^{inf} and p_i^{sup}, respectively, the *infimum* and *supremum* of the feasible p-points of the i-th DER, while q_i^{inf} and q_i^{sup} represent, respectively, the *infimum* and *supremum* of the feasible q-points, i.e.

$$p_i^{\text{inf}} := \inf\{p \,|\, \exists q \text{ so that } (p, q) \in \mathcal{F}_i\} \tag{3a}$$
$$p_i^{\text{sup}} := \sup\{p \,|\, \exists q \text{ so that } (p, q) \in \mathcal{F}_i\} \tag{3b}$$

$$q_i^{\text{inf}} := \inf\{q \,|\, \exists p \text{ so that } (p, q) \in \mathcal{F}_i\} \tag{3c}$$
$$q_i^{\text{sup}} := \sup\{q \,|\, \exists p \text{ so that } (p, q) \in \mathcal{F}_i\} \tag{3d}$$

Note that for any given positive scalar $\varepsilon > 0$, there exist positive integers $M_p(\varepsilon)$ and $M_q(\varepsilon)$ defined as:

$$M_p(\varepsilon) := \left\lceil \frac{1}{\varepsilon} \sum_{i=1}^{N} \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right) \right\rceil. \tag{4a}$$

$$M_q(\varepsilon) := \left\lceil \frac{1}{\varepsilon} \sum_{i=1}^{N} \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right) \right\rceil. \tag{4b}$$

If we were to split the whole p-range in the aggregate domain into equal-sized bins of length ε, we would need $M_p(\varepsilon)$ of such bins. Similarly to cover the q-range, we need $M_q(\varepsilon)$ bins. For reasons that will become clearer later, we will refer to ε as the *'tightness'* parameter - with smaller ε implying *tighter* results.

We are now in a position to describe the algorithm of approximate computation of M-sum. We will first describe two critical steps of the process, the *initial discretization* step and the *pixelization* step, and then move on the describe the complete algorithm.

4.1. Initial Discretization Step

At the start of the process each of the individual flexibility domains $\mathcal{F}_i$ are discretized into sets of at most $M_p(\varepsilon) \times M_q(\varepsilon)$ rectangular blocks. There are several ways this discretization can be performed. To be specific, in this paper we consider discretization of the following form:

$$\forall i: \ \mathcal{F}_i \mapsto \mathcal{F}_i^U(\varepsilon) = \bigcup_{j=1}^{m_i} \left\{ (p, q) \left| \begin{array}{c} p \in [\underline{p}_i^j, \overline{p}_i^j] \\ q \in [\underline{q}_i^j, \overline{q}_i^j] \end{array} \right. \right\}$$

where, $\overline{p}_i^j - \underline{p}_i^j \leq \dfrac{\varepsilon \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right)}{\sum_{i=1}^{N} \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right)} \quad \forall j = 1, \ldots, m_i$

$\overline{q}_i^j - \underline{q}_i^j \leq \dfrac{\varepsilon \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right)}{\sum_{i=1}^{N} \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right)} \quad \forall j = 1, \ldots, m_i$

$m_i \leq M_p(\varepsilon) \cdot M_q(\varepsilon)$

and $\mathcal{F}_i \subseteq \mathcal{F}_i^U(\varepsilon)$.

Note that, for example, since the discrete switching (on/off) loads have only two discrete points in the feasibility space, for those loads $\mathcal{F}_i = \mathcal{F}_i^U(\varepsilon)$.

Remark 2 *The last condition ensures that the discrete approximation is a superset of the corresponding actual flexibility domain. This condition can be modified to reflect a subset or any other type of approximation.*

4.2. Pixelization Step

Let us consider the M-sum of two domains each of which is represented as a union of at most $M_p(\varepsilon) \times M_q(\varepsilon)$ rectangular blocks, as follows:

$$\forall i \in \{1, 2\}: \ \mathcal{F}_i^U(\varepsilon) = \bigcup_{j=1}^{m_i} \left\{ (p, q) \left| \begin{array}{c} p \in [\underline{p}_i^j, \overline{p}_i^j] \\ q \in [\underline{q}_i^j, \overline{q}_i^j] \end{array} \right. \right\}$$

$$\forall j = 1, \ldots, m_i: \ \overline{p}_i^j - \underline{p}_i^j \leq \dfrac{\varepsilon \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right)}{\sum_{i=1}^{N} \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right)}$$

$$\overline{q}_i^j - \underline{q}_i^j \leq \dfrac{\varepsilon \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right)}{\sum_{i=1}^{N} \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right)}$$

$$m_i \leq M_p(\varepsilon) \cdot M_q(\varepsilon).$$

We want to approximate their M-sum $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$ as another union of at most $M_p(\varepsilon) \times M_q(\varepsilon)$ rectangular blocks, denoted by $\mathcal{F}_\Sigma^U$ such that

$$\mathcal{F}_\Sigma^U \supseteq \mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$$

$$= \bigcup_{i=1}^{m_1} \bigcup_{j=1}^{m_2} \left\{ (p, q) \left| \begin{array}{c} p \in [\underline{p}_1^i + \underline{p}_2^j, \overline{p}_1^i + \overline{p}_2^j] \\ q \in [\underline{q}_1^i + \underline{q}_2^j, \overline{q}_1^i + \overline{q}_2^j] \end{array} \right. \right\}.$$

Clearly,

$$m_1 m_2 \leq M_p(\varepsilon) \cdot M_q(\varepsilon) \implies \mathcal{F}_\Sigma^U = \mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon).$$

When, however, $m_1 m_2 > M_p(\varepsilon) \cdot M_q(\varepsilon)$, we propose the following two steps for appropriate pixelization of the M-sum $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$.

