Sizing Solar Panels and Storage for Multiple Roofs

Brad Guanqiao Huang
University of Waterloo
brad.huang@uwaterloo.ca

Fiodar Kazhamiaka
Stanford University
fiodar@stanford.edu

Srinivasan Keshav
University of Cambridge
sk818@cam.ac.uk

ABSTRACT

Choosing the number of solar panels and the amount of storage needed to meet a certain fraction of the load in a microgrid setting is a difficult problem that needs to balance the competing objectives of efficiency, robustness, and cost. Prior work in this area makes the unrealistic assumption that solar panels are to be installed on a single roof that is capable of supporting all the panels required. In reality, we may need to deploy solar panels on several roof segments, each of limited size, and each with its own tilt, orientation and installation cost. This paper presents an algorithm for sizing solar panels and storage in this context. We evaluate the robustness of our approach using traces derived from the Pecan Street Dataport dataset and demonstrate the value of our approach by using it to size a hypothetical installation on the British Antarctic Survey's research base in Antarctica.

CCS CONCEPTS

• **Hardware** → **Batteries**; **Renewable energy**.

KEYWORDS

Energy Storage, Solar Power, Sizing

ACM Reference Format:

Brad Guanqiao Huang, Fiodar Kazhamiaka, and Srinivasan Keshav. 2021. Sizing Solar Panels and Storage for Multiple Roofs. In *The Twelfth ACM International Conference on Future Energy Systems (e-Energy '21), June 28–July 2, 2021, Virtual Event, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447555.3464858

1 INTRODUCTION

With the rapid decline in the cost of both solar photovoltaic (PV) generation and storage, solar and storage systems are increasingly adopted to provide carbon-free renewable energy throughout the day [14]. However, PV systems and storage are still quite expensive in absolute terms, so it is necessary to find the smallest possible size of the system that meets electricity load needs [16]. Specifically, given a *solar profile*, the typical hourly solar generation from a single solar panel, and the *load profile*, characterized as hourly demand for a year or more, we would like to choose the number of solar panels and the amount of storage (a *sizing*) so that the load profile is met with a certain quality of service (QoS). We call a sizing that meets the desired QoS criterion a *feasible* sizing.

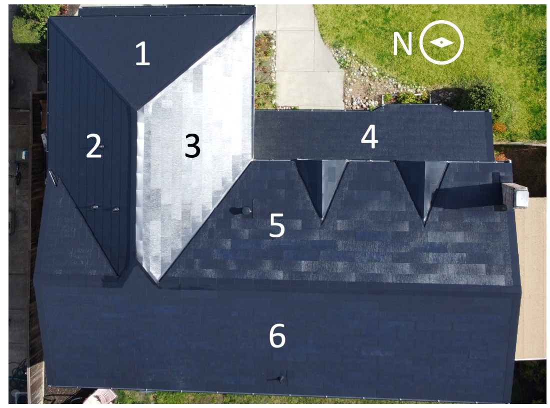

Figure 1: A rooftop with six roof segments.

Choosing a feasible sizing is complex because it needs to balance multiple objectives, including:

- *Pareto efficiency*: It should not be possible to simultaneously reduce both the number of solar panels and the size of the storage without violating the QoS criterion.
- *Robustness*: Feasibility should not be violated due to small variations in solar energy and load in the future. Note that robustness and efficiency cannot be simultaneously satisfied.
- *Cost*: The cost should be as low as possible, taking into account the cost of panels and storage, the installation cost, and the need for robustness.

This important and practical problem has been studied in the literature using approaches ranging from optimization and stochastic network calculus to simulation [16]. However, prior work makes the simplifying assumption that solar panels are to be installed on a single site, typically a rooftop, and that the rooftop is capable of supporting as many solar panels as desired. Moreover, all portions of the roof are assumed to have the same tilt and orientation. In reality, we may wish to deploy solar PV on several roof segments[1], as in Figure 1, each of limited size, with its own tilt, orientation and installation cost. While this may appear to be a trivial change, the problem of sizing solar and storage, even for a system with only two roof segments, is *far* more complex. Intuitively, the reason for this complexity is that some roof segments may have a higher potential for solar energy production or may better match the load profile, but also have higher installation costs or smaller panel capacity. In contrast, others may have lower installation costs and larger capacity, but have a poor solar production potential. Choosing the number of panels to place on each roof segment to produce sufficient energy to meet the QoS criterion at the least overall cost

[1]For uniformity of notation, we use 'roof segment' to mean either the entire roof or a part of it, on one building or several.

e-Energy '21, June 28–July 2, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8333-2/21/06...$15.00
https://doi.org/10.1145/3447555.3464858

is a complex problem and the focus of our work. Specifically, our contributions are:

- We formally state the problem of sizing solar PV and storage on a set of roof segments with different orientations and installation costs and present an algorithm to solve it.
- We implement the algorithm and evaluate its correctness on realistic solar and load traces.
- We use our approach to size a hypothetical solar PV and storage for the British Antarctic Survey's research base in Antarctica, based on real data.

2 RELATED WORK

A person looking to install a rooftop solar PV and battery system wants to know the least-cost sizing that meets their system performance target, e.g. meeting 50% of their load. The optimal sizing depends on the location, tilt, and orientation of each roof segment, as well as the building's load profile and the fixed and marginal costs of system components and installation. In practice, the sizing of solar PV and battery systems has often been done using a "rule-of-thumb"; for example, Tesla [2, 3] takes into account a location, roof layout, and a proxy for load, such as the monthly electricity bill, to calculate a system sizing. Such methods are imprecise, since they rely on aggregate PV generation and load metrics.

In the literature, the sizing of PV and storage systems has been studied in several contexts, including micro-grids [9, 10, 13] and building-scale systems [16, 21, 22]. At a high level, the strategy used by existing work is to assume the availability of load and PV generation measurements for one or more years, which in recent years have become more readily available for consumers [4, 18], and use them to compute a sizing that would be optimal over the given data. Notably, ReOpt Lite [8, 20] is a web-hosted sizing calculator where users can specify location, hourly load profile, roof layout, and other relevant parameters to compute a sizing that maximizes value or robustness metrics.

Our recent work [15, 16] takes into account the stochastic nature of PV generation and load profiles, and proposes a method based for computing the cheapest *robust* sizing that meet a target performance with a specified level of statistical confidence; this work–and others such as ReOpt–rely on the assumption that solar panels are installed on a single roof segment, and cannot be applied to multi-roof environments. In this paper, we present an algorithm for computing a robust sizing for multiple roof segments.

3 PROBLEM FORMULATION

This section presents our model for the sizing problem. We assume that a user, such as a prospective system owner, can obtain (a) a set of solar traces corresponding to the generation from a single panel placed on each roof segment and (b) a representative load trace. We also assume that some storage is required in the system, that the solar panels are connected to the same storage unit, and that there is no loss of power on these connections. Our goal is to find a feasible, robust sizing that is Pareto efficient, and therefore has the least cost. Table 1 shows the notation used in the remainder of the paper. Using this notation, we can make this goal more precise, as follows.

Goal: Given number of roof segments N, time period on which the QoS criterion is calculated T, target unmet load ε, confidence interval γ, fixed and marginal cost for the i^{th} solar panels c_i^f and c_i^m, marginal cost for storage c^b, largest possible number of solar panels that can be installed on the i^{th} roof a_i^{max}, largest possible battery size b^{max}, N solar traces $\mathcal{S}_i$, and the load trace $\mathcal{L}$, find solar sizing $\mathcal{A}$ and storage sizing b such that

- It is feasible, i.e. for all time t, with probability greater than 1-γ
$$\frac{u(\mathcal{A}, b, t, t+T)}{|\mathcal{L}[t, t+T]|} \leq \varepsilon \tag{1}$$
- It minimizes the cost function
$$C(\mathcal{A}, b) = \sum_{i|a_i \neq 0} (c_i^f + c_i^m * a_i) + (c^b * b) \tag{2}$$

where the bound is *robust*, that is, holds for future loads that are statistically the same as in the past. Note that, for any non-zero load profile, a sizing with a sufficiently small number of solar panels is always infeasible.

