

Technical report 19-004

Efficient convex optimization for optimal PMU placement in large distribution grids*

M. Picallo, A. Anta, and B. De Schutter

If you want to cite this report, please use the following reference instead:

M. Picallo, A. Anta, and B. De Schutter, "Efficient convex optimization for optimal PMU placement in large distribution grids," *Proceedings of the 13th IEEE PowerTech 2019*, Milan, Italy, 6 pp., June 2019. doi:10.1109/PTC.2019.8810467

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: <https://www.dcsc.tudelft.nl>

*This report can also be downloaded via https://pub.deschutter.info/abs/19_004.html

Efficient Convex Optimization for Optimal PMU Placement in Large Distribution Grids

Miguel Picallo*, Adolfo Anta[†], Bart De Schutter*

*Delft Center for Systems and Control, Delft University of Technology
Delft, The Netherlands

{m.picallocruz,b.deschutter}@tudelft.nl

[†]Austrian Institute of Technology

Vienna, Austria

Adolfo.Anta@ait.ac.at

Abstract—The small amount of measurements in distribution grids makes their monitoring difficult. Topological observability may not be possible, and thus, pseudo-measurements are needed to perform state estimation, which is required to control elements such as distributed generation or transformers at distribution grids. Therefore, we consider the problem of optimal sensor placement to improve the state estimation accuracy in large-scale, 3-phase coupled, unbalanced distribution grids. This is an NP-hard optimization problem whose optimal solution is unpractical to obtain for large networks. For that reason, we develop a computationally efficient convex optimization algorithm to compute a lower bound on the possible value of the optimal solution, and thus check the gap between the bound and heuristic solutions. We test the method on a large test feeder, the standard IEEE 8500-node, to show the effectiveness of the approach.

Index Terms—optimal sensor placement, phasor measurement units, distribution grid state estimation, projected gradient descent, optimal design of experiments

I. INTRODUCTION

Accurate monitoring of voltages, currents, and loads is essential to manage an electrical power network. In this context, State Estimation (SE) consists in estimating the network state, represented typically by the bus voltage phasors. Normally, SE computes the state that best concurs with the available measurements by solving a weighted least-squares problem using an iterative approach like Newton-Raphson [1], [2].

Network monitoring can be improved with the introduction of Phasor Measurements Units (PMUs). Some previous work proposes to achieve topological observability [3] by using integer linear programming [4]. However, although cheap PMUs are becoming available [5], their operational and network communication costs [6] may still prevent installing enough units to guarantee topological observability and thus also numerical observability, which is necessary for SE [3]. Therefore, PMUs need to be combined with Supervisory Control And Data Acquisition (SCADA) measurements to solve the SE problem [7]. Recent work [8] presents the problem of optimal PMU placement as the minimization of the error covariance matrix in SE through some metric from optimal design of experiments [9]. This transforms the optimal PMU placement problem into

a combinatorial optimization problem with binary variables and a nonlinear objective function. As a consequence, as the network becomes larger, the optimal solution cannot be computed within a reasonable time span, due to the number of possible combinations of measurements.

The problem is even more complex in distribution grids, where even when SCADA measurements are used, the number of measurements is not enough to achieve observability. As a consequence, SE for distribution grids needs to rely on so-called pseudo-measurements, like load forecasts, which may have a high uncertainty associated to them (approx. 50% relative error [10]). This motivates the use of PMUs in distribution grids [11] to increase the SE accuracy, by for example using greedy and random combinatorial approaches of sensors [12], [13] or evolutionary algorithms [14], [15]. However, these methods do not provide optimality guarantees, nor do they inform how good their suboptimal solutions are with respect to the optimal one. A convexity-based lower bound of the value of the optimal solution can provide this information [8], but this is difficult to compute for large grids.

Our main contribution is to propose a computationally efficient and scalable approach to derive this convexity-based lower bound for the optimal solution of the PMU placement problem under a budget constraint in large distribution grids. In particular, we extend the projection algorithm in [16] to the budget-constrained case by using the Karush-Kuhn-Tucker conditions. Then, we derive subgradient expressions for the A,D,E,M-optimal metrics [9] and combine them with the projection algorithm to create a projected subgradient descent method to solve the convex relaxed optimal PMU placement problem in large distribution grids.

The rest of the paper is structured as follows: Section II presents some background about power networks. Section III discusses the different types of measurements. Section IV summarizes the newly proposed methodology for SE [17]. Section V presents the metrics considered and states the problem of optimal sensor placement problem under a budget constraint. Section VI shows the optimization algorithms used to solve the problem. Section VII shows the results on a test case of a large distribution network. Finally, Section VIII presents the conclusions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675318 (INCITE).

II. DISTRIBUTION GRID MODEL

A distribution grid consists of buses, where power is injected or consumed, and branches, each connecting two buses. This system can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with nodes $\mathcal{V} = \{1, \dots, N_{\text{bus}}\}$ representing the buses, edges $\mathcal{E} = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$ representing the branches, and edge weights $\mathcal{W} = \{w_{i,j} \mid (v_i, v_j) \in \mathcal{E}\}$ representing the admittance of the branches, which are determined by the length and type of the line cables.

In 3-phase networks buses may have up to 3 phases, so that the voltage at bus i is $V_{\text{bus},i} \in \mathbb{C}^{n_{\phi,i}}$, where $n_{\phi,i} \leq 3$ (and the edge weights $w_{i,j} \in \mathbb{C}^{n_{\phi,i} \times n_{\phi,j}}$). The state of the network is then typically represented by the vector bus voltages $V_{\text{bus}} = [V_{\text{src}}^T, V^T]^T \in \mathbb{C}^{N+3}$, where $V_{\text{src}} \in \mathbb{C}^3$ denotes the known voltage of the 3 phases at the source bus, and $V \in \mathbb{C}^N$ the voltages in the non-source buses, where N depends on the number of buses and phases per bus.

