A Data-Driven Method for Locating Sensors and Selecting Alarm Thresholds to Identify Violations of Voltage Limits in Distribution Systems

Paprapee Buason, *Student Member, IEEE,* Sidhant Misra, *Member, IEEE,*
Daniel K. Molzahn, *Senior Member, IEEE*

***Abstract*—Stochastic fluctuations in power injections from distributed energy resources (DERs) combined with load variability can cause constraint violations (e.g., exceeded voltage limits) in electric distribution systems. To monitor grid operations, sensors are placed to measure important quantities such as the voltage magnitudes. In this paper, we consider a sensor placement problem which seeks to identify locations for installing sensors that can capture all possible violations of voltage magnitude limits. We formulate a bilevel optimization problem that minimizes the number of sensors and avoids false sensor alarms in the upper level while ensuring detection of any voltage violations in the lower level. This problem is challenging due to the nonlinearity of the power flow equations and the presence of binary variables. Accordingly, we employ recently developed conservative linear approximations of the power flow equations that overestimate or underestimate the voltage magnitudes. By replacing the nonlinear power flow equations with conservative linear approximations, we can ensure that the resulting sensor locations and thresholds are sufficient to identify any constraint violations. Additionally, we apply various problem reformulations to significantly improve computational tractability while simultaneously ensuring an appropriate placement of sensors. Lastly, we improve the quality of the results via an approximate gradient descent method that adjusts the sensor thresholds. We demonstrate the effectiveness of our proposed method for several test cases, including a system with multiple switching configurations.**

***Index Terms*—Sensor placement, voltage violations, bilevel optimization, approximate gradient descent.**

I. Introduction

Distributed energy resources (DERs) are being rapidly deployed in distribution systems. Fluctuations in the power outputs of DERs and varying load demands can potentially cause violations of voltage limits, i.e., voltages outside the bounds imposed in the ANSI C84.1 standard. These violations can cause equipment malfunctions, failures of electrical components, and, in severe situations, power outages.

To mitigate the impacts of violations, distribution system operators (DSOs) must identify when power injection fluctuations lead to voltages exceeding their limits. To do so, sensors are placed within the distribution system to measure and communicate the voltage magnitudes at their locations. Due to the high cost of sensors and the structure of distribution systems, sensors are not placed at all buses. The question arises whether a voltage violation occurring at a location where a sensor is not placed can be detected.

Various studies have proposed sensor siting methods to capture constraint violations and outages in power systems. For instance, references [1] and [2] focus on a cost minimization problem that aims to capture all node (e.g., voltage magnitude) and line (e.g., power flow) outages. However, these references assume that a power source/generation is only located at the root node, which is not always the case, especially in the distribution systems where DERs can be located further down a feeder. Additional research efforts such as [3]–[6] seek to locate the minimum number of sensors to achieve full observability for the system. Alternatively, instead of considering full observability for the entire system, [7] considers satisfying observability requirements given a probability of observability at each bus. Other research efforts, such as [8] and [9], focus on voltage control schemes that prevent voltage violations. These efforts take a control perspective rather than extensively considering how to best place sensors. There is also research on siting phasor measurement units, which are utilized as sensors in power grids [10].

In this paper, we consider a *sensor placement* problem which seeks to locate the minimum number of sensors and determine corresponding sensor alarm thresholds in order to reliably identify all possible violations of voltage magnitude limits in a distribution system. We formulate this sensor placement problem as a bilevel optimization with an upper level that minimizes the number of sensors and chooses sensor alarm thresholds and a lower level that computes the most extreme voltage magnitudes within given ranges of power injection variability. This problem additionally aims to reduce the number of false positive alarms, i.e., violations of the sensors' alarm thresholds that do not correspond to an actual voltage limit violation.