4 SOLUTION APPROACH

Our solution extends prior work by Kazhamiaka *et. al.* [16], which solved the sizing problem for a single roof segment. The key idea in their work is to simulate the behaviour of an ensemble of historical load and solar traces to construct a *distribution* of feasible and efficient sizings; A Chebyshev bound on this distribution finds the least-cost sizing that meets the QoS with a given confidence.

Specifically, their sizing algorithm first computes the *feasible Pareto frontier* of solar and storage $(\mathcal{A}, b)$ sizing tuples corresponding to each solar and load trace pair. By definition, decreasing the number of panels in a sizing on this frontier necessarily increases the storage capacity and *vice versa*. For a given trace pair, they compute the first point on the frontier by choosing the maximum number of panels and using simulations to compute the amount of unmet load as they progressively increase the battery size from 0. This finds the minimum battery size needed to ensure that the QoS target is met (so that the sizing is both feasible and efficient). They then reduce the amount of solar generation by removing one panel and recompute the minimum battery size to find the next point on the frontier. Given a set of such frontiers, they compute a Chebyshev bound on the set. The least-cost point on this bound is a feasible, robust sizing that also meets the QoS target (it is not Pareto efficient, but then no Pareto efficient sizing is robust). Note that the system cost minimization objective is only brought in at the final step of the algorithm.

We cannot directly use this algorithm in a multi-roof segment setting because computing the entire Pareto frontier turns out to be intractable. Specifically, to compute the next step on the frontier, the reference algorithm reduces the number of panels by one, then finds the corresponding storage needed to meet the QoS bound. When there are N roof segments, since each roof segment has its own energy generation profile corresponding to its tilt and orientation, reducing the number of panels by one on different roof segments would result in different amounts of storage needed to compensate, resulting in a Pareto frontier that is a hyper-surface, not a line. Computing the hyper-surfaces leads to combinatorial

Symbol	Meaning (units)
N	Number of roof segments
i	Index of roof segment
T	Time period over which the which QoS criterion is computed (days)
γ	confidence level for robustness
ε	the target upper bound on unmet energy as a fraction of overall load $\lvert\mathcal{L}\rvert$
c_i^f	Cost of installing a panel on the i^{th} roof segment (\$)
c_i^m	Marginal cost of installing a panel on the i^{th} roof segment (\$)
c^b	Marginal cost of storage (\$/kWh)
a_i^{max}	Largest possible number of panels on the i^{th} roof segment
b^{max}	Largest possible battery size (kWh)
n	Total number of hours provided in each solar/load trace
$\mathcal{S}_i$	An hourly solar power generation trace for one panel on the i^{th} roof segment; $\lvert\mathcal{S}_i\rvert = n$
$\mathcal{S}$	A vector of hourly solar power generation trace for panels on each roof segment; $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$
$\mathcal{S}_i[t_1, t_2]$	A subset of the ith solar trace from time step t_1 to time step t_2 (kW)
$\lvert\mathcal{S}_i[t_1, t_2]\rvert$	Total energy produced by panels allocated in the i^{th} segment from time step t_1 to time step t_2 (kWh)
$\mathcal{L}$	An hourly load trace (kW); $\lvert\mathcal{L}\rvert = n$
$\mathcal{L}[t_1, t_2]$	A subset of the load trace from time step t_1 to time step t_2 (kW)
$\lvert\mathcal{L}[t_1, t_2]\rvert$	Total energy used in the load trace from time step t_1 to time step t_2 (kWh)
a_i	Number of panels, as computed by the algorithm, allocated to i^{th} roof segment; $0 \le a_i \le \mathcal{A}_i^{max}$
$\mathcal{A}$	The sizing, a vector of allocations a_1, a_2, ..., a_N
b	Size of battery, as computed by the algorithm; $0 \le b \le b^{max}$ (kWh)
u	Unmet energy for a given allocation vector, battery size, load trace, and solar trace (kWh)

Table 1: Table of notation for the problem formulation. Note that $\mathcal{A}, b$, and u are outputs of the algorithm and depend on the trace pair ($\mathcal{S}_i, \mathcal{L}$), but for clarity of notation, this dependency is not explicitly denoted.

explosion, and, even if computed, it is not obvious how to find a statistical bound on them.

To fix ideas, consider a roof with five roof segments, indexed $1 \ldots 5$, which can host up to $(a_1, \ldots, a_5)$ panels. Given a trace pair, we first use simulations to compute the storage needed for the allocation $(a_1, \ldots, a_5)$. To find the next points on the Pareto frontier, we would need to compute the storage needed for the allocations $(a_1{-}1, \ldots, a_5), \ldots, (a_1, \ldots, a_5{-}1)$. In the next step, we would need to consider all the ways in which we could reduce the total number of panels by 2, which leads to a combinatorial explosion. As described in the next section, instead of trying to compute the entire Pareto frontier, we use a stochastic gradient descent approach to try to find the least-cost sizing on the Pareto frontier. Intuitively, instead of bringing cost in only at the end, we use the system cost to guide a stochastic exploration of the Pareto frontier.

Symbol	Meaning (units)
η	Number of subintervals sampled, one of each a least-cost sizing is found. Calculated at the beginning of the algorithm as detailed in Section 5.5.
ζ	AdaDelta hyperparameter. Fudge factor for numerical stability. Default 0.1. Refer to Algorithm 2.
ρ	AdaDelta hyperparameter. Decay rate used in calculating exponentially moving average for gradients. Default 0.9. Refer to Algorithm 2.
ξ	A PV-battery sizing pair $(\mathcal{A}, b)$.
μ_η	The empirical mean of all η least-cost sizings.
Σ_η	The empirical covariance of all η least-cost sizings.
Ξ	A set of Chebyshev upper-bounded sizings, found in Section 5.4.
ξ^*	The output of the algorithm, the minimum-cost sizing on boundary Ξ.
Λ^2	The "practicality sizing" factor. Determines the distance between ξ^* and μ_η.
β	Hyperparameter to strike a balance between η and Λ^2. Default 0.1. For details see Section 5.5.

Table 2: Table of notation used only in the sizing algorithm.

5 SIZING ALGORITHM

This section provides a detailed description, pseudocode, and visualization of the algorithm, including the generation of solar and load traces, the system simulation and stochastic gradient descent process, the statistical bound, and how to extract a robust sizing.

5.1 Algorithm Overview

Before diving into the technical details of our algorithm, we give an overview of our solution with reference to later subsections. Additional notation used in this section can be found in Table 2.

At a high level, the process of computing a sizing consists of the following three steps:

(1) Acquire PV generation traces for each roof segment and load traces, which can be from on-site measurements or synthetically generated. Sample η intervals from these traces to form an ensemble of (possibly overlapping) PV generation and load *trace tuples* ($\mathcal{S}_i$, $\mathcal{L}$).
(2) For each trace tuple, find a set of sizings through stochastic gradient descent that meets the target QoS criterion.
(3) Compute a statistical bound on the set of sizings and select a sizing from those found along the bound.

The number of trace tuples, η, represents a trade-off between algorithm runtime and the tightness of the statistical bound on the sizing. The runtime scales linearly with η, while the bound gets asymptotically tighter with more samples; the precise relationship is described later in Section 5.5.