Using the Laplacian matrix $Y \in \mathbb{C}^{(N+3) \times (N+3)}$ of the weighted graph \mathcal{G} , called admittance matrix [1], the power flow equations to compute the currents I and the power loads S are:

$$\begin{bmatrix} I_{\text{src}} \\ I \end{bmatrix} = Y \begin{bmatrix} V_{\text{src}} \\ V \end{bmatrix}, \quad S = \text{diag}(\bar{I})V \quad (1)$$

where $\bar{(\cdot)}$ denotes the complex conjugate, $\text{diag}(\cdot)$ represents the diagonal operator, converting a vector into a diagonal matrix.

III. MEASUREMENTS

As explained in [17], several different sources of information can be available to solve the SE problem:

- 1) *Pseudo-measurements*, i.e., load estimations S_{psd} based on predictions and/or known installed load capacity at every bus. Since these pseudo-measurements are estimations, they are modeled as noisy measurements with a Gaussian distribution with a relative large standard deviation (a typical value can be $\sigma_{\text{psd}} \approx 50\%$ [10]).
- 2) *Virtual measurements*, i.e., buses with zero-injections, no loads connected. They can be modeled as physical constraints for the voltage states by defining the set of indices of zero-injection buses $\varepsilon = \{i, \dots, j\}$:

$$(S)_{\varepsilon} = 0, \quad (I)_{\varepsilon} = 0 \quad (2)$$

where $(\cdot)_{\varepsilon}$ denotes the elements at indices in ε .

- 3) *Real-time PMU measurements*, i.e., voltage and current GPS-synchronized measurements of magnitude and phase angle. According to the IEEE standard for PMU [18], we model the noises with a low standard deviation for the magnitude and the angle, $\sigma_{\text{mag}} \approx 1\%$ and $\sigma_{\text{ang}} \approx 0.01$ rad respectively. They can be expressed using a linear approximation [17] with magnitude and angle noise due to the measurements and imperfect synchronization. For a number of N_{meas} measurements $z_{\text{meas}} \in \mathbb{C}^{N_{\text{meas}}}$ we have:

$$z_{\text{meas}} \approx C_{\text{meas}}V + \text{diag}(C_{\text{meas}}V)(\omega_{\text{mag}} + j\omega_{\text{ang}}) \quad (3)$$

where $\omega_{\text{mag}} \sim \mathcal{N}(0, \sigma_{\text{mag}}I_{d, N_{\text{meas}}})$, $\omega_{\text{ang}} \sim \mathcal{N}(0, \sigma_{\text{ang}}I_{d, N_{\text{meas}}})$, with $I_{d,n}$ denoting the identity matrix of dimension n . C_{meas}

is the matrix mapping state voltages to measurements. Then, for measurement j at phase l of bus i we have:

$$(C_{\text{meas}}V)_j = (C_{\text{meas}})_{j,\bullet}V = \begin{cases} V_{i_l} & \text{for a voltage measurement} \\ (Y)_{i_l,\bullet}V & \text{for a current measurement} \\ (Y)_{i_l, m_l}(V_{i_l} - V_{m_l}) & \text{for a branch-current } i \rightarrow m \text{ measurement} \end{cases} \quad (4)$$

where $(\cdot)_{j,\bullet}$ denotes row j . Since the measurement noises in (3) are small according to the PMUs standard [18], their covariance matrices can be approximated using the measurements:

$$\begin{aligned} \Sigma_{\text{meas}} &= (\sigma_{\text{mag}}^2 + \sigma_{\text{ang}}^2)\text{diag}(|C_{\text{meas}}V|^2) \\ &\approx (\sigma_{\text{mag}}^2 + \sigma_{\text{ang}}^2)\text{diag}(|z_{\text{meas}}|^2) \end{aligned}$$

IV. STATE ESTIMATION

Typically, SE consists in finding the voltages that best match the measurements by solving a weighted least-squares problem [1]: $\min_V \left\| [z_{\text{meas}}^T, S_{\text{psd}}^T]^T - h(V) \right\|_W^2$, where $h(\cdot)$ is a measurement function and $\|x\|_W^2 = x^T W x$ for a weight matrix $W \succ 0$. As proposed in [17], SE can be decomposed in two parts: First, using the pseudo-measurement S_{psd} , we solve the power flow to obtain a prior estimate V_{prior} :

$$V_{\text{prior}} = \text{PowerFlow}(S_{\text{psd}}) \quad (5)$$

Then, using the real-time PMU measurements z_{meas} , a posterior solution V_{post} can be derived using a linear filter:

$$V_{\text{post}} = V_{\text{prior}} + K(z_{\text{meas}} - C_{\text{meas}}V_{\text{prior}}) \quad (6)$$

where the gain matrix K is obtained by minimizing the error covariance $\text{tr}(\Sigma_{\text{post}}) = \text{tr}(\mathbb{E}[(V_{\text{post}} - V)^H(V_{\text{post}} - V)])$, with $(\cdot)^H$ denoting conjugate transpose:

$$K = \Sigma_{\text{prior}} C_{\text{meas}}^H (C_{\text{meas}} \Sigma_{\text{prior}} C_{\text{meas}}^H + \Sigma_{\text{meas}})^{-1} \quad (7)$$

where Σ_{prior} and Σ_{meas} are the expected error covariance of the prior estimate V_{prior} and the measurements z_{meas} respectively.