In contrast to previous work, this problem *does not* attempt to ensure full observability of the distribution system. Rather, we seek to locate (a potentially smaller number of) sensors that can nevertheless identify all voltage limit violations for any power injections within a specified range of power injection variability. With a small number of sensors, the proposed formulation also provides a simple means to design corrective actions if voltage violations are encountered in real-time operations. By restoring voltages at these few critical locations

P. Buason and D.K. Molzahn are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30313 USA. Email: {pbuason6, molzahn}@gatech.edu. They were supported by the National Science Foundation under grant number 2023140.

S. Misra is with Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: sidhant@lanl.gov. He is supported by the advanced grid modeling (AGM) program of the Department of Energy, Office of Electricity.

to within their alarm thresholds, the system operator can guarantee feasibility of the voltage limits for the full system. This guarantee is obtained by our sensor placement method purely by analyzing the geometric properties of the feasible set. We do not consider the design details of the feedback control protocol and thus dynamic properties of the sensors such as latency are not relevant in our approach.

Due to the nonlinear nature of the AC power flow equations, computing a globally optimal solution is challenging. We utilize *conservative linear approximations* of the power flow equations to convert the lower-level problem to a linear program [11]. This bilevel problem can be reformulated to a single-level problem using the Karush-Kuhn-Tucker (KKT) conditions with binary variables via a big-M formulation [12]–[14]. In this paper, we consider a duality-based approach, which has substantial computational advantages over traditional KKT-based approaches.

Note that conservativeness from the conservative linear approximations may increase the number of false positive alarms. We therefore propose an approximate gradient descent method as a post-processing step to further improve the quality of the results. This method iteratively adjusts the sensor thresholds while ensuring that all violations are still detected.

In summary, our main contributions are:

(*i*) A bilevel optimization formulation for a sensor placement problem that minimizes the number of sensors needed to capture all possible violations of voltage limits while minimizing the number of false positive alarms.
(*ii*) Reformulations that substantially improve the computational tractability of this bilevel problem.
(*iii*) An approximate gradient descent method to improve solution quality.
(*iv*) Numerical demonstration of our proposed problem formulations for a variety of test cases, including networks with multiple switching configurations.

This paper is organized as follows. Section II formulates the sensor placement problem using bilevel optimization. Section III proposes different techniques to reformulate the optimization problem. Section IV provides our numerical tests. Section V concludes the paper and discusses future work.

II. Sensor Placement Problem

This section describes the sensor placement problem by introducing notation, presenting the bilevel programming formulation that is the focus of this paper, and detailing the objective function that simultaneously minimizes the number of sensors and reduces the number of false positive alarms.

A. Notation

Consider an n-bus power system. The sets of buses and lines are denoted as $\mathcal{N} = \{1, \dots, n\}$ and $\mathcal{L}$, respectively. One bus in the system is specified as the slack bus where the voltage is $1\angle 0^\circ$ per unit. For the sake of simplicity, the remaining buses are modeled as PQ buses with given values for their active (P) and reactive (Q) power injections. Extensions to consider PV buses, which have given values for the active power (P) and the voltage magnitude (V), are straightforward. (The controlled voltage magnitudes at PV buses imply that voltage violations cannot occur at these buses so long as the voltage magnitude setpoints are within the voltage limits.) The set of all nonslack buses is denoted as $\mathcal{N}_{PQ}$. The set of neighboring buses to bus i is defined as $\mathcal{N}_i := \{k \,|\, (i,k) \in \mathcal{L}\}$. Subscript $(\cdot)_i$ denotes a quantity at bus i, and subscript $(\cdot)_{ik}$ denotes a quantity associated with the line from bus i to bus k, unless otherwise stated. Conductance (susceptance) is denoted as G (B) as the real (imaginary) part of the admittance.

To illustrate the main concepts in this paper, we consider a balanced single-phase equivalent network representation rather than introducing the additional notation and complexity needed to model an unbalanced three-phase network. Our work does not require assumptions regarding a radial network structure, and we are able to handle multiple network configurations (i.e., a network with a set of topologies), as discussed later in Section III-H. Extensions to consider other limits such as restrictions on line flows, alternative characterizations of power injection ranges such as budget uncertainty sets, and unbalanced three-phase network models impose limited additional complexity.