Our approach to finding the least-cost sizing in the search space is to adapt a stochastic gradient descent algorithm. We focus our search on the Pareto-optimal sizing subspace. The unmet load of a sizing is calculated by simulating the operation of the system over the given traces. A simulated gradient of total system costs is computed to guide the direction of the search. The simulation and stochastic gradient descent process are detailed in Section 5.3.

We find the least-cost sizing for every data sub-sample, creating a set of such sizings. We compute the statistical bound on the distribution of sizings represented by the set. As in Reference [15], an empirical multivariate Chebyshev bound [23] is used to calculate a hyper-ellipsoid bound that is centered at the mean the sizings set and scaled according to the desired confidence γ. Finally, we generate a set of sizings that lie on the upper portion of the bound and select the lowest-cost sizing from this set as our output. The process of finding the Chebyshev bound is detailed in Section 5.4.

5.2 Obtaining Solar and Load Traces

The algorithm relies on hourly PV generation (one per roof segment) and load traces ($\mathcal{S}_i, \mathcal{L}$). The user may have access to solar or load traces if they own a pyranometer or a smart electricity meter, although it is possible to obtain both load and per-roof-segment PV generation traces using the proxy techniques described next.

5.2.1 Solar Traces. In the United States, hourly solar traces can be generated through tools such as NREL's PVWatts calculator [7]. The calculator requires parameters that are more easily obtained, including the geographical location of the roof segments, and, for each roof segment, its tilt, orientation, and an estimated performance loss. It then calculates the solar traces based on solar radiation data measured in past years. Alternatively, if hourly horizontal solar irradiance is measured through a pyranometer, then in-plane irradiance can be calculated using the equations in [12].

5.2.2 Load Traces. Unlike solar activity, load traces depend on human behaviours, making generating synthetic load traces more challenging and less reliable. However, recent work has shown that an ARMA model for generating load traces, when trained on load patterns from neighbouring houses where data is available, generates traces that can be used for sizing [24]. Using a load profile database like the one provided by EERE [1], it is often possible to find a load dataset that closely matches the monthly aggregate load values of the target site, which are typically available.

5.3 Finding Minimum-Cost Sizings

Given the solar and load traces, we sample η sub-intervals of length T from each trace using a sliding window approach. creating an ensemble of equally-represented, shorter solar-load subsamples. For each subsample in the ensemble, we want to compute a sizing that meets the target performance at a minimum cost via simulated system operations to obtain the unmet load, denoted as $u(\mathcal{A}, b, t_1, t_2)$ over the time interval $[t_1, t_2]$, as in [15].

A stochastic gradient descent algorithm can be used to quickly find the minimum-cost sizing if the search space is convex and differentiable. However, this does not hold at the edges where a_i goes from 0 to 1 due to fixed per-roof-segment installation costs. To get around this problem, we split the search space into several convex search spaces where fixed costs are ignored: one for each combination of roof segments. For example, if the search space has three roof segments A, B, C, then we have seven sub-spaces: A, B, C, AB, AC, BC, ABC. We then use a stochastic gradient descent algorithm to efficiently search each space and find least-cost sizing across all of search spaces taking into account fixed costs *post hoc*. This mechanism, along with selecting the sub-intervals, are demonstrated later in Algorithm 4 under Section 5.6.

Stochastic gradient descent requires a starting point on the Pareto frontier and a cost function. An initial sizing is the maximum number of panels on each roof segment and a derived minimum storage size b^* that leads to a feasible sizing via binary search as specified in Algorithm 1. The cost function is simply the cost of the system: $C(\mathcal{A}^*) = C(\mathcal{A}^*, b^*)$ via Equation (2).

Algorithm 1 Cost Function for a Solar Allocation

function C($\mathcal{A}$)
  $b^* \leftarrow min_{0 \le b \le b^{max}}\ s.t.\ \frac{u(\mathcal{A}, b, t_1, t_2)}{|\mathcal{L}[t_1, t_2]|} \le \varepsilon$, via binary search
  if b^* does not exist **then**
    return ∞
  end if
  return $C(\mathcal{A}, b^*) = \sum_{i|a_i \neq 0}(c_i^f + c_i^m * a_i) + (c^b * b^*)$
end function

Given the cost function for a solar allocation, we then adjust $\mathcal{A}^*$ via an iteration of a stochastic gradient descent algorithm, with cost function C and a finite difference approximation of gradient, i.e.

$$\frac{\partial C}{\partial a_i}\mathcal{A}^* \approx C(\{a_1, \ldots, a_i + 1, \ldots, a_n\}) - C(\mathcal{A}^*) \tag{3}$$

In our implementation, we experimented with several different stochastic gradient descent algorithms and settled on AdaDelta [26]. Its exponentially decaying gradient mechanism eliminates the need of setting an initial learning rate and we found that it works consistently well with different orders of magnitudes of solar/load trace combinations. Other algorithms such as RMSProp [25] and AdaGrad [11] work equally well but require manual adjustment of learning rate to be efficient.

In our implementation, we set AdaDelta's two hyperparameters, ρ and ζ, to 0.9 and 0.5 respectively. The first hyperparameter represents the decay rate used to calculate an exponentially decaying running average for gradients and the objective function, and the second is used in a division to maintain numerical stability. The stopping condition of the algorithm is when the cost C($\mathcal{A}^*$) exceeds the decaying average of the cost function for previous iterations. The pseudocode for this process is described in Algorithm 2.

Figure 2 shows a typical search path of AdaDelta in two dimensions of PVs. The background color gradient shows the cost at each PV sizing computed via grid search. The cost value starts high near $\mathcal{A}^{max}$ and gradually decreases as allocations on both roof segments decrease, and the gradient stays relatively constant. However, at pv1=3 the cost starts to increase with a unit decrease of pv1, and the search path changes direction to trade-off fewer allocations of pv2 and replace it with more allocations of pv1, keeping in mind cost efficiency. Ultimately the search terminates near the true minimum in the search space, denoted by the X mark.

5.4 Chebyshev Bound

As in Reference [15], we rely on a multivariate concentration bound, based on Theorem 1 in Reference [23] to find a robust sizing by

Algorithm 2 Find the least-cost sizing through system simulation and stochastic gradient descent

function FIND_SIZING(t_1, t_2, subsets)
  $\mathcal{A}^* \leftarrow \{\mathcal{A}_i^{max} : i \in \text{subsets}\}$
  search_path ← empty list
  $S \leftarrow \vec{0}$ ▷ S, Δ are AdaDelta intermediate variables
  $\Delta \leftarrow \vec{0}$
  $m \leftarrow \mathrm{C}(\mathcal{A}^*)$
  while $\mathrm{C}(\mathcal{A}^*) \leq m$ **do**
    search_path.add($\mathcal{A}^*$)
    $\nabla C(\mathcal{A}^*) \leftarrow \{\frac{\partial C}{\partial a_i}\mathcal{A}^* : i \in \text{subsets}\}$
    ▷ Approximation by equation (3)
    $S \leftarrow \rho S + (1-\rho)\nabla C(\mathcal{A}^*)^2$
    $G \leftarrow \dfrac{\sqrt{\Delta}+\zeta}{\sqrt{S}+\zeta} \circ \nabla C(\mathcal{A}^*)$ ▷ Element-wise product
    $\mathcal{A}^* \leftarrow \mathcal{A}^* - G + N(0,1)$
    $\Delta \leftarrow \rho\Delta + (1-\rho)G^2$
    $m \leftarrow \rho m + (1-\rho)\,\mathrm{C}(\mathcal{A}^*)$
  end while
  return $arg\,min_{\mathcal{A}\in\text{search\_path}}\, \mathrm{C}(\mathcal{A}) \cup \{0 : i \notin \text{subsets}\}$
end function

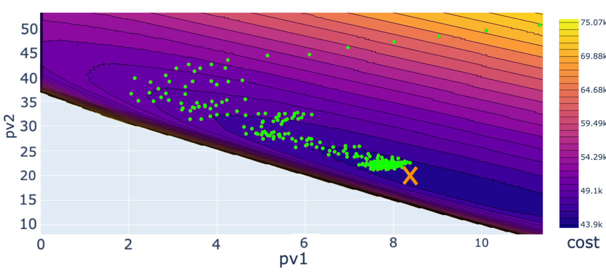

Figure 2: Starting from the top right, the green dots mark the search path for a typical AdaDelta run. The X marks the optimal (min-cost) sizing.

treating the set of points on the Pareto frontier as i.i.d. samples from an unknown distribution. The bound is parameterized by the covariance matrix of the set of sizings, the mean of each dimension, and the desired confidence γ.