Remark 1. *Extra measurements from non-synchronized real-time sensors, like magnitude measurements from SCADA, can be included in the posterior update (6) for a greater improvement of the posterior estimate V_{post} by using the first-order approximation of the measurement function [17].*

Remark 2. *As shown in [7], splitting the problem in two steps yields the same first-order approximation as solving the problem in one step. Moreover, the posterior minimum-variance estimator using the linear update (6), is equal to the maximum-likelihood using a weighted least-squares approach [17]. Therefore, we can conclude that for an SE method that assumes Gaussian noises and performs a maximum likelihood estimation, the posterior covariance will be the same Σ_{post} , and thus the method developed here for optimal sensor placement can be also extended for other SE techniques satisfying these conditions.*

Since V_{post} in (6) is an unbiased estimator, SE accuracy can be defined as a function of the posterior covariance matrix Σ_{post} , which needs to be minimized to improve the

SE accuracy. After some manipulations, the error covariance Σ_{post} for the posterior estimation V_{post} can be expressed as:

$$\Sigma_{\text{post}} = (\Sigma_{\text{prior}}^{-1} + C_{\text{meas}}^H \Sigma_{\text{meas}}^{-1} C_{\text{meas}})^{-1} \quad (8)$$

Since measurement errors are caused separately by each sensor, they are independent, i.e. Σ_{meas} is diagonal, and we can split Σ_{post} by every measure:

$$\begin{aligned} \Sigma_{\text{post}} &= (\Sigma_{\text{prior}}^{-1} + \sum_i (C_{\text{meas}})_{i,\bullet}^H (C_{\text{meas}})_{i,\bullet} (\Sigma_{\text{meas}}^{-1})_{i,i})^{-1} \\ &= (\Sigma_{\text{prior}}^{-1} + \sum_i x_i (\tilde{C}_{\text{meas}})_{i,\bullet}^H (\tilde{C}_{\text{meas}})_{i,\bullet} (\Sigma_{\text{meas}}^{-1})_{i,i})^{-1} \end{aligned} \quad (9)$$

so that Σ_{post} is a function of x , where $x_i \in \{0, 1\}$, $x_i = 1$ if the physical quantity i has a sensor measuring its value, 0 otherwise; and \tilde{C}_{meas} is the special case of C_{meas} with all possible measurements of all types (bus voltage, bus current, and line current) for all nodes and lines in each phase.

V. OPTIMAL SENSOR PLACEMENT PROBLEMS

In order to improve the accuracy of the SE, the problem of optimal sensor placement consists in minimizing $\Sigma_{\text{post}}(x)$ in (9), according to a metric $m(\cdot)$ and under a set of constraints $h(\cdot)$ to limit the number of sensors or the total cost:

Definition 1. Optimal Sensor Placement problem:

$$\min_x m(\Sigma_{\text{post}}(x)) \text{ s.t. } h(x) \leq 0, x_i \in \{0, 1\} \forall i \quad (10)$$

For simplicity, we define: $f(x) \equiv m(\Sigma_{\text{post}}(x))$.

A. Metrics for Sensor Placement

There are many possible metrics $f(x)$ available in the context of optimal design of experiments [9]:

- **A-optimal:** $f_A(x) = \text{tr}(\Sigma_{\text{post}}(x))$. This corresponds to minimizing the sum of the eigenvalues of Σ_{post} and thus the sum of the lengths of the axes of the confidence ellipsoid [19]. This is the metric typically used for SE methods, since standard SE maximizes the log-likelihood through a weighted least-squares minimization [1], which is equivalent to the minimum variance estimator using the trace [17].
- **D-optimal:** $f_D(x) = \log \det(\Sigma_{\text{post}}(x))$, where $\det(\cdot)$ is the determinant. This corresponds to minimizing the logarithm of the product of the eigenvalues of Σ_{post} and is related to the logarithm of the volume of the confidence ellipsoid [19].
- **E-optimal:** $f_E(x) = \lambda_{\max}(\Sigma_{\text{post}}(x))$, where $\lambda_{\max}(\cdot)$ is the maximum eigenvalue. This corresponds to minimizing the length of the longest axis of the confidence ellipsoid [19].
- **M-optimal:** $f_M(x) = \max_i (\Sigma_{\text{post}}(x))_{i,i}$, which corresponds to minimizing the highest diagonal term of $\Sigma_{\text{post}}(x)$.

When relaxing x_i in (10) to be continuous, i.e. $x_i \in [0, 1]$, the metrics are all convex in x [19, Section 7.5] [8].

B. Solutions under a budget constraint

So far we have looked at the cost function in (10); now we will focus on the constraints: we will consider a budget constraint limiting the total cost of the deployed sensors, which may have a different cost. This constraint allows to take into account the extra cost of installing a sensor in a remote area, or in locations where specific rights might be required, or different types of sensors, etc. This is a more realistic and general approach than using a cardinality constraint, which is the most typical approach in the literature [8], [10], [20], and

which corresponds to the particular case where sensors have equal costs. The optimal sensor placement problem under a budget constraint can then be expressed as:

$$x_{\text{opt}} = \arg \min_x f(x) \text{ s.t. } \sum_i c_i x_i \leq b, x_i \in \{0, 1\} \forall i \quad (11)$$

where b represents the budget and c_i the cost of installing a sensor at location i . As mentioned before, computing this optimum is unpractical, and instead suboptimal solutions are typically developed. Therefore, convexity-based bounds are helpful to evaluate the performance of such suboptimal solutions. The relaxed convex problem would be:

$$x_{\text{convex}} = \arg \min_x f(x) \text{ s.t. } \sum_i c_i x_i \leq b, x_i \in [0, 1] \forall i \quad (12)$$

We denote the value corresponding to (12) as $f(x_{\text{convex}}) = f_{\text{convex}}$. However, x_{convex} will not necessarily be feasible to (11). A feasible solution x_{feas} (with respective value $f(x_{\text{feas}}) = f_{\text{feas}}$) can be built using the convex solution x_{convex} of (12), by selecting the sensors corresponding to the elements of x_{convex} with the highest values. This can be done iteratively until the budget is filled, i.e. $\mathcal{B} = \{i \mid c_i \leq b - \sum_j c_j x_j^{(K-1)}\} = \emptyset$. At every iteration K , we would add a new sensor i^* such that:

$$i^* = \arg \max_{i \in \mathcal{B} \cup \{i \mid x_i^{(K-1)} = 0\}} x_{\text{convex},i}, x^{(K)} = x^{(K-1)} + e^{i^*} \quad (13)$$

where e^i is the vector of the natural basis with $e_i^i = 1$, $e_{j \neq i}^i = 0$, and $x^{(K)}$ denotes the value of x at iteration K and $x_i^{(0)} = 0 \forall i$. Ties are broken arbitrarily if there are elements with the same values $x_{\text{convex},i} = x_{\text{convex},j}$ with $i \neq j$. This also applies for further possible ties throughout the paper.