4.2.1. (PIXELIZATION) STEP 1:

We first compute the estimated number of *pixels* on the p- and q-axes. In order to maintain at most $M_p(\varepsilon) \times M_q(\varepsilon)$ pixels, we use the *pixel sizes* on the p-axis and q-axis as:

$$\widehat{\varepsilon}_p = \varepsilon \frac{\sum_{i=1}^{2} \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right)}{\sum_{i=1}^{N} \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right)} \leq \varepsilon \qquad (5a)$$

$$\widehat{\varepsilon}_q = \varepsilon \frac{\sum_{i=1}^{2} \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right)}{\sum_{i=1}^{N} \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right)} \leq \varepsilon \qquad (5b)$$

With the appropriate pixel-sizes $\widehat{\varepsilon}_p$ and $\widehat{\varepsilon}_q$ determined, the rectangular space defined by

$$\left\{ (p,q) \;\middle|\; \begin{array}{l} p \in [\underline{p}_1^{\text{inf}} + \underline{p}_2^{\text{inf}}, \, \overline{p}_1^{\text{sup}} + \overline{p}_2^{\text{sup}}] \\ q \in [\underline{q}_1^{\text{inf}} + \underline{q}_2^{\text{inf}}, \, \overline{q}_1^{\text{sup}} + \overline{q}_2^{\text{sup}}] \end{array} \right\},$$

which is a superset of the aggregated flexibility space $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$, can be represented with the help of a total of $M_p(\varepsilon) \times M_q(\varepsilon)$ equal-sized pixels, $M_p(\varepsilon)$ pixels on the p-axis and $M_q(\varepsilon)$ pixels on the q-axis. Note that

$$\left\lceil \frac{1}{\widehat{\varepsilon}} \sum_{i=1}^{2} \left(p_i^{\text{sup}} - p_i^{\text{inf}}\right) \right\rceil = M_p(\varepsilon)$$

$$\left\lceil \frac{1}{\widehat{\varepsilon}} \sum_{i=1}^{2} \left(q_i^{\text{sup}} - q_i^{\text{inf}}\right) \right\rceil = M_q(\varepsilon).$$

The domain defined by each (k,l)-th pixel, where $k \in \{1, \ldots, M_p(\varepsilon)\}$ and $l \in \{1, \ldots, M_q(\varepsilon)\}$, is defined by:

$$\mathbf{pixel}(k,l) \qquad\qquad\qquad\qquad (6)$$

$$:= \left\{ (p,q) \;\middle|\; \begin{array}{l} (k-1)\widehat{\varepsilon}_p \leq p - \left(p_1^{\text{inf}} + p_2^{\text{inf}}\right) \leq k\widehat{\varepsilon}_p \\ (l-1)\widehat{\varepsilon}_q \leq q - \left(q_1^{\text{inf}} + q_2^{\text{inf}}\right) \leq l\widehat{\varepsilon}_q \end{array} \right\}$$

4.2.2. (PIXELIZATION) STEP 2:

As the final step in the *pixelization* step, we identify the *pixels* that have some overlap with the sum $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$. Recall that the sum is a union of $m_1 m_2$ rectangular blocks. Thus, for every pair of (i,j), $i \in \{1, \ldots, m_1\}$, $j \in$

$\{1, \ldots, m_2\}$, there is a rectangular block in the sum $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$ defined by

$$\left\{ (p,q) \;\middle|\; \begin{array}{l} p \in [\underline{p}_1^i + \underline{p}_2^j, \, \overline{p}_1^j + \overline{p}_2^j] \\ q \in [\underline{q}_1^i + \underline{q}_2^j, \, \overline{q}_1^j + \overline{q}_2^j] \end{array} \right\}.$$