We denote each of the η sizing pairs $\mathcal{A}^{(k)}, b^{(k)}$ as an $(N+1)$-dimensional sizing $\xi^{(k)} \triangleq (a_1^{(k)}, a_2^{(k)}, \ldots, a_N^{(k)}, b^{(k)})$. Then, the *unbiased empirical covariance* Σ_η is defined by

$$\Sigma_\eta \triangleq \frac{1}{\eta-1}\sum_{k=1}^{\eta}(\xi^{(k)}-\mu_\eta)(\xi^{(k)}-\mu_\eta)^{\mathrm{T}} \tag{4}$$

where

$$\mu_\eta \triangleq \frac{1}{\eta}\sum_{k=1}^{\eta}\xi^{(k)} \tag{5}$$

is the empirical mean. Note that Σ_η is required to be non-singular (invertible) for subsequent computations. When it is not, it implies that there is no variability in one or more roof i because it is either not used or maximized for all subsamples. We shall remove it from

Algorithm 3 Chebyshev Bound-Finding Algorithm

1: **function** FIND_BOUND($\Lambda^2, \xi^{(1)}, \ldots, \xi^{(k)}$)
2:   $\mu_\eta \leftarrow \frac{1}{\eta}\sum_{k=1}^{\eta}\xi^{(k)}$
3:   $\Sigma_\eta \leftarrow \frac{1}{\eta-1}\sum_{k=1}^{\eta}(\xi^{(k)}-\mu_\eta)(\xi^{(k)}-\mu_\eta)^{\mathrm{T}}$
4:   singular_dimensions ← empty list
5:   **while** Σ_η is singular **do**
6:     Find i such that $\Sigma_{\eta_i} = 0$
7:     singular_dimensions.add(i, μ_{η_i})
8:     Remove i^{th} row and column for Σ_η, remove μ_{η_i}
9:   **end while**
10:   search_queue ← empty queue
11:   **for** $i \in \{1,\ldots,\}\setminus$ singular_dimensions **do**
12:     $\xi^* \leftarrow \mu_\eta$
13:     $\xi^*{}_i \leftarrow arg\,min_{\mu_{\eta_i}\leq\xi_i\leq a_i^{max}} L(\{\mu_{\eta_1},\ldots,\xi_i,\ldots,\mu_{\eta_N}\})$
14:     **if** $\xi^*{}_i$ exists **then**
15:       search_queue.enqueue(ξ^*)
16:     **end if**
17:   **end for**
18:   $\Xi \leftarrow$ empty set
19:   **while** search_queue is not empty **do**
20:     $\xi^* \leftarrow$ search_queue.dequeue()
21:     dir $\leftarrow L(\xi^*) \geq \Lambda^2$? 1 : −1
22:     all_inside? ← true
23:     **for** $i \in \{1,\ldots,\}\setminus$ singular_dimensions **do**
24:       $\xi_{\text{neighbor}} \leftarrow \{\xi_1^*,\ldots,\xi_i^*+\text{dir},\ldots,\xi_N^*\}$
25:       **if** $L(\xi_{\text{neighbor}}) \geq \Lambda^2$ **then**
26:         all_inside? ← false
27:       **end if**
28:       search_queue.enqueue(ξ_{neighbor})
29:     **end for**
30:     **if** $L(\xi^*) \geq \Lambda^2$ and all_inside? **then**
31:       $\Xi \leftarrow \Xi \cup \xi^*$
32:     **end if**
33:   **end while**
34:   **return** $\{\xi \cup \{\mu_{\eta_i} : i \in \text{singular\_dimensions}\} : \xi \in \Xi\}$
35: **end function**

subsequent calculation and use $\mu_{\eta,i}$ as its the final allocation, as implemented in line 4 of Algorithm 3.

The multivariate Chebyshev bound is expressed as an $(N+1)$-dimensional hyper-ellipsoid of sizings Ξ that bounds the γ probability density mass of the empirical distribution. Since we are interested in the upper-bound of each dimension, we also specify that any point in Ξ must be non-dominated by other points inside the hyper-ellipsoid, i.e. if $\xi \in \Xi$, then no sizings inside the hyper-ellipsoid (which may not satisfy our robustness requirement) can be strictly larger than ξ. We denote this in equation (7).

To define Ξ, we first denote

$$L(\xi) \triangleq (\xi-\mu_\eta)^T\Sigma_\eta^{-1}(\xi-\mu_\eta) \tag{6}$$

for any sizing ξ. L represents the distance between ξ and μ_η. Then we have

$$\Xi = \{\xi : L(\xi) = \Lambda^2; \nexists\xi' > \xi, L(\xi') < \Lambda^2.\} \tag{7}$$

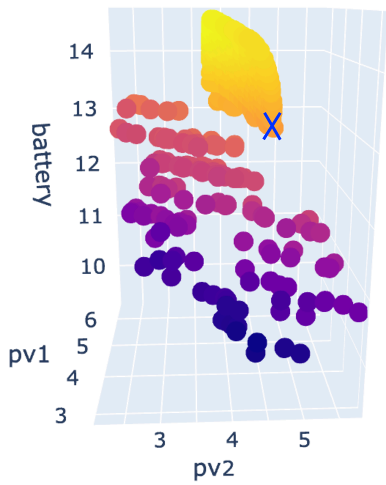

Figure 3: A three-dimensional Chebyshev bound on the upper yellow "dome". Lower points represent 132 least-cost feasible sizings computed over each subsample. Warmer colors represent higher total cost. The X marks the least-cost robust sizing.

where Λ^2 satisfies the following equation:

$$\frac{(N+1)(\eta^2 - 1 + \eta\Lambda^2)}{\eta^2\Lambda^2} = (1-\gamma). \tag{8}$$

Finally, the output of the algorithm (ξ^*) is the cheapest sizing in Ξ,

$$\xi^* = arg\,min_{(\mathcal{A},b)=\xi\in\Xi}\, C(\mathcal{A}, b) \tag{9}$$

Given the equation L of the hyper-ellipsoid, we want to find the boundary through Tabu search, using function L to determine if a point is inside or outside the boundary. We shall start at sizing μ_η and for each dimension i, using binary search to determine the least-cost point outside the boundary as the starting point of the search, as implemented on and after line 10 of Algorithm 3.

We then perform a breadth-first Tabu search starting with these sizings, admitting a point into Ξ if and only if it is outside or on the boundary and all its lower-by-one neighbors are inside of the boundary.

Algorithm 3 describes the computation of the bound on lines 2-17, and the search for the cheapest system on the bound on lines 18-33. In practice, the algorithm is implemented with a hashmap to cache previously searched sizings, as well as boundary limits to prune unrealistic sizings. See Figure 3 for an example of a three-dimensional (two roof segments and one storage), non-dominating partial hyper-ellipsoid Ξ that represents a particular sizing set.