Another simple way to create a feasible solution would be using a forward greedy sensor selection, like the cost-effective algorithm proposed in [21], which in every iteration adds the sensor with the lowest ratio of objective function improvement divided by sensor cost:

$$i^* = \arg \min_{i \in \mathcal{B} \cup \{i \mid x_i^{(K-1)} = 0\}} \frac{f(x^{(K-1)} + e^i)}{c_i} \quad (14)$$

$$x^{(K)} = x^{(K-1)} + e^{i^*}$$

We denote the value corresponding to the final solution x_{greedy} of (14) by f_{greedy} . Then, since x_{opt} is a feasible suboptimal solution of (12), and x_{feas} and x_{greedy} are feasible suboptimal solutions of (11), the following holds for all metrics:

$$f_{\text{convex}} \leq f_{\text{opt}} \leq \min(f_{\text{greedy}}, f_{\text{feas}}) \quad (15)$$

VI. OPTIMIZATION METHODS

Since we are considering large networks as the 8500-node feeder in [22], which translates into having a large number of optimization variables and constraints, standard optimization methods, such as second-order or dual Lagrangian methods, are not well suited to solve the convex problem (12). In this section, we take advantage of the structure of the constraints to propose an approach based on first-order projected subgradient methods to get the optimal solution:

$$\begin{aligned} x^{(k+1)} &= \Pi_{\mathcal{X}}(x^{(k)} - \alpha^{(k)} \nabla f(x^{(k)})) \\ \mathcal{X} &= \{x \mid \sum_i c_i x_i \leq b, x_i \in [0, 1]\} \end{aligned} \quad (16)$$

where $\Pi_{\mathcal{X}}(\cdot)$ denotes the projection on \mathcal{X} . We use $\alpha^{(k)} = \frac{\alpha}{k \|\nabla f(x^{(k)})\|_2}$ to guarantee convergence of the method [23], where α is a design parameter.

A. Subgradient Computation

When developing first-order methods, the gradient expressions are required. The gradients ∇f for $f \in \{f_A, f_D\}$ can be derived analytically using matrix calculus [24]:

$$\begin{aligned} (\nabla f_A(x))_i &= -\text{tr}\left(\Sigma_{\text{post}}^2(x)(\tilde{C}_{\text{meas}})_{i,\bullet}^H(\tilde{C}_{\text{meas}})_{i,\bullet}\right)(\Sigma_{\text{meas}}^{-1})_{i,i} \\ (\nabla f_D(x))_i &= -\text{tr}\left(\Sigma_{\text{post}}(x)(\tilde{C}_{\text{meas}})_{i,\bullet}^H(\tilde{C}_{\text{meas}})_{i,\bullet}\right)(\Sigma_{\text{meas}}^{-1})_{i,i} \end{aligned} \quad (17)$$

For $f \in \{f_M, f_E\}$ no direct expression for the gradient is available, but the subgradients can be expressed as follows (see Appendix B):

$$\begin{aligned} (\partial f_M(x))_i &= -\text{tr}\left(\Sigma_{\text{post}}(x)F^H e_{\text{max}}(x)e_{\text{max}}(x)^T F \Sigma_{\text{post}}(x)\right. \\ &\quad \left.(\tilde{C}_{\text{meas}})_{i,\bullet}^H(\tilde{C}_{\text{meas}})_{i,\bullet}\right)(\Sigma_{\text{meas}}^{-1})_{i,i} \\ (\partial f_E(x))_i &= -\text{tr}\left(\Sigma_{\text{post}}(x)F^H u_{\text{max}}(x)u_{\text{max}}(x)^H F \Sigma_{\text{post}}(x)\right. \\ &\quad \left.(\tilde{C}_{\text{meas}})_{i,\bullet}^H(\tilde{C}_{\text{meas}})_{i,\bullet}\right)(\Sigma_{\text{meas}}^{-1})_{i,i} \end{aligned} \quad (18)$$

where $e_{\text{max}}(x) = e^{i^*}$ with the index i^* corresponding to the maximum diagonal entry: $i^* = \arg \max_i (\Sigma_{\text{post}}(x))_{i,i}$; and $u_{\text{max}}(x) = \arg \max_{\|u\|_2=1} u^H \Sigma_{\text{post}}(x)u$ is the eigenvector corresponding to $\lambda_{\text{max}}(x)$. For a more efficient implementation in terms of number of operations, we rewrite (17) and (18) as

$$\begin{aligned} (\nabla f_A(x))_i &= -(\tilde{C}_{\text{meas}})_{i,\bullet} \Sigma_{\text{post}}^2(x) (\tilde{C}_{\text{meas}})_{i,\bullet}^H (\Sigma_{\text{meas}}^{-1})_{i,i} \\ (\nabla f_D(x))_i &= -(\tilde{C}_{\text{meas}})_{i,\bullet} \Sigma_{\text{post}}(x) (\tilde{C}_{\text{meas}})_{i,\bullet}^H (\Sigma_{\text{meas}}^{-1})_{i,i} \\ (\partial f_M(x))_i &= -\left|(\tilde{C}_{\text{meas}})_{i,\bullet} \Sigma_{\text{post}}(x) F^H e_{\text{max}}(x)\right|^2 (\Sigma_{\text{meas}}^{-1})_{i,i} \\ (\partial f_E(x))_i &= -\left|(\tilde{C}_{\text{meas}})_{i,\bullet} \Sigma_{\text{post}}(x) F^H u_{\text{max}}(x)\right|^2 (\Sigma_{\text{meas}}^{-1})_{i,i} \end{aligned} \quad (19)$$