Associated with it are the following indexes:

$$k_0 = \max\left\{ 1, \left\lceil \frac{1}{\widehat{\varepsilon}_p} \left(\underline{p}_1^i + \underline{p}_2^j - p_1^{\text{inf}} - p_2^{\text{inf}} \right) \right\rceil \right\} \quad (7a)$$

$$k_f = \max\left\{ 1, \left\lceil \frac{1}{\widehat{\varepsilon}_p} \left(\overline{p}_1^i + \overline{p}_2^j - p_1^{\text{inf}} - p_2^{\text{inf}} \right) \right\rceil \right\} \quad (7b)$$

$$l_0 = \max\left\{ 1, \left\lceil \frac{1}{\widehat{\varepsilon}_q} \left(\underline{q}_1^i + \underline{q}_2^j - q_1^{\text{inf}} - q_2^{\text{inf}} \right) \right\rceil \right\} \quad (7c)$$

$$l_f = \max\left\{ 1, \left\lceil \frac{1}{\widehat{\varepsilon}_q} \left(\overline{q}_1^i + \overline{q}_2^j - q_1^{\text{inf}} - q_2^{\text{inf}} \right) \right\rceil \right\} \quad (7d)$$

such that the union of all the $\mathbf{pixel}(k,l)$ with $k \in [k_0, k_f]$ and $l \in [l_0, l_f]$ is a superset and an appropriate *pixelization* of the corresponding rectangular block in the sum $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$. Repeating step 2 for every pair of (i,j) in the M-sum, we readily obtain the appropriately pixelated superset of $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$.

Remark 3 *The computational efforts in step 2 for the pixelization are minimal, since it requires only algebraic computation of the indexes* (7) *to obtain the overlap. Moreover, the step 1 is to be done only once for every sum* $\mathcal{F}_1^U(\varepsilon) \cup \mathcal{F}_2^U(\varepsilon)$, *and involve only minimal algebraic calculations. Therefore the computational complexity of the steps 1-2 combined is* $\mathcal{O}(m_1 m_2)$.

4.3. Complete Algorithm

We are finally in a position to state the complete algorithm for scalable computation of the close superset of the M-sum of a heterogeneous collection of N DERs,

$$\mathcal{F}_\Sigma^U \supseteq \mathcal{F}_\Sigma := \bigcup_{i=1}^{N} \mathcal{F}_i. \qquad\qquad (8)$$

The idea is to perform an N-step iterative process to compute the M-sum, where each step $t \in \{0, \ldots, N-1\}$ (we denote $t = 0$ as the initial step) involves the computation of M-sum of a pair of domains. Let us denote by $\mathcal{F}_\Sigma^{U(t+1)}$ the approximation of the M-sum $\mathcal{F}_\Sigma$, at the completion of step-t, such that

$$\mathcal{F}_\Sigma^{U(1)} \subseteq \mathcal{F}_\Sigma^{U(2)} \subseteq \cdots \subseteq \mathcal{F}_\Sigma^{U(N)} := \mathcal{F}_\Sigma^U.$$

The algorithmic steps are outlined as below:

4.3.1. (MINKOWSKI) INITIALIZATION STEP-0:
At this step, we perform two tasks. The first task is to discretize each flexibility domain $\mathcal{F}_i$ into $\mathcal{F}_i^U(\varepsilon)$, a set of at most $M_p(\varepsilon) \times M_q(\varepsilon)$ rectangular blocks, as described in Sec. 4.1. Moreover we initialize $\mathcal{F}_\Sigma^{U(1)}$ as $\mathcal{F}_1^U(\varepsilon)$ (where the DERs are sorted in no particular order), i.e. we perform the following:

$$\text{compute} \quad \mathcal{F}_i^U(\varepsilon) \supseteq \mathcal{F}_i \, \forall i \,. \tag{9a}$$

$$\text{initialize} \quad \mathcal{F}_\Sigma^{U(1)} \leftarrow \mathcal{F}_1^U(\varepsilon) \,. \tag{9b}$$

Note that, by construction, each of $\mathcal{F}_i^U(\varepsilon) \, \forall i$ and $\mathcal{F}_\Sigma^{U(1)}$ is represented by the union of at most $M_p(\varepsilon) \times M_q(\varepsilon)$ rectangular blocks.

4.3.2. (MINKOWSKI) ITERATIVE STEP-$t\,(\geq 1)$:
In the following iterative steps-t, $t = 1, 2, \ldots, N-1$, we perform the *pixelization* on the M-sum of $\mathcal{F}_{t+1}^U(\varepsilon)$ and $\mathcal{F}_\Sigma^{U(t)}$. Applying the two-steps *pixelization* procedure, outlined in Sec. 4.2, on the M-sum of $\mathcal{F}_{t+1}^U(\varepsilon)$ and $\mathcal{F}_\Sigma^{U(t)}$, we obtain

$$\text{pixelize} \quad \mathcal{F}_\Sigma^{U(t+1)} \supseteq \mathcal{F}_{t+1}^U(\varepsilon) \cup \mathcal{F}_\Sigma^{U(t)}$$

By construction, $\mathcal{F}_\Sigma^{U(t+1)}$ is also a union of at most $M_p(\varepsilon) \times M_q(\varepsilon)$ rectangular blocks. Repeat until $t = N-1$.