5.5 Computing η, the Number of Samples

We now discuss how to compute η, the number of data sub-samples. Recall from equation (8) that given fixed N, γ, and η, we compute Λ^2 which determines how close the Chebyshev bound is to μ_η. η and Λ^2 are inversely related; the larger η is, the smaller Λ^2. In other words, using more samples can get us a cheaper system with the same robustness guarantee. From Theorem 2 in Reference [23], when $\eta \to \infty$, Λ^2 approaches

$$\Lambda^{2^*} = \frac{N+1}{(1-\gamma)} \tag{10}$$

from above. This is the lower limit of the gap between the center of the ellipse (at μ_η) and the bound.

Since we can obtain many samples from a large enough PV generation and load dataset, and the bound only gets asymptotically tighter with more samples, we need to decide how many samples are enough. We want to find an η that balances the runtime and system cost. The trade-off can be controlled by introducing a parameter $\beta > 0$ and setting $\Lambda^2 = (1+\beta)\Lambda^{2^*}$. For example, if $\beta = 0.1$, we get a Λ^2 that is a factor 1.1 greater than the minimum Λ^{2^*} achieved when $\eta \to \infty$. From equation (8), this reduces to

$$\eta^2\beta - \eta\Lambda^2 + 1 = 0, \tag{11}$$

which forms a quadratic equation that has one real solution greater than 1, giving us the number of samples corresponding to β:

$$\eta = \frac{\Lambda^2 + \sqrt{\Lambda^4 - 4\beta}}{2\beta}. \tag{12}$$

In practice, we find that setting $\beta = 0.1$ gives a reasonable trade-off, requiring 220 simulations for 85% confidence level and 660 for 95% confidence level over two roof segments, and performing within 1% cost of the sizing result for a lower β value such as 0.01, which requires 10 times more computation.

5.6 Putting it Together

Recall that the algorithm consists of three parts: sampling η sub-intervals from a PV generation and load dataset, running simulations and gradient descent to find the minimum cost sizing for each data sample $\xi^{(1)}, \dots, \xi^{(\eta)}$ over each separated convex search space, and computing the Chebyshev bound Ξ. Algorithm 4 provides pseudocode that composes the algorithms described in this section to computer a min-cost robust sizing.

Algorithm 4 Robust Sizing Algorithm

function SIZING($N, n, T, \varepsilon, c_i^f, c_i^m, c^b, a_i^{max}, b^{max}, \mathcal{S}_i, \mathcal{L}$)
  $\Lambda^2 \leftarrow (1+\beta)\dfrac{N+1}{(1-\gamma)}$
  $\eta \leftarrow \dfrac{\Lambda^2 + \sqrt{\Lambda^4 - 4\beta}}{2\beta}$
  $\Xi^* \leftarrow$ empty set
  for subsets $\in P(\{1, \dots, N\}) \setminus \varnothing$ **do**
    ▷ Use power set to separate convex search spaces
    for $i \in 1..\eta$ **do**
      $t \leftarrow i\lfloor \dfrac{n-T}{\eta} \rfloor$ ▷ sample with sliding window
      $\xi^{(k)} \leftarrow$ FIND_SIZING($t, t+T$, subsets)
    end for
    $\Xi \leftarrow$ FIND_BOUND($\Lambda^2, \xi^{(1)}, \dots, \xi^{(\eta)}$)
    $\xi^* \leftarrow arg\,min_{(\mathcal{A},b)\in\Xi}\, C(\mathcal{A}, b)$
    $\Xi^* \leftarrow \Xi^* \cup \xi^*$
  end for
  return $arg\,min_{(\mathcal{A},b)\in\Xi^*}\, C(\mathcal{A}, b)$
end function

6 EVALUATION

We evaluate our multi-roof sizing algorithm using two datasets. The first dataset is extracted from the Pecan Street Dataport [6] and has four years of PV and residential load data. Here, we combine data from multiple homes to synthesize a multi-roof sizing problem. In section 6.1, we present the sizing results using leave-one-year-out cross-validation to show its robustness. In section 6.2, we compare our results with a sizing recommended by a preliminary sizing calculator available on Tesla's website [3].

The second dataset consists of pyranometer and load measurements from the British Antarctic Survey's Rothera station, where solar PV can be deployed on up to five roofs across different buildings. We apply our sizing algorithm and present the results in 6.3.

6.1 Evaluating Robustness on Residential Load

We evaluate the robustness of our algorithm using four years of PV residential load and data measured at 49 homes in Austin, Texas [6]. We run leave-one-year-out experiments, where 3 years of data are used as input to the sizing algorithm, and the final year is used as validation to check whether the sizing met the QoS targets. Given a confidence γ, we expect that the computed sizing will have at most 1-γ fraction of the tests exceed the unmet load target ε.

To create a multi-roof sizing scenario, we use PV generation data from two homes to represent two roof segments, dubbed pv1 and pv2, that have noticeably different PV generation profiles as shown in Figure 4. In addition, each home in the dataset has a distinct load pattern. To compare the sizings according to the shape of each home's load profile rather than its magnitude, we rescale the data so that on average, across four years, each solar panel generates 0.2 kW and the mean load is 2 kW. We assume that each home can install up to 60 panels (12 kW) on each roof segment, and up to 120 kWh of storage. We also assume a fixed cost of \$2000 for each roof segment, a variable cost of \$2000/kW [19], and a battery cost of \$500/kWh, similar to the current cost of a Powerwall [2].

We evaluate our algorithm for two QoS targets, $\varepsilon = 0.1$ and 0.5. The first target represents a scenario of near-total grid independence, while the second represents a more fiscally-prudent scenario where solar PV primarily meets loads during the day and the grid is used to meet load at night. We also evaluate two confidence levels, $\gamma = 0.85$ and 0.95, with the former level expected to produce a cheaper albeit less robust sizing.

Figure 5 shows the aggregate results of 196 tests consisting of 4 leave-one-year-out experiments across 49 homes. As the target validation loss increases from 10% to 50%, the distribution visibly shifts to the right, as expected. Moreover, when the target loss is 10% ($\varepsilon = 0.1$), only 4.1% and 1% of the tests exceeded the loss at 85% and 95% confidence levels respectively. Similar results are seen with a loss target of 50% ($\varepsilon = 0.5$), with 5.1% and 2% of tests exceeding the loss target at 85% and 95% confidence level respectively. These results empirically demonstrate that the sizings computed by our algorithm are feasible and robust. They are also reasonably tight, as seen by the increase in the density of the loss distribution left of the loss target indicated by the red line in each figure.

Figure 6 shows the average sizing results across all homes and years given different unmet load target and confidence level combinations. Notably, pv2 is slightly more favored than pv1. This can be

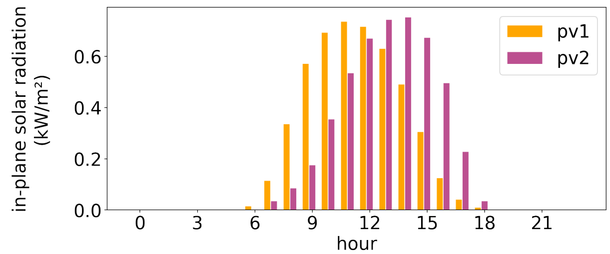

Figure 4: Average hourly solar generation for two roof segments in the Pecan Street dataset. Note that pv1 peaks in the morning hours and pv2 peaks in the afternoon hours.