B. Bilevel optimization formulation

The main goal of this problem is to find sensor location(s) such that sensor(s) can capture all possible voltage violations. We formulate this problem as a bilevel optimization with the following upper-level and lower-level problems.

- **Upper level:** Determine sensor locations and alarm thresholds such that whenever the voltages at the sensors are within the chosen thresholds, the voltages at all other buses are within pre-specified safety limits.
- **Lower level:** Find the extreme achievable voltages at all buses given the sensor locations, sensor alarm thresholds, and the specified range of power injection variability.

The sensor locations and alarm thresholds output from the upper-level problem are input to the lower-level problem, and the extreme achievable voltage magnitudes output from the lower-level problem are used to evaluate the bounds in the upper-level problem. We first introduce notation for various quantities associated with the voltage at bus i:

$\underset{\sim}{V}_i$ ($\widetilde{V}_i$)	: Lower (Upper) sensor alarm threshold.
$\underline{V}_i$ ($\overline{V}_i$)	: Lowest (Highest) achievable voltage obtained from the lower-level problem.
$\underset{\sim}{U}_i$ ($\widetilde{U}_i$)	: Translation of lower (upper) sensor threshold via a big-M formulation; see (1c).
$V_i^{\min}$ ($V_i^{\max}$)	: Specified lower (upper) voltage limit.

We formulate the following bilevel optimization formulation:

$$\min_{\mathbf{s},\widetilde{\mathbf{V}},\underset{\sim}{\mathbf{V}}} \; c(\mathbf{s}, \widetilde{\mathbf{V}}, \underset{\sim}{\mathbf{V}}) \tag{1a}$$

$$\text{s.t. } (\forall i \in \mathcal{N}_{PQ})$$

$$\underline{V}_i \geq V_i^{\min},\ \overline{V}_i \leq V_i^{\max}, \tag{1b}$$

$$\underset{\sim}{U}_i = \underset{\sim}{V}_i s_i,\ \widetilde{U}_i = \widetilde{V}_i s_i + M(1 - s_i), \tag{1c}$$

$$\underline{V}_i = \mathbb{L}_i(\mathbf{s}, \widetilde{\mathbf{U}}, \underset{\sim}{\mathbf{U}}),\ \overline{V}_i = \mathbb{U}_i(\mathbf{s}, \widetilde{\mathbf{U}}, \underset{\sim}{\mathbf{U}}), \tag{1d}$$

where c is the cost function associated with sensor installation, $\mathbf{s}$ is a vector of sensor locations modeled as binary variables (1 if a sensor is placed, 0 otherwise). All bold quantities are vectors. The quantities $\mathbb{L}_i(\mathbf{s}, \widetilde{\mathbf{U}}, \underset{\sim}{\mathbf{U}})$ and $\mathbb{U}_i(\mathbf{s}, \widetilde{\mathbf{U}}, \underset{\sim}{\mathbf{U}})$ are the solutions to the lower-level problems which, for each $i \in \mathcal{N}_{PQ}$, are given by

$$\mathbb{L}_i(\mathbf{s}, \widetilde{\mathbf{U}}, \underset{\sim}{\mathbf{U}}) = \min_{V_i} \; V_i \quad (\mathbb{U}_i(\mathbf{s}, \widetilde{\mathbf{U}}, \underset{\sim}{\mathbf{U}}) = \max_{V_i} \; V_i) \tag{2a}$$

$$\text{s.t. } (\forall j \in \mathcal{N}_{PQ})$$

$$P_j = V_j \sum_{k \in \mathcal{N}_j} V_k (G_{jk} \cos(\theta_{jk}) + B_{jk} \sin(\theta_{jk})), \tag{2b}$$

$$Q_j = V_j \sum_{k \in \mathcal{N}_j} V_k (G_{jk} \sin(\theta_{jk}) - B_{jk} \cos(\theta_{jk})), \tag{2c}$$