Remark 4 *It is unnecessary to perform the pixelization at the final step ($t = N - 1$) at which a simple M-sum of $\mathcal{F}_N^U(\varepsilon)$ and $\mathcal{F}_\Sigma^{U(N-1)}$ suffices.*

At the conclusion of $(N-1)$-th step, we set

$$\mathcal{F}_\Sigma^U \leftarrow \mathcal{F}_\Sigma^{U(N)} \,. \tag{10}$$

4.4. Analysis
Recall that, by construction (enforced by both the *initial discretization* as well as the *pixelization* steps), the estimated flexibility domain ($\mathcal{F}_\Sigma^U$) is a guaranteed superset of the true M-sum ($\mathcal{F}_\Sigma$). Moreover, by definition of the *tightness* parameter ε, for every point (p^*, q^*) in the calculated M-sum, there always exists a feasible point (p, q) with $p \in [p^* - \varepsilon, p^* + \varepsilon]$ and $q \in [q^* - \varepsilon, q^* + \varepsilon]$, i.e. within the Euclidean distance of $\sqrt{2}\,\varepsilon$ from (p^*, q^*). Furthermore, because of the *pixelization* step, the *worst-case* computational complexity of each iteration step is $\mathcal{O}\left(M_p(\varepsilon)^2 M_q(\varepsilon)^2\right)$, which brings the *worst-case* complexity of the complete algorithm down to $\mathcal{O}\left(N M_p(\varepsilon)^2 M_q(\varepsilon)^2\right)$. The performance of the algorithm is summarized in the following result:

Theorem 1 *(WORST-CASE COMPLEXITY) For any tightness parameter ε, the algorithm in Sec. 4.3 generates a guaranteed superset of the true M-sum of N arbitrary shaped 2D-domains, such that for every point (p^*, q^*) in the calculated M-sum, there always exists a feasible point (p, q) with*

$$\max\left(|p - p^*|, |q - q^*|\right) \leq \varepsilon \,.$$

Moreover, the worst-case complexity is given by $\mathcal{O}\left(N M_p(\varepsilon)^2 M_q(\varepsilon)^2\right)$.

While Theorem 1 describes the worst-case complexity of the proposed algorithm in the generic case, we discuss some specific scenarios in the rest of this section.

4.4.1. SPECIAL CASE I:
Consider the case when the DERs in the ensemble are similarly sized, i.e.

$$\forall i \,: \quad p_i^{\sup} - p_i^{\inf} = \Delta p \qquad \text{(i.e. uniform)} \tag{11a}$$

$$q_i^{\sup} - q_i^{\inf} = \Delta q \qquad \text{(i.e. uniform)} \tag{11b}$$

Proposition 1 *In the specific case outlined in (11), the computational complexity is $\mathcal{O}\left(N^3/\varepsilon^4\right)$.*

Proof According to the assumption of *uniformity*, the flexibility domain of each DER in the ensemble is discretized into $\Delta p/\varepsilon$-bins on p-axis and $\Delta q/\varepsilon$-bins on q-axis, i.e. a total of $(\Delta p \, \Delta q)/\varepsilon^2$ rectangular blocks (or, pixels). The M-sum of first two DERs (in no particular order) would result in (at most) $(2\Delta p/\varepsilon) \times (2\Delta q/\varepsilon)$-pixels. It is not difficult to see that the M-sum of first $n < N$ DERs can be similarly represented by $n^2 (\Delta p \, \Delta q)/\varepsilon^2$ pixels. Note that, as per (4) and (11)

$$M_p(\varepsilon) \, M_q(\varepsilon) \geq n^2 (\Delta p \, \Delta q)/\varepsilon^2 \quad \forall n \leq N \,.$$

Complexity of the computation of the incremental M-sum of the $(n + 1)$-th DER and the M-sum of the first n DERs is proportional to n^2/ε^4. Complexity of computing the M-sum of N DERs is then proportional to $\sum_{i=1}^N n^2/\varepsilon^4$, resulting in a complexity of $\mathcal{O}\left(N^3/\varepsilon^4\right)$. ∎

It can be argued that the above result holds for many practical scenarios when the ensemble has similarly sized DERs, while guaranteeing a desired accuracy.

4.4.2. SPECIAL CASE II:
In certain scenarios, due to limitations on the computing resources, it might be necessary to put a bound on the number of bins used to pixelize the p- and q-axes. For example, let us consider the scenario when the number of bins on the q-axis

is fixed to $\overline{M}_q$ while the bins on the q-axis are upper bounded by $\overline{M}_p$, i.e.

$$M_q = \overline{M}_q \text{ but } M_p \leq \overline{M}_p, \qquad (12)$$

Proposition 2 *In the specific case when* (12) *and* (11) *hold, the complexity is: 1)* $\mathcal{O}(N^2/\varepsilon^2)$ *for sufficiently large* $\overline{M}_p$, *and 2)* $\mathcal{O}(N/\varepsilon)$ *when* $\overline{M}_p$ *is small. Moreover, the tightness of the calculated M-sum deteriorates linearly with the size of ensemble.*

Proof Follows similarly as the one for Proposition 1. Note that the flexibility domain of each DER can be discretized in a total of $(\Delta p \, \overline{M}_q)/\varepsilon$ pixels. The M-sum of first n DERs can be represented by

$$\begin{cases} n\left(\Delta p \, \overline{M}_q\right)/\varepsilon \text{ pixels if } n < \overline{M}_p \, \varepsilon/\Delta p \\ \\ \text{and } \overline{M}_p \text{ pixels if otherwise.} \end{cases}$$