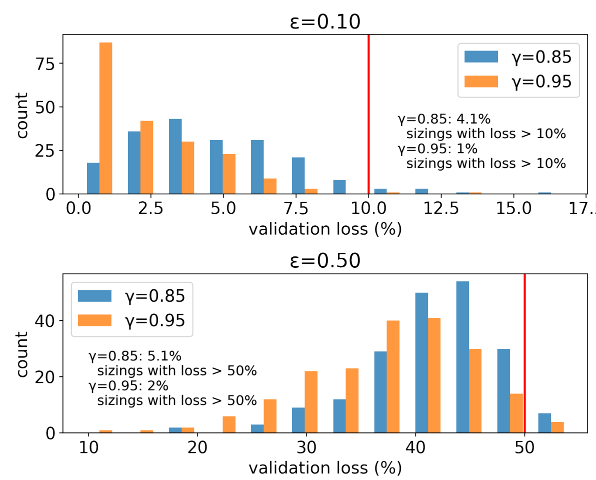

Figure 5: Aggregated leave-one-year-out test results on 196 test across 49 houses, showing % of tests that land outside the unmet load target for $\varepsilon = 10\%$ and 50%, with the red vertical line indicating the unmet load target.

explained by an observation that the majority of houses have load peaks in the evening and storage costs are reduced when more panels are allocated to pv2. Also, there is a 4-5× increase in battery and 2-3× increase in PV when ε goes from 0.5 to 0.1. This is because meeting the more stringent loss target requires the system to generate and store PV generation by day for use at night.

We also found that the sizing for the majority of the homes includes panels on both roofs, despite each having enough space to hold all the allocated panels. We hypothesize that the differing peaking times for the two roof segments cause PV generation to be more spread out over the course of the day, which reduces the need for storage. To confirm this, we studied two load patterns shown in Figure 7. The first house has a more pronounced evening load peak, which better matches the generation profile from pv2. Indeed, for this home, our algorithm suggests a sizing that uses only pv2. In contrast, the sizing for the second home uses both roof segments. This confirms our intuition that the optimal sizing attempts to match PV generation profiles to load profiles.

To summarize, our experiments using the Pecan Street Dataport dataset confirm that our algorithm produces feasible sizings that are robust to variations in the solar and load profiles.

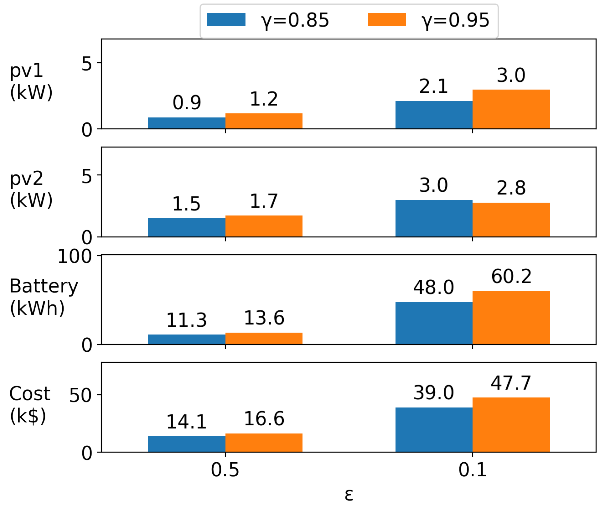

Figure 6: Average sizing results across all houses and years, including PV allocation for the two roof segments, battery amount and total system cost.

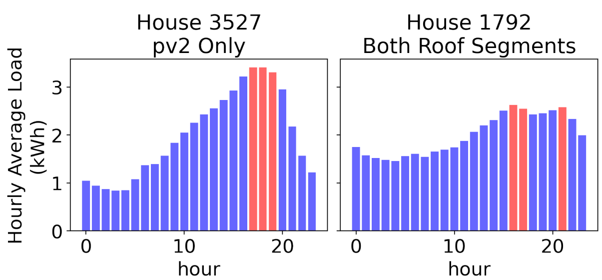

Figure 7: Two typical residential load patterns. House 3527 shows load more concentrated in the evening and its sizing contains only pv2. House 1792's load is more evenly distributed and its sizing contains both roof segments.

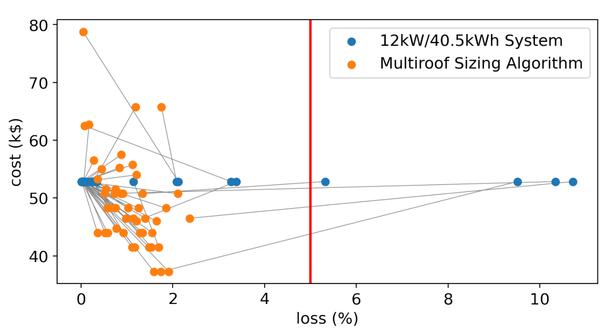

Figure 8: Comparison of cost and loss for a system sized by our algorithm (orange) and the Tesla's algorithm (blue) across 49 houses; results for a given house are connected by a line. The red vertical line indicates the unmet load target used in our algorithm.

6.2 Comparison with Tesla's Sizing Algorithm

We now compare our sizing algorithm to the one on Tesla's website that is used to generate a preliminary sizing estimate[2] [3]. We caution that this is not an "apples-to-apples" comparison because their online calculator does not explicitly consider multiple roof segments, or incorporate a loss objective or a confidence criterion, as in our work; rather, this comparison serves to illustrate how our sizing approach compares to a widely-used alternative.

The Tesla calculator's goal is to size a system that can generate enough power to meet the home's net load and a battery that can power the home for approximately one day. It uses two parameters: location and monthly bill. Our normalized load traces correspond to a 1440 kWh monthly load, which leads to a $165 monthly bill under Austin Energy's residential pricing scheme [5]. Using Tesla's algorithm, the sizing computed is 12.24 kW solar panels and 3 Powerwalls, equivalent to 40.5 kWh of storage, at a total cost of $52,850. Note that this assumes a single rooftop. Hence, to compare it with our algorithm, we select the same two roof segments as before and try three allocations: all on pv1, all on pv2, and half on each roof. We take the allocation with the lowest validated loss to compare with the results of the same house as computed by our algorithm.

We run our sizing algorithm with a target of 5% unmet load over any 365-day period, a confidence interval of 75%, and a search precision comparable to the calculator: 1 kW for PV, and 13.5 kWh (one Powerwall) for storage. We evaluate the loss from the two sizings using simulations, Specifically, we sample 50 sub-intervals, 1-year of length, over 4 years of validation data. We then compute unmet loads for each subinterval and report the median loss across all windows.

The sizing comparison is summarized in Figure 8. On average, the sizings generated by our algorithm require fewer PV panels and a larger battery compared to the Tesla algorithm, at 6.8 kW and 60.5 kWh respectively. The average cost is $49,370, a 6.5% decrease, and all houses have under 5% median validated loss.

A closer look shows that our algorithm suggests cheaper systems for houses that are oversized by Tesla. 39 out of 49 houses have ≤0.5% loss using the Tesla algorithm (this is the blue cluster on the far left). Of these, 18 houses also have ≤0.5% loss even when using a smaller 8 kW/27 kWh system (33% smaller PV and battery), indicating oversizing. With our algorithm, these 18 houses have a higher loss, but still under 5%, and also have an average cost of $47,639, an 8.4% saving.

Occasionally, our algorithm suggests a more expensive sizing than the Tesla algorithm (orange points to the right and above the corresponding blue point). In these cases, we found a large variance in load profiles across the years, which results in a more conservative sizing due to our use of the Chebyshev bound.

Finally, our algorithm found a better sizing for houses that are under-sized by the Tesla algorithm. For such houses, which have a high loss (>5%) with the Tesla algorithm, our algorithm found a smaller sizing and thus a cheaper system, while still meeting the QoS target. This difference reflects the benefits of considering load profiles rather than only monthly bills.

[2] This preliminary estimate is later updated to generate a final sizing by a consultant.