B. Projection Algorithm

Computing the projection $\Pi_{\mathcal{X}}(\cdot)$ in (16) by solving an optimization problem or using the algorithm proposed in [8] may be unpractical for large networks since they may require a large number of iterations, and thus function evaluations. Therefore, we propose a more computationally efficient projection algorithm, in the sense that it is noniterative with a time complexity $\mathcal{O}(n_x)$, where n_x is the dimension of x . First, we consider the change of variables $y_i = x_i c_i$ and function $f_c(y) = f\left(\frac{y}{c}\right)$, and solve the equivalent problem:

$$\min_y f_c(y) \text{ s.t. } \sum_i y_i \leq b, y_i \in [0, c_i] \forall i \quad (20)$$

using the projected subgradient descent method:

$$\begin{aligned} y^{(k+1)} &= \Pi_{\mathcal{Y}}(y^{(k)} - \alpha^{(k)} \nabla f_c(y^{(k)})) \\ \mathcal{Y} &= \{y \mid \sum y_i \leq b, y_i \in [0, c_i]\} \end{aligned} \quad (21)$$

Then, an algorithm for a computationally efficient projection $\Pi_{\mathcal{Y}}(\cdot)$ can be derived by modifying the scaled boxed-simplex projection algorithm proposed in [16]. This algorithm consists in finding a δ and assigning $y_i = \min(\max(z_i - \delta, 0), c_i) \forall i$, such that $\sum y_i \leq b$, see Appendix A. Then the algorithm can be summarized in the following steps:

1 Null sensors: Find sensor indices i that will have $y_i = 0$ after the projection, by sorting z in ascending order and determining the index $i_{0,\geq b}$ of the largest $z_{i_{0,\geq b}}$ such that $\sum_i \min(\max(z_i - z_{i_{0,\geq b}}, 0), c_i) \geq b$.

2 Full sensors: Find sensor indices i that will have $y_i = c_i$ after the projection, by sorting $\tilde{z} = z - c$ in descending order and determine the index $i_{1,\leq b}$ of the smallest $\tilde{z}_{i_{1,\leq b}}$ such that $\sum_i \min(\max(z_i - \tilde{z}_{1,\leq b}, 0), c_i) \leq b$.

3 Partial sensors: Find the values y_i of sensor indices i that will have $y_i \in (0, c_i)$ after the projection, by computing δ such that $b = \sum_{\{i \mid z_i \geq \tilde{z}_{1,\leq b} + c_i\}} c_i + \sum_{\{i \mid z_{i_{0,\geq b}} < z_i < \tilde{z}_{1,\leq b} + c_i\}} (z_i - \delta)$:

$$\delta = \frac{-b + \sum_{\{i \mid z_i \geq \tilde{z}_{1,\leq b} + c_i\}} c_i + \sum_{\{i \mid z_{i_{0,\geq b}} < z_i < \tilde{z}_{1,\leq b} + c_i\}} z_i}{\left| \{i \mid z_{i_{0,\geq b}} < z_i < \tilde{z}_{1,\leq b} + c_i\} \right|} \quad (22)$$

and return: $y_i = \min(\max(z_i - \delta, 0), c_i) \forall i$.

Algorithm 1 $\Pi_{\mathcal{Y}}$: Projection onto \mathcal{Y}

Require: $b \in \mathbb{R}, z, c \in \mathbb{R}^{n_x}, z \notin \mathcal{Y}, z_i \geq 0 \forall i$

```

1:  $z_{i_{0,\geq b}} \leftarrow 0$ 
2: if  $\{j \mid \sum_i \min(\max(z_i - z_j, 0), c_i) \geq b\} \neq \emptyset$  then
3:    $z_{i_{0,\geq b}} \leftarrow \max\{j \mid \sum_i \min(\max(z_i - z_j, 0), c_i) \geq b\} z_j$  (step 1)
4: end if
5:  $\tilde{z} = z - c$ 
6:  $\tilde{z}_{i_{1,\leq b}} \leftarrow \min\{j \mid \sum_i \min(\max(z_i - \tilde{z}_j, 0), c_i) \leq b\} \tilde{z}_j$  (step 2)
7: if  $\sum_i \min(\max(z_i - z_{i_{0,\geq b}}, 0), c_i) = b$  then
8:    $\delta \leftarrow z_{i_{0,\geq b}}$  (case 1)
9: else if  $\sum_i \min(\max(z_i - \tilde{z}_{i_{1,\leq b}}, 0), c_i) = b$  then
10:   $\delta \leftarrow \tilde{z}_{i_{1,\leq b}}$  (case 2)
11: else
12:   $\delta \leftarrow$  (22) (case 3, step 3)
13: end if
14:  $y_i \leftarrow \min(\max(z_i - \delta, 0), c_i) \forall i$ 
15: return  $y$ 

```

Note that Algorithm 1 requires $z_i \geq 0 \forall i$, which will be always satisfied since in (21) we have $y_i^{(k)} \geq 0$ and $(\nabla_y f_c(y)|_{y=y^{(k)}})_i = \frac{1}{c_i} (\nabla_x f(x)|_{x=y^{(k)}})_i \leq 0$ due to the expressions in (19). Moreover, Algorithm 1 has time complexity $\mathcal{O}(n_x)$, since finding $z_{i_{0,\geq b}}$ and $\tilde{z}_{i_{1,\leq b}}$ requires $\mathcal{O}(n_x)$ operations.