Therefore, for sufficiently large $\overline{M}_p$, the complexity of computing the M-sum of N DERs can be shown to be proportional to $\sum_{i=1}^{N} n/\varepsilon^2$, resulting in a complexity of $\mathcal{O}(N^2/\varepsilon^2)$. On the other hand, for small $\overline{M}_p$, the complexity of computing the M-sum of N DERs can be shown to be proportional to $\sum_{i=1}^{N} 1/\varepsilon$, resulting in a complexity of $\mathcal{O}(N/\varepsilon)$. For the tightness argument, note from (4) and (11),

$$\varepsilon \geq \max\left(N\Delta p/\overline{M}_p, \ N\Delta q/\overline{M}_q\right),$$

i.e. for every point in the calculated M-sum, there exists a feasible point within some Euclidean distance that scales linearly with N. ∎

5. Numerical Results

We start by illustrating in Figure 2 how M-sum looks like in some arbitrarily generated ensemble of DERs (including air-conditioners, wind inverters, electric water-heaters and solar photovoltaic inverters). In order to better demonstrate the dependence of the computational complexity of the proposed algorithm on the ensemble size and the tightness parameter, we run a set of tests. First we randomly choose a 10 DER ensemble with of one air-conditioners, two water-heaters, two batteries, two wind inverters and three photovoltaic inverters. This population is then replicated multiple times to create ensembles of larger size (in the multiples of 10). Moreover, the tightness parameter ε is varied from 0.01 kW (or, kVAR) to 0.64 kW (or, kVAR). The computation times are calculated by running the algorithm for various combinations of ensemble size and tightness parameters. Moreover, we also run the tests for two different cases: one in which
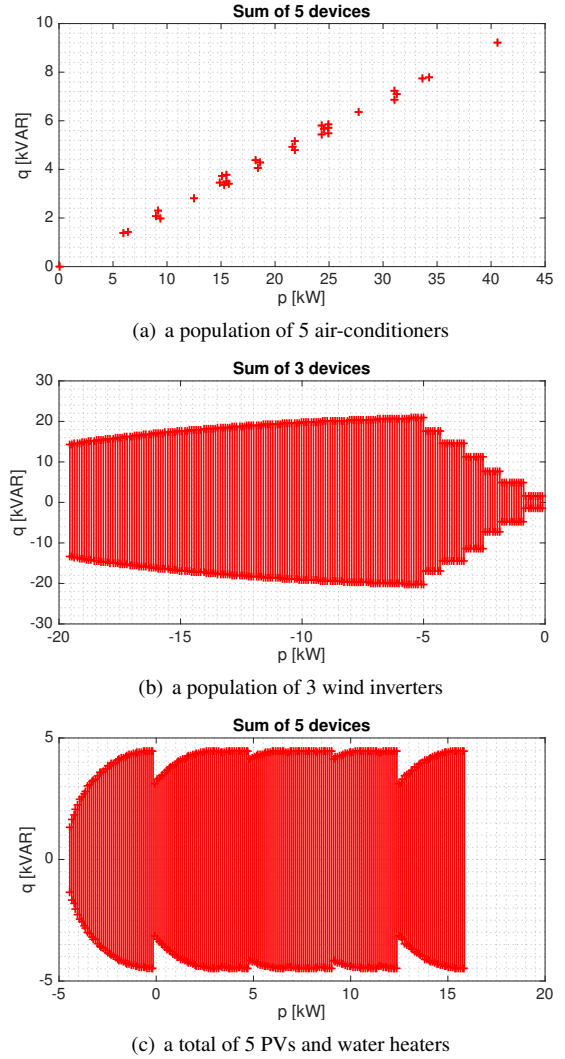

(a) a population of 5 air-conditioners

(b) a population of 3 wind inverters

(c) a total of 5 PVs and water heaters

Figure 2: Aggregated flexibility domains for various arbitrary sample cases.

$\overline{M}_p$ (in Proposition 2) is chosen to be 600, and another in which it is chosen to be 4000 (very high). Figures 3 and 4 present the test results. It can be seen that the results align with the analysis summarized in Proposition 2. Specifically, the computational complexity is seen to be roughly linear with the ensemble size when $\overline{M}_p$ is set to 600 (small), while it is $\mathcal{O}(N^2)$ when $\overline{M}_p$ is set to 4000 (large).