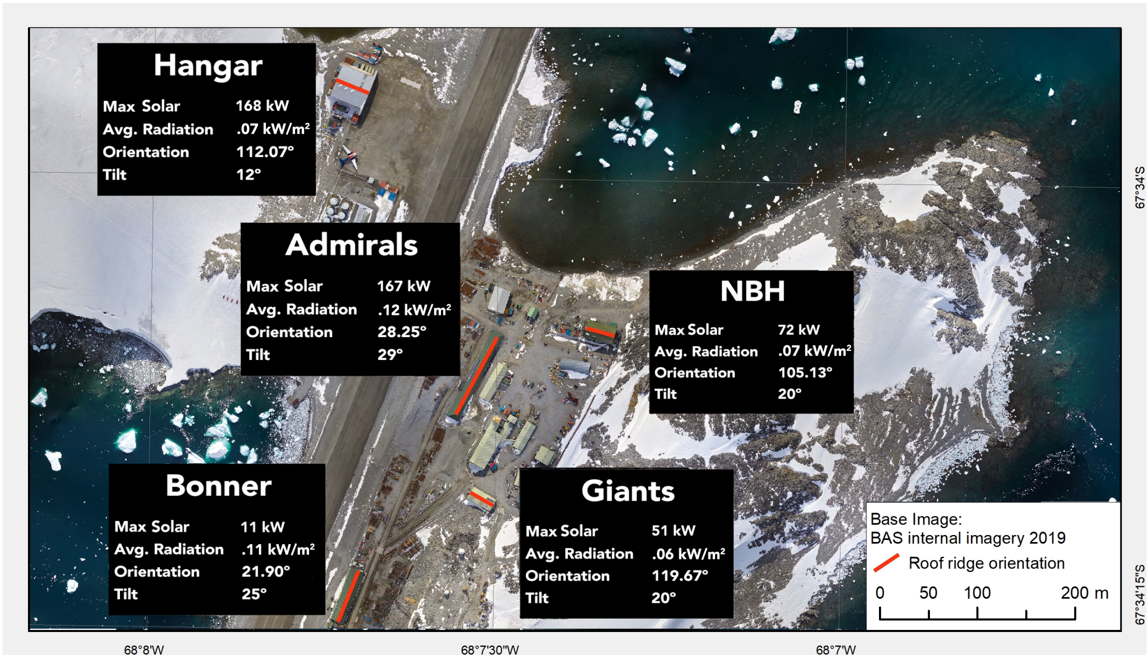

Figure 9: Satellite picture of the Rothera research station, with buildings annotated with name, solar power that it can support, average PV radiation, orientation angle from true north, and tilt angle to the ground plane. Image used by permission of the BAS.

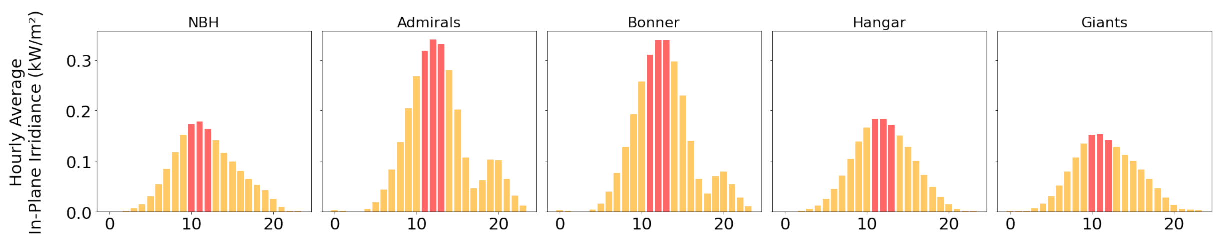

Figure 10: Solar patterns averaged by hour for the five roof segments in the Rothera Station, with the three peak hours emphasized in red. Note that the Admirals and Bonner buildings receive more sunlight, on average, compared to the others.

6.3 Decarbonizing the BAS Rothera Station in Antarctica

We have applied our sizing algorithm to the use case of partly decarbonizing the British Antarctic Survey's (BAS) Rothera research station, based on real traces. The average load of the station is 95 kW, with a peak of 130 kW. So far, the station has been powered by diesel power generators that use 60 m^3 of fuel per month on average, which is carbon-intensive and expensive to deliver to such a remote location. Through its current modernization program, the BAS aims to decarbonize the station by 2030. We study how the station might hypothetically install a solar+storage system to contribute to decarbonization.

The BAS provided us with six years of hourly horizontal solar irradiance as measured through a pyranometer, and one year of representative hourly load data from 2015, which was before the initiation of the modernization program. At present, five buildings in the station have been identified as suitable for solar PV installation, as shown in Figure 9. We therefore use the irradiance trace to compute five separate PV generation traces, according to the tilt and orientation of each potentially suitable roof segment as suggested in Reference [12]; the resulting set of daily average PV generation profiles is shown in Figure 10. Note that the Admirals and Bonner have higher solar generation potential than the other three and are therefore the best candidates for PV panels. However, the roof segment on Bonner is relatively small. Three other roof segments are less desirable, with Hangar having a slightly larger average radiation and the largest available area. Collectively, the five roof segments can support up to 450 kWp of PV generation. Note the two peaks in generation on the Admirals and Bonner roofs, due to their northeast-facing orientation and the extended day during the Antarctic summer. In contrast, the other roofs have a southeast-facing orientation and single, lower generation peaks.

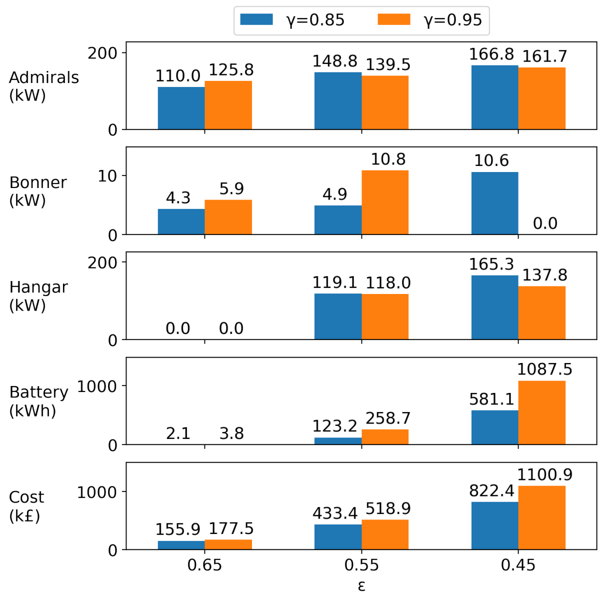

Figure 11: Results of sizing for different confidence levels and target unmet load. Note that allocation for less-efficient roof segments NBH and Giants are 0 across all tested loss/confidence combinations.

We also trained an ARMA model on the load data to create synthetic load traces spanning six years, using the process described in Reference [24].

The costs used in this study are a £5000 fixed cost for equipping each roof, £1.18/Wp marginal PV cost (equivalent to £188 per m^2 panel which is assumed to be 160 Wp), and £670/kWh for batteries. This data was used to compute the sizing for several system performance targets, with ε ranging from 0.35 (meeting 65% of load) to 0.65 (meeting 35% of load), and confidence level γ at 0.85 and 0.95.

The results of the sizing are summarized in Figure 11, which includes the per-roof segment panel allocation, battery size, and total cost. No feasible sizing existed for $\varepsilon = 0.35$, due to the fact that very little sunlight is seen in Antarctica between May and August. Other ε targets resulted in varied systems sizings: with $\varepsilon = 0.65$, the sizing requires very little battery capacity and most of the solar panels are on a single roof segment, while for $\varepsilon = 0.45$ the sizing was split more equally across two of the roof segments and required roughly 500-1000 kWh of storage to meet night-time load, depending on the desired confidence level. Notably, for $\varepsilon = 0.45$ and $\gamma = 0.95$, the preferred roof segments are not always those which receive the most radiation (Admirals and Bonner), but a combination of those that spread out PV generation over the course of a full day (Admirals and Hangar).