Proposition 1. *Algorithm 1 produces the projection $y = \Pi_{\mathcal{Y}}(z)$ for a $z \notin \mathcal{Y}$ (proof in Appendix A).*

VII. TEST CASE

We have tested the algorithms on the 8500-node test feeder [22] for the different metrics. We assign random normal distributed costs: $c_i \sim \mathcal{N}(1, 0.1)$. The algorithms are coded in Python and run on an Intel Core i7-6700HQ CPU at 2.60GHz with 16GB of RAM. Fig. 1 shows the bounds for the 8500-node test feeder. The A,D,E,M-optimal metrics are analyzed under a budget constraint. The yellow shaded area with horizontal and vertical lines shows the area between the minimum upper bound and the maximum lower bound, and thus the possible locations of the optimal values $f_{\{A,D,E,M\},\text{opt}}$.

It can be observed is that for all metrics, the feasible solutions $f_{\{A,D,E,M\},\text{feas}}$ (13) perform worse than the greedy ones. For the A,E,M-optimal metrics the convexity-based bound $f_{\{A,E,M\},\text{convex}}$ (12) is relatively close to the greedy solution

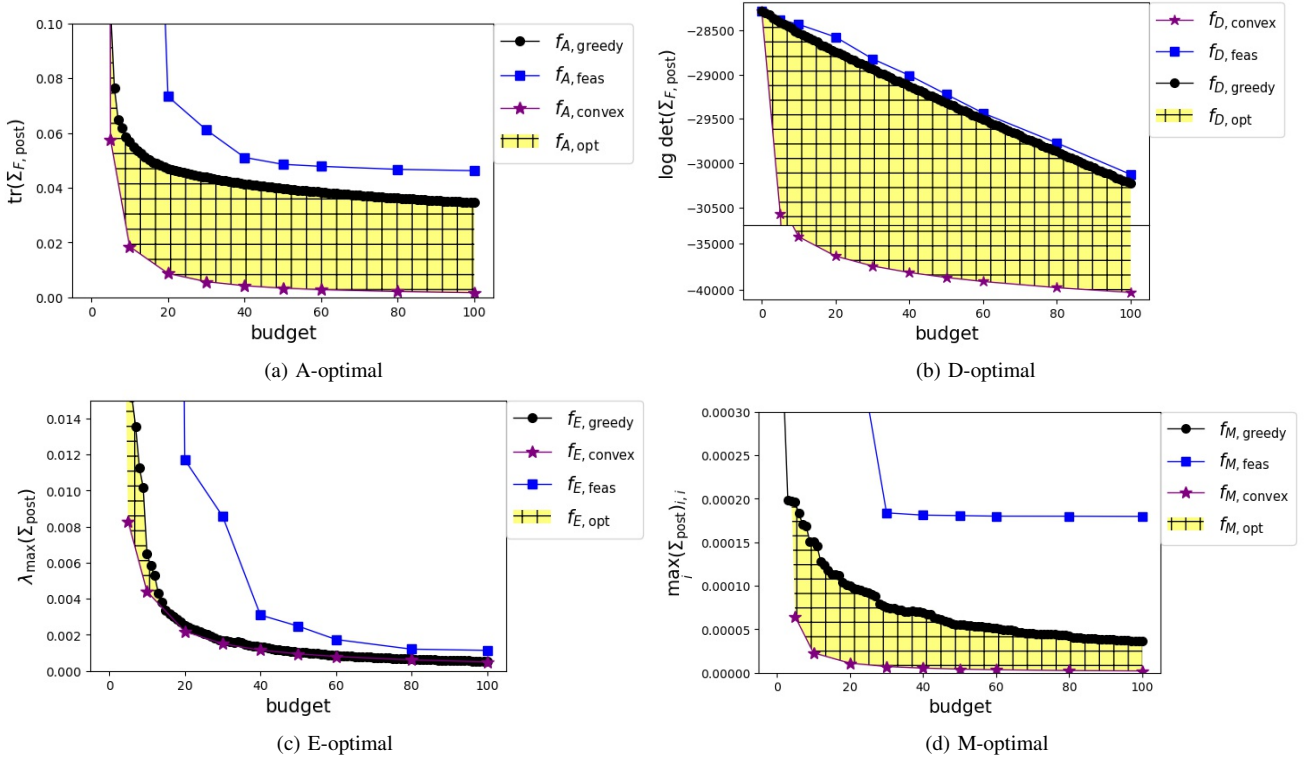


Fig. 1: Plots for the A,D,E,M-optimal metrics under a budget constraint, showing the lower bounds based on convex relaxations and the upper bounds given by the greedy and feasible solutions. The yellow shaded area with horizontal and vertical lines shows the possible locations of the optimal values $f_{\{A,D,E,M\},opt}$.

$f_{\{A,E,M\},greedy}$ (14), especially for the E-optimal metric. This means that the performance of the greedy solutions is close to the performance of the optimal one. For the D-optimal metric, the convexity-based bound is far away from the greedy and feasible solutions. Therefore, this bound does not inform about the quality of the greedy and feasible solutions.

VIII. CONCLUSIONS

We have analyzed the problem of optimal PMU placement to minimize the uncertainty of state estimation in distribution grids under a budget constraint. Using the convexity of the metrics considered, we have developed computationally efficient first-order projected subgradient descent methods that combine subgradient expressions with efficient projection algorithms. We have used these optimization methods to solve the convexified optimal sensor placement problem and to derive a lower bound for the value of the optimal solution for large distribution grids. For the A,E,M-optimal metrics, we have observed how suboptimal solutions have a close to optimal performance.

Future work could include extending these results to take into account network reconfiguration due to switches. Additionally, more exhaustive search algorithms could be developed to obtain solutions with performance closer to the convexity-based lower bound, and in parallel, apply branch and bound methods to push up the convexity-based lower bounds. Moreover, for the D-optimal metric, others bounds based on other

properties like supermodularity could be used to obtain tighter bounds [20].