Next, we generate representative DER scenarios, that represent real world test cases to demonstrate the proposed DER flexibility aggregation methodology. The proposed approach entails the following three steps:
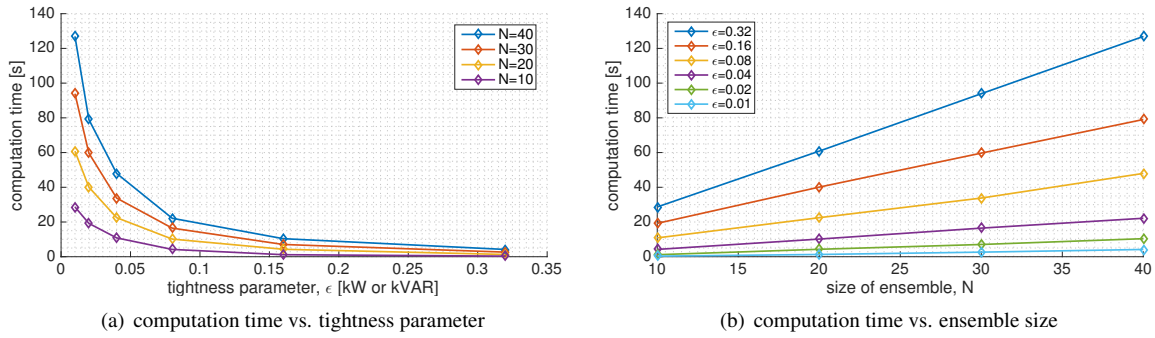
(a) computation time vs. tightness parameter

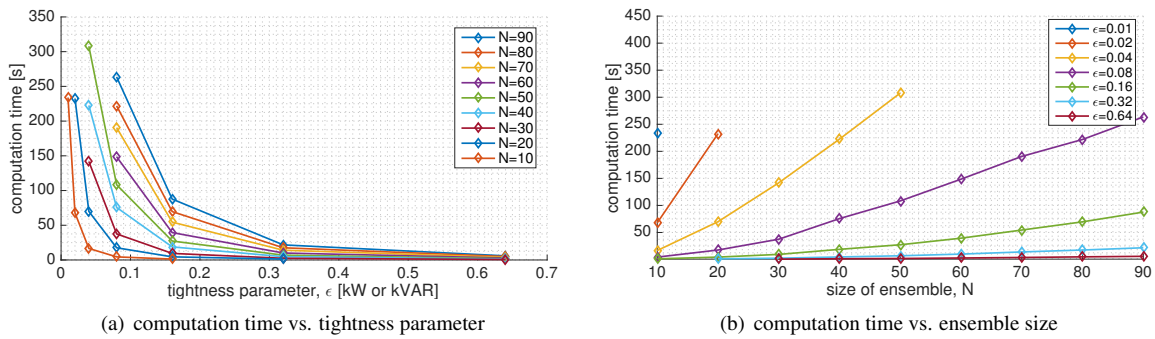(b) computation time vs. ensemble size

Figure 3: Computational complexity with respect to ensemble size (N) and tightness parameter (ε), under the scenario (12) when $\overline{M}_p$ is set to 600 (small).

(a) computation time vs. tightness parameter

(b) computation time vs. ensemble size

Figure 4: Computational complexity with respect to ensemble size (N) and tightness parameter (ε), under the scenario (12) when $\overline{M}_p$ is set to 4000 (large).

5.1. Identify contributing factors

We include five different types of DERs that were discussed previously - heating, ventilation and air-conditioning (HVAC) units, electric water heaters (EWHs), solar photovoltaic (PV) inverters, wind generators, battery electric vehicles (BEVs). We limit ourselves to the consideration of three factors - income level, climate type, and economic policies (with regard to incentives for renewables) - that can potentially affect the penetration of each of these DERs in a certain geographical region. In particular, we assume that the income level and climate type affect the penetration of HVAC and EWHs. For renewable DERs, i.e. PV and wind generators, we assume that climate type and level of incentives determine their prevalence. Lastly, for BEVs, we consider income level and level of incentives as the primary factors affecting their penetration.

We create combinations of the three factors identified above, such that each combination represents a DER scenario. For this we consider the following binary variations of each of these factors: 1) climate type - *mild or extreme*; 2) income level: *high or low*;

and 3) incentive level: *high or low*. To simplify the analysis further, we assume that the climate type classification is with regard to heat and humidity, and correspondingly the HVAC type considered refers to space cooling devices. The above variations result in a total of $2^3 = 8$ scenarios.

5.2. Generate a DER mix per scenario

For each of the above scenarios, we identify a relative mix of DERs expressed as a percentage assignment for each type of DER present in the mix. We use several previous studies and surveys to estimate these assigments, representing a futuristic scenario with a generally high level of DERs. For HVAC, we use data from *Residential Energy Consumption Survey* (RECS) [20] for quantifying the impact of income level, and the climate zone definitions in the Builing America (BA) Program [21] for climate type classification. For renewable DERs (solar and wind), we use data from the *Annual Energy Outlook* (AOE) 2017 [22] which provides projected levels of solar and wind capacity installations expected to be in place by 2050, for 22 US

regions classified by the the *North American Electric Reliability Corporation* (NERC) [23]. For each of these regions, the climate type classification was based on BA, while the incentive level was assigned based on the *Database of State Incentives for Renewables and Efficiency* (DSIRE) [24]. For EWHs, similar to HVAC, data from RECS and BA was used. For BEVs, data from [25] was used to determine the incentive level for 15 states in the US. The income level for each of these states was obtained from [26]. These factors were then mapped to projected sales of BEVs in these states in 2040 based on [27]. The resulting scenarios are shown in Table 1. In Table 1, the percentages for each DER should be interpreted as follows: 1) [HVAC, EWH and BEV]: percentage of consumers which use these resources; 2) [solar and wind]: installed capacity as a percentage of total installed electricity capacity. We use the fact that the total installed electric capacity in the US is about 1074.64 GW [28].