Note that because we only had access to only one year of load data, we were unable to carry out a leave-one-out analysis to evaluate the robustness of our sizing.

7 DISCUSSION AND CONCLUSION

In this work, we present an algorithm for choosing the number of solar panels and the amount of storage needed to meet a certain fraction of the load in a microgrid setting. Unlike prior work, which assumes that the desired number of panels can always be accommodated on a rooftop, we take into account the pragmatic issue that roofs typically incorporate multiple roof segments, each with its own panel capacity, tilt, and orientation. This unexpectedly leads to a much more complex sizing problem. Our solution, which is based on stochastic gradient descent, allows us to compute sizings despite the non-linear nature of the problem. We demonstrate the robustness of our approach using a leave-one-out analysis and the Pecan Street dataset. We also use our approach to compute the sizing needed for different levels of decarbonization of the British Antarctic Survey station in Rothera.

Our approach deals with the large search space by using stochastic gradient descent to find least-cost, feasible sizings over repeated trials and the computation of a Chebyshev bound over the results of these trials. This approach has some limitations, as discussed next.

First, when using solar and load traces from multiple years, one of the years may have atypically low PV generation or high load. A sizing computed using an atypical year tends to be more conservative, as is a Chebyshev bound that includes this sizing. Unfortunately, there does not appear to be a systematic approach to identify anomalous traces in a fairly small set of traces.

Second, to make our search more efficient, we partition the non-linear search space into a set of convex sub-spaces. This is reasonable for a small number of roof segments (for example, in Rothera, we used this approach with 5 roof segments). However, this approach does not scale well with the number of roof segments due to a combinatorial explosion.

Third, we compute the Chebyshev bound using a multidimensional breath-first Tabu search. This turns out to be memory intensive, especially when searching a large N-dimensional space where each point has N neighbours.

Fourth, a Chebyshev bound sometimes leads to unintuitive results, an example being the recommendation of a handful of panels on the Bonner building in Rothera. This is because the Bonner building is part of a least-cost sizing for some runs of the stochastic gradient descent algorithm and not others. When computing a Chebyshev bound, however, this results in a small number of panels being allocated to this building, a counter-intuitive result.

Finally, on an unrelated matter, we note that our sizing approach assumes a microgrid setting, that is, with no payments for over-generation from the grid using net-metering or feed-in-tariff. If such payments are introduced, then the sizing problem becomes far more complex, since the storage operation policy depends in detail upon the nature of the grid payment scheme [17]. We defer analysis of this more complex scenario, as well as to overcome the limitations listed above, to future work.

8 ACKNOWLEDGEMENTS

We would like to thank the Pecan Street Dataport for access to their solar and load traces. Thanks also to the British Antarctic Survey for sharing their load and solar data sets, as well as the building orientation and size parameters, from Rothera. Thanks to Prof. Madhavapeddy at Cambridge for access to computing resources. Finally, thanks to Jackson Taylor for his roof.

REFERENCES

[1] Commercial and residential hourly load profiles for all tmy3 locations in the united states. https://openei.org/doe-opendata/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states. Accessed: 2021-02-15.

[2] Tesla powerwall design. https://www.tesla.com/powerwall/design. Accessed: 2021-02-15.

[3] Tesla solar panel sizing and design. https://www.tesla.com/support/energy/solar-panels/going-solar/sizing-and-design. Accessed: 2021-02-15.

[4] U.S. Energy Information Administration, Smart Meter Installations. https://www.eia.gov/tools/faqs/faq.php?id=108&t=3. Accessed: 2021-02-15.

[5] Austin energy residential rates. https://austinenergy.com/ae/rates/residential-rates/residential-electric-rates-and-line-items, Jan 2020. Accessed: 2021-05-19.

[6] Pecan street dataport. https://dataport.cloud/, 2021.

[7] Pvwatts calculator. https://pvwatts.nrel.gov/, 2021. Accessed: 2021-02-15.

[8] Reopt lite. https://reopt.nrel.gov/tool, 2021.

[9] Clairand, J.-M., Arriaga, M., Cañizares, C. A., and Álvarez-Bel, C. Power generation planning of galapagos' microgrid considering electric vehicles and induction stoves. *IEEE Transactions on Sustainable Energy 10*, 4 (2018), 1916–1926.

[10] Das, I., and Cañizares, C. A. Renewable energy integration in diesel-based microgrids at the canadian arctic. *Proceedings of the IEEE 107*, 9 (2019), 1838–1856.

[11] Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* (2011), 2121–2159.

[12] Gao, X. Should you clean your solar panels now? Master's thesis, 2014. Accessed: 2021-02-15.

[13] Hafez, O., and Bhattacharya, K. Optimal planning and design of a renewable energy based supply system for microgrids. *Renewable Energy 45* (2012), 7–15.

[14] Hidalgo-León, R., Siguenza, D., Sanchez, C., León, J., Jácome-Ruiz, P., Wu, J., and Ortiz, D. A survey of battery energy storage system (bess), applications and environmental impacts in power systems. In *2017 ieee second ecuador technical chapters meeting (etcm)* (2017), IEEE, pp. 1–6.

[15] Kazhamiaka, F., Ghiassi-Farrokhfal, Y., Keshav, S., and Rosenberg, C. Robust and practical approaches for solar pv and storage sizing. In *Proceedings of the Ninth International Conference on Future Energy Systems* (2018), pp. 146–156.

[16] Kazhamiaka, F., Ghiassi-Farrokhfal, Y., Keshav, S., and Rosenberg, C. Comparison of different approaches for solar pv and storage sizing. *IEEE Transactions on Sustainable Computing* (2019), 1–1.

[17] Kazhamiaka, F., Rosenberg, C., and Keshav, S. Practical strategies for storage operation in energy systems: design and evaluation. *IEEE Transactions on Sustainable Energy 7*, 4 (2016), 1602–1610.

[18] Krah, K. Behind-the-meter solar+ storage modeling tool comparison. Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2019.

[19] Matasci, S. How much do solar panels cost? 2021 guide: Energysage. https://news.energysage.com/how-much-does-the-average-solar-panel-installation-cost-in-the-u-s/, Feb 2021.

[20] Mishra, S., Pohl, J., Laws, N., Cutler, D., Kwasnik, T., Becker, W., Zolan, A., Anderson, K., Olis, D., and Elgqvist, E. Computational framework for behind-the-meter der techno-economic modeling and optimization: Reopt lite. *Energy Systems* (2021), 1–29.

[21] Orlando, M., Bottaccioli, L., Patti, E., Macii, E., Vinco, S., and Poncino, M. Optimal configuration and placement of pv systems in building roofs with cost analysis. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)* (2020), IEEE, pp. 1411–1416.

[22] Ru, Y., Kleissl, J., and Martinez, S. Storage size determination for grid-connected photovoltaic systems. *IEEE Transactions on Sustainable Energy 4*, 1 (Jan 2013), 68 – 81.

[23] Stellato, B., Van Parys, B. P., and Goulart, P. J. Multivariate Chebyshev inequality with estimated mean and variance. *The American Statistician 71*, 2 (2017), 123–127.

[24] Sun, S., Kazhamiaka, F., Keshav, S., and Rosenberg, C. Using synthetic traces for robust energy system sizing. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems* (New York, NY, USA, 2019), e-Energy '19, Association for Computing Machinery, p. 251–262.

[25] Tieleman, T., and Hinton, G. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural Networks for Machine Learning* (2012).

[26] Zeiler, M. D. Adadelta: An adaptive learning rate method. *CoRR abs/1212.5701* (2012).