$$\theta_1 = 0, \tag{2d}$$

$$\underset{\sim}{U}_j \le V_j \le \widetilde{U}_j, \tag{2e}$$

$$P_j^{\min} \le P_j \le P_j^{\max}, \tag{2f}$$

$$Q_j^{\min} \le Q_j \le Q_j^{\max}, \tag{2g}$$

where P_j and Q_j denote the active and reactive power injections at bus j within a particular lower-level problem, $\theta_{jk} := \theta_j - \theta_k$ denotes the voltage angle difference between buses j and k, and superscripts max and min denote upper and lower limits, respectively, on the corresponding quantity. The quantities $\mathbb{L}_i$ and $\mathbb{U}_i$ are functions of $\mathbf{s}$, $\widetilde{U}$, and $\underset{\sim}{U}$ as shown in (1d), but these dependencies are omitted later in this paper for the sake of notational brevity. For the upper-level problem, the objective function in (1a) minimizes a cost function $c(\mathbf{s}, \widetilde{\mathbf{V}}, \underset{\sim}{\mathbf{V}})$ associated with the sensor locations $\mathbf{s}$ and alarm thresholds $\widetilde{\mathbf{V}}$, $\underset{\sim}{\mathbf{V}}$ while ensuring that the extreme achievable voltage magnitudes calculated in the lower-level problem, $\underline{V}_i, \overline{V}_i$ are within safety limits as shown in (1b). The cost function $c(\mathbf{s}, \widetilde{\mathbf{V}}, \underset{\sim}{\mathbf{V}})$ will be detailed in the following subsection. In the lower-level problem, the objective function (2a) computes the maximum or minimum voltage magnitude for each PQ bus $i \in \mathcal{N}_{PQ}$. For each lower-level problem, constraints (2b)–(2c) are the power flow equations at each bus j, constraint (2e) enforces the voltage magnitudes to be within voltage alarm thresholds (if a sensor is placed at the corresponding bus), and constraints (2f)–(2g) model the range of variability in the net power injections. Constraint (2d) sets the angle reference for the power flow equations.

C. Cost function

Overly restrictive sensor thresholds can potentially trigger an alarm even when there are no voltage violations actually occurring in the system, thus resulting in a *false positive*. To reduce both the number of sensors and the number of false positive alarms due to unnecessarily restrictive alarm thresholds, our cost function, $c(\mathbf{s}, \underset{\sim}{\mathbf{V}}, \widetilde{\mathbf{V}})$, is:

$$c(\mathbf{s}, \underset{\sim}{\mathbf{V}}, \widetilde{\mathbf{V}}) = \sum_{i \in \mathcal{N}} c_i(s_i, \underset{\sim}{V}_i, \widetilde{V}_i) \tag{3}$$

where

$$c_i(s_i, \underset{\sim}{V}_i, \widetilde{V}_i) = \begin{cases} (\underset{\sim}{V}_i - \underline{V}_i) + (\overline{V}_i - \widetilde{V}_i) + \delta; & s_i = 1, \\ 0; & s_i = 0, \end{cases} \tag{4}$$

where δ is a specified cost of placing a sensor. When $s_i = 1$, the objective $c(\mathbf{s}, \underset{\sim}{\mathbf{V}}, \widetilde{\mathbf{V}})$ in (4) seeks to reduce the restrictiveness of the sensor alarm thresholds to have fewer false positives. Changing the value of δ in (4) models the tradeoff between placing an additional sensor and making the sensor range more restrictive. This is a crucial part of our formulation since our main goal is to identify a small number of critical locations that carry sufficient information about the feasibility of the entire network. Beyond the clear financial benefit of having to place fewer sensors, this also provides a simple and practical mechanism for deploying corrective actions in real-time. Indeed, when the system operator encounters a voltage violation, a reactive power compensation protocol that brings the voltages at these few critical locations to within the alarm thresholds will guarantee feasibility of the voltage limits for the entire network.