We use the procedure described in this paper to approximate the aggregated flexibility associated with various DER scenarios in Table 1. Using a total population size of 20 DERs, we generate random ensembles based on the percentages from the Table 1. Specifically, we show the plots for scenarios 1, 3, 7 and 8 . Comparing between the scenarios 7 and 8, we see that the increase in penetration of solar PVs show up, for example, as a sharper boundary on the right-hand side. On the other hand, the scenario 3 has a distinctive narrower affine shape, which can be attributed to its relatively higher penetration of air-conditioning loads compared to other loads.

6. Conclusions

This paper is motivated from the need for distributed control and optimization at the power distribution networks. Specifically, the problem of modeling short-term flexibility in ensembles of heterogeneous DERs is addressed. We present a scalable algorithm to approximate, within some specified error tolerance, the aggregated short-term flexibility in an ensemble of DERs. Performance analysis of the proposed algorithm is presented in terms of complexity and accuracy of the results. We show that the algorithm can achieve $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ complexity in some special scenarios, while achieving $\mathcal{O}(N^3)$ complexity in likely realistic scenarios. Numerical results are provided to demonstrate the applicability of the algorithm.

Acknowledgment

References

[1] D. S. Callaway, "Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1389–1400, 2009.

[2] S. Kundu and I. A. Hiskens, "Hysteresis-based charging control of plug-in electric vehicles," in *Proceedings of the IEEE 51st Annual Conference on Decision and Control*, pp. 5598–5604, 2012.

[3] C. Perfumo, E. Kofman, J. H. Braslavsky, and J. K. Ward, "Load management: Model-based control of aggregate power for populations of thermostatcally controlled loads," *Energy Conversion and Management*, vol. 55, pp. 36–48, Mar. 2012.

[4] J. L. Mathieu, S. Koch, and D. S. Callaway, "State estimation and control of electric loads to manage real-time energy imbalance," *IEEE Transactions on Power Systems*, vol. 28, pp. 430–440, Feb 2013.

[5] W. Zhang, J. Lian, C. Y. Chang, and K. Kalsi, "Aggregated modeling and control of air conditioning loads for demand response," *IEEE Transactions on Power Systems*, vol. 28, pp. 4655–4664, Nov 2013.

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

[7] L. Zhao, W. Zhang, H. Hao, and K. Kalsi, "A geometric approach to aggregate flexibility modeling of thermostatically controlled loads," *arXiv preprint arXiv:1608.04422*, 2016.

[8] D. S. Callaway and I. A. Hiskens, "Achieving controllability of electric loads," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 184–199, 2011.

[9] A. Bernstein, L. Reyes-Chamorro, J.-Y. Le Boudec, and M. Paolone, "A composable method for real-time control of active distribution networks with explicit power setpoints. part i: Framework," *Electric Power Systems Research*, vol. 125, pp. 254–264, 2015.

[10] S. Kundu, K. Kalsi, and S. Backhaus, "Approximating Flexibility in Distributed Energy Resources: A Geometric Approach," *ArXiv e-prints*, Mar. 2018.

[11] G. Ramkumar, "An algorithm to compute the minkowski sum outer-face of two simple polygons," in *Proceedings of the twelfth annual symposium on Computational geometry*, pp. 234–241, ACM, 1996.

[12] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr, "On the design of cgal a computational geometry algorithms library," *Software: Practice and Experience*, vol. 30, no. 11, pp. 1167–1202, 2000.

[13] P. K. Agarwal, E. Flato, and D. Halperin, "Polygon decomposition for efficient construction of minkowski sums," *Computational Geometry*, vol. 21, no. 1-2, pp. 39–61, 2002.

[14] R. Wein, "Exact and efficient construction of planar minkowski sums using the convolution method," in *European symposium on algorithms*, pp. 829–840, 2006.

[15] E. Behar and J.-M. Lien, "Fast and robust 2d minkowski sum using reduced convolution," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1573–1578, IEEE, 2011.

[16] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004.