APPENDIX A PROOF OF BUDGET PROJECTION

Proof. Note that the projection $y = \Pi_Y(z)$ is equivalent to solving the optimization problem:

$$\min_y \frac{1}{2} \|y - z\|_2^2 \text{ s.t. } \sum y_i \leq b, y_i \in [0, c_i] \quad (23)$$

with Lagrangian $L = \frac{1}{2} \|y - z\|_2^2 + \bar{\mu}^T (y - c) + \underline{\mu}^T (-y) + \lambda (\sum y_i - b)$. Here we prove that Algorithm 1 produces a y that satisfies the Karush-Kuhn-Tucker conditions for some $\bar{\mu}, \underline{\mu}, \lambda$:

$$\begin{aligned} \text{stability: } & \nabla_{y_i} L = y_i - z_i + \bar{\mu}_i - \underline{\mu}_i + \lambda = 0 \quad \forall i \\ \text{primal feasibility: } & y_i - c_i \leq 0, -y_i \leq 0 \quad \forall i, \sum y_i - b \leq 0 \\ \text{dual feasibility: } & \bar{\mu}_i \geq 0, \underline{\mu}_i \geq 0 \quad \forall i, \lambda \geq 0 \\ \text{complementary slackness: } & (y_i - c_i) \bar{\mu}_i = y_i \underline{\mu}_i = 0 \quad \forall i \\ & \lambda (\sum y_i - b) = 0 \end{aligned} \quad (24)$$

According to the complementary slackness and stability conditions we need to have:

$$\begin{aligned} \text{If } y_i = c_i, & \text{ then } \underline{\mu}_i = 0, \bar{\mu}_i = z_i - c_i - \lambda \\ \text{If } y_i = 0, & \text{ then } \underline{\mu}_i = \lambda - z_i, \bar{\mu}_i = 0 \\ \text{If } y_i \in (0, c_i), & \text{ then } \underline{\mu}_i = 0, \bar{\mu}_i = 0, y_i = z_i - \lambda \\ \text{If } \sum y_i < b, & \text{ then } \lambda = 0 \end{aligned} \quad (25)$$

Given (25), finding a solution corresponds to finding a $\lambda \geq 0$ so that when assigning $y_i = \min(\max(z_i - \lambda, 0), c_i) \quad \forall i$, the

conditions $\sum y_i \leq b$, $\lambda(\sum y_i - b) = 0$ are met. The remaining conditions in (24) are satisfied automatically by construction. If Algorithm 1 falls into cases 1 or 2, the choices for λ are simple: $z_{i_0, \geq b}$ or $\tilde{z}_{i_1, \leq b}$ respectively. Let us consider case 3, which means that there is at least one $y_i \in (0, c_i)$, $y_i < z_i$, so that $\lambda > 0$ and $\sum y_i = b$. First, define the indices of z and \tilde{z} following immediately after $z_{i_0, \geq b}$ and $\tilde{z}_{i_1, \leq b}$ respectively in Algorithm 1:

$$\begin{aligned} i_{0, < b} &\leftarrow \arg \min_{\{j | z_j > z_{i_0, \geq b}\}} z_j \\ &= \arg \min_{\{j | \sum_i \min(\max(z_i - z_j, 0), c_i) < b\}} z_j \\ i_{1, > b} &\leftarrow \arg \max_{\{j | \tilde{z}_j < \tilde{z}_{i_1, \leq b}\}} \tilde{z}_j \\ &= \arg \max_{\{j | \sum_i \min(\max(z_i - \tilde{z}_j, 0), c_i) > b\}} \tilde{z}_j \end{aligned} \quad (26)$$

If falling into case 3, by construction we know that a λ satisfying the conditions in (24) will be so that

$$z_{i_0, \geq b} < \lambda < z_{i_0, < b}, \tilde{z}_{i_1, \leq b} > \lambda > \tilde{z}_{i_1, > b} \quad (27)$$

Otherwise we would end in contradictions:

$$\begin{aligned} \lambda \leq \tilde{z}_{i_1, > b} &\implies b = \sum \min(\max(z_i - \lambda, 0), c_i) \\ &\geq \sum \min(\max(z_i - \tilde{z}_{i_1, > b}, 0), c_i) > c^3 b \\ \lambda \geq \tilde{z}_{i_1, \leq b} &\implies b = \sum \min(\max(z_i - \lambda, 0), c_i) \\ &\leq \sum \min(\max(z_i - \tilde{z}_{i_1, \leq b}, 0), c_i) < c^3 b \\ \lambda \geq z_{i_0, < b} &\implies b = \sum \min(\max(z_i - \lambda, 0), c_i) \\ &\leq \sum \min(\max(z_i - z_{i_0, < b}, 0), c_i) < c^3 b \\ \lambda \leq z_{i_0, \geq b} &\implies b = \sum \min(\max(z_i - \lambda, 0), c_i) \\ &\geq \sum \min(\max(z_i - z_{i_0, \geq b}, 0), c_i) > c^3 b \end{aligned}$$

where the last strict inequalities $< c^3$, $> c^3$ are due to being in case 3. Hence, we have

$$\begin{aligned} b &= \sum y_i = \sum \min(\max(z_i - \lambda, 0), c_i) \\ &\stackrel{(27)}{=} \sum_{\{i | z_i \geq \tilde{z}_{i_1, \leq b} + c_i\}} c_i + \sum_{\{i | z_{i_0, \geq b} < z_i < \tilde{z}_{i_1, \leq b} + c_i\}} (z_i - \lambda) \end{aligned}$$