III. Reformulations of the Sensor Placement Problem

The bilevel problem (1) is computationally challenging due to the non-convexity in the lower-level problem induced by the AC power flow equations in (2b)–(2c) and the presence of two levels. Moreover, as we will show numerically in Section IV, traditional methods for reformulating the bilevel problem into a single-level problem suitable for standard solvers using the Karush-Kuhn-Tucker (KKT) conditions turn out to yield computationally burdensome problems. In this section, we provide methods for obtaining a tractable version of the bilevel sensor placement problem. We first use recently proposed *conservative linear approximations* of the power flow equations to convert the lower-level problem to a more tractable linear programming formulation that nevertheless retains characteristics from the nonlinear AC power flow equations. We then use various additional reformulation techniques that yield significantly more tractable problems than standard KKT-based reformulations. These reformulations first yield a (single-level) mixed-integer bilinear programming formulation that can be solved using commercial mixed-integer programming solvers like Gurobi. Via further reformulations that discretize the sensor alarm thresholds, we transform the bilinear terms to obtain a mixed-integer linear program (MILP) that has further computational advantages and is suitable for a broader range of solvers.

A. Conservative Linear Power Flow Approximations

To address challenges from power flow nonlinearities, we use a linear approximation of the power flow equations that is adaptive (i.e., tailored to a specific system and a range of load variability) and conservative (i.e., over- or under-estimate a quantity of interest to avoid constraint violations). These linear approximations are called *conservative linear approximations* (CLAs) and were first proposed in [11]. As a sample-based approach, the CLAs are computed using the solution to a constrained regression problem. They linearly relate the voltage magnitudes at a particular bus to the power

injections at all PQ buses. An example of an *overestimating* CLA of the voltage magnitude at bus i is the linear expression

$$a_{i,0} + \boldsymbol{a}_{i,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}$$

such that the following relationship is satisfied for some specified range of power injections $\mathbf{P}$ and $\mathbf{Q}$:

$$V_i - \left(a_{i,0} + \boldsymbol{a}_{i,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix} \right) \leq 0, \tag{5}$$

where and superscript T denotes the transpose. We replace the AC power flow equations in (2b)–(2d) with a CLA as in (5) for all $i \in \mathcal{N}_{PQ}$. The bilevel problem in (1) is modified by replacing the lower-level problem in (2) by

$$\mathbb{L}_i = \min_{\mathbf{P},\mathbf{Q}} \ \underline{a}_{i,0} + \underline{\boldsymbol{a}}_{i,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \quad \left(\mathbb{U}_i = \max_{\mathbf{P},\mathbf{Q}} \ \overline{a}_{i,0} + \overline{\boldsymbol{a}}_{i,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \right) \tag{6a}$$

$$\text{s.t.} \ \ (\forall j \in \mathcal{N}_{PQ} \setminus \{i\})$$

$$\overline{a}_{j,0} + \overline{\boldsymbol{a}}_{j,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \geq \underset{\sim}{U}_j, \tag{6b}$$

$$\underline{a}_{j,0} + \underline{\boldsymbol{a}}_{j,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \leq \widetilde{U}_j, \tag{6c}$$

$$\begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^{\min} \leq \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \leq \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^{\max}. \tag{6d}$$

In (6), superscripts i denote quantities associated with the i^{th} lower-level problem. Replacing the AC power flow equations with CLAs yields a linear programming formulation for the lower-level problem rather than the non-convex lower-level problem in (1). Comparing (6b)–(6c) and (2b)–(2e), we see that satisfaction of (6b)–(6c) is sufficient to ensure satisfaction of (1b), assuming that the conservative linear approximations do indeed reliably overestimate and underestimate the voltage magnitudes. Thus, the resulting optimization problems are sufficient to ensure that the sensor locations and alarm thresholds will identify all potential voltage limit violations. As a result, solving the reformulation (6) will compute sensor locations and thresholds that avoid *false negatives*, i.e., alarms will always be raised if there are indeed violations of the voltage limits. The computational advantages provided by linearity of the reformulated lower-level problem come with the potential tradeoff of additional false positives, i.e., spurious alarms. Various methods proposed in [11] for improving the accuracy of the CLAs naturally yield variants