Table 1: Scenarios with varying DER distributions

Scenario	Income level	Climate type	Incentive level	% HVAC	%solar	%wind	%EWHs	%BEVs
1	Low	Mild	Low	78.9	9.6	15.3	41.4	30.6
2	Low	Mild	High	78.9	48.1	16	41.4	47.4
3	Low	Extreme	Low	89.8	8.5	12.6	55.4	30.6
4	Low	Extreme	High	89.8	45.8	13.2	55.4	47.4
5	High	Mild	Low	81.7	9.6	15.3	36.2	36.2
6	High	Mild	High	81.7	48.1	16	36.2	51.5
7	High	Extreme	Low	92.6	8.5	12.6	50.2	36.2
8	High	Extreme	High	92.6	45.8	13.2	50.2	51.5

(a) scenario 1
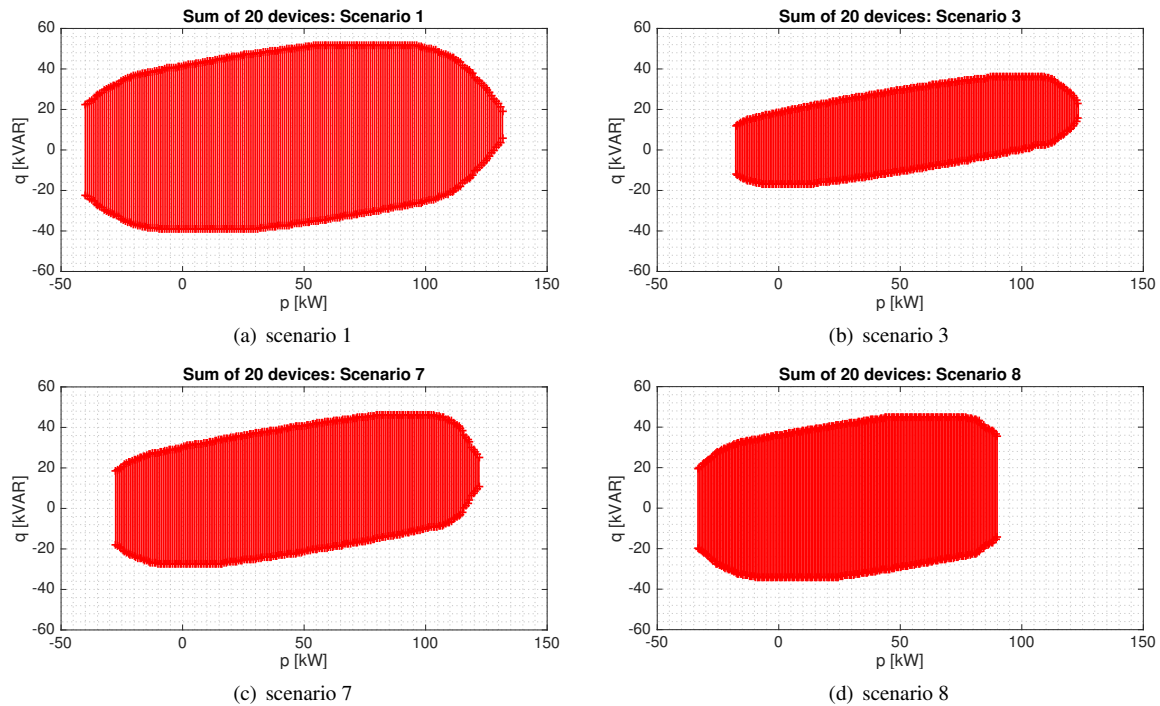

(b) scenario 3

(c) scenario 7

(d) scenario 8

Figure 5: Aggregated flexibility domains for some representative DER scenarios listed in Table 1.

[17] T. Lund, P. Sørensen, and J. Eek, "Reactive power capability of a wind turbine with doubly fed induction generator," *Wind energy*, vol. 10(4), pp. 379–394, 2007.

[18] J. Tian, C. Su, and Z. Chen, "Reactive power capability of the wind turbine with doubly fed induction generator," in *39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 5312–5317, Nov 2013.

[19] J. A. Martin and I. A. Hiskens, "Reactive power limitation due to wind-farm collector networks," in *PowerTech, 2015 IEEE Eindhoven*, pp. 1–6, June 2015.

[20] EIA, "Residential Energy Consumption Survey." https://www.eia.gov/consumption/residential/index.php.

[21] EERE, "Building America Climate-Specific Guidance." Available online from https://energy.gov/eere/buildings/building-america-climate-specific-guidance.

[22] EIA, "Annual Energy Outlook." Available online from https://www.eia.gov/outlooks/aeo/.

[23] EIA, "Electricity Market Module Regions." https://www.eia.gov/outlooks/aeo/pdf/nerc_map.pdf.

[24] NC Clean Energy Technology Center, "Database of Statae Incentives for Renewables and Incentives." http://www.dsireusa.org/.

[25] L. Jin, S. Searle, and N. Lutsey, "Evaluation of state-level us electric vehicle incentives," *The International Council on Clean Transportation*, 2014.

[26] United States Census Bureau, "Median Household Income (in 2013 Inflation-adjusted dollars)." https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?src=bkmk.

[27] Bloomberg New Energy Finance, "Electric Vehicles to be 35% of Global New Car Sales by 2040." https://about.bnef.com/blog/electric-vehicles-to-be-35-of-global-new-car-sales-by-2040/.

[28] EIA, "International Energy Statistics." Available from https://www.eia.gov/beta/international/data/browser/.