Therefore we choose $\lambda = \delta$ from (22) to satisfy (24). ■

APPENDIX B

SUBGRADIENTS OF $f_{\{E, M\}}$

Let us consider the function: $f_u(x) = u^H \Sigma_{\text{post}}(x) u$, which is convex in x . Then the metrics can be reformulated as $f_E(x) = \max_{\|u\|_2=1} f_u(x)$, $f_M(x) = \max_{u \in \{e^1, \dots, e^N\}} f_u(x)$. Let x_E^* , x_M^* denote the optimum x for each metric: $x^* = \arg \min_{x^T x \leq b} f(x)$. Then we have $\forall \tilde{x}$:

$$\begin{aligned} (\nabla_x f_{u_{\max}(\tilde{x})}(x)|_{x=\tilde{x}})^T (x_E^* - \tilde{x}) &\stackrel{a)}{\leq} f_{u_{\max}(\tilde{x})}(x_E^*) - f_{u_{\max}(\tilde{x})}(\tilde{x}) \\ &\stackrel{b)}{\leq} f_{u_{\max}(x_E^*)}(x_E^*) - f_{u_{\max}(\tilde{x})}(\tilde{x}) \\ (\nabla_x f_{e_{\max}(\tilde{x})}(x)|_{x=\tilde{x}})^T (x_M^* - \tilde{x}) &\stackrel{a)}{\leq} f_{e_{\max}(\tilde{x})}(x_M^*) - f_{e_{\max}(\tilde{x})}(\tilde{x}) \\ &\stackrel{b)}{\leq} f_{e_{\max}(x_M^*)}(x_M^*) - f_{e_{\max}(\tilde{x})}(\tilde{x}) \end{aligned}$$

where a) holds because $f_u(x)$ is convex in $x \forall u$; b) because $u_{\max}(x_E^*)$ and $e_{\max}(x_M^*)$ maximize $f_u(x_E^*)$ and $f_{e^i}(x_M^*)$ over u and e^i respectively. So $\nabla_x f_{u_{\max}(\tilde{x})}(\tilde{x})$ and $\nabla_x f_{e_{\max}(\tilde{x})}(\tilde{x})$ are subgradients at \tilde{x} of f_E and f_M respectively, and can be computed using matrix algebra [24] as in (17):

$$\begin{aligned} (\nabla_x f_u(x)|_{x=\tilde{x}})_i &= -\text{tr} \left(\Sigma_{\text{post}}(x) F^H u u^H F \Sigma_{\text{post}}(x) \right. \\ &\quad \left. (\tilde{C}_{\text{meas}})_{i, \bullet}^H (\tilde{C}_{\text{meas}})_{i, \bullet} \right) (\Sigma_{\text{meas}}^{-1})_{i, i} \end{aligned}$$

REFERENCES

[1] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. CRC Press, 2004.

[2] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

[3] T. Baldwin, L. Mili, M. Boisen, and R. Adapa, "Power system observability with minimal phasor measurement placement," *IEEE Transactions on Power Systems*, vol. 8, no. 2, pp. 707–715, 1993.

[4] B. Gou, "Generalized integer linear programming formulation for optimal PMU placement," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1099–1104, 2008.

[5] B. Pinte, M. Quinlan, and K. Reinhard, "Low voltage micro-phasor measurement unit (μ PMU)," in *IEEE Power and Energy Conference at Illinois (PECI)*, Feb 2015, pp. 1–4.

[6] M. B. Mohammadi, R. Hooshmand, and F. H. Fesharaki, "A new approach for optimal placement of PMUs and their required communication infrastructure in order to minimize the cost of the WAMS," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 84–93, Jan 2016.

[7] M. Zhou, V. A. Centeno, J. S. Thorp, and A. G. Phadke, "An alternative for including phasor measurements in state estimators," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1930–1937, 2006.

[8] V. Kekatos, G. B. Giannakis, and B. Wollenberg, "Optimal placement of phasor measurement units via convex relaxation," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1521–1530, 2012.

[9] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 2006.

[10] L. Schenato, G. Barchi, D. Macii, R. Arghandeh, K. Poolla, and A. V. Meier, "Bayesian linear state estimation using smart meters and pmu measurements in distribution grids," in *IEEE International Conference on Smart Grid Communications*, Nov 2014, pp. 572–577.

[11] A. von Meier, D. Culler, A. McEachern, and R. Arghandeh, "Micro-synchrophasors for distribution systems," in *IEEE/PES Innovative Smart Grid Technologies Conference (ISGT)*, Feb 2014, pp. 1–5.

[12] R. Singh, B. C. Pal, and R. B. Vinter, "Measurement placement in distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 668–675, 2009.

[13] R. Singh, B. C. Pal, R. A. Jabr, and R. B. Vinter, "Meter placement for distribution system state estimation: an ordinal optimization approach," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2328–2335, 2011.

[14] J. Liu, J. Tang, F. Ponci, A. Monti, C. Muscas, and P. A. Pegoraro, "Trade-offs in pmu deployment for state estimation in active distribution grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 915–924, June 2012.

[15] S. Prasad and D. M. V. Kumar, "Trade-offs in PMU and IED deployment for active distribution state estimation using multi-objective evolutionary algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1298–1307, June 2018.

[16] A. Aghazadeh, M. Golbabaee, A. Lan, and R. Baraniuk, "Insense: Incoherent sensor selection for sparse signals," *Signal Processing*, vol. 150, pp. 57–65, 2018.

[17] M. Picallo, A. Anta, A. Panosyan, and B. De Schutter, "A two-step distribution system state estimator with grid constraints and mixed measurements," in *IEEE Power Systems Computation Conference*, June 2018.

[18] K. Martin *et al.*, "Exploring the IEEE standard C37. 118–2005 synchrophasors for power systems," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 1805–1811, 2008.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[20] Q. Li, R. Negi, and M. D. Ilić, "Phasor measurement units placement for power system state estimation: A greedy approach," in *IEEE Power and Energy Society General Meeting*, July 2011, pp. 1–8.

[21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2007, pp. 420–429.

[22] R. F. Arritt and R. C. Dugan, "The IEEE 8500-node test feeder," in *IEEE PES Transmission and Distribution Conference and Exposition*, April 2010, pp. 1–6.

[23] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[24] M. Brooks, "The Matrix Reference Manual," <http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html>, 2011, [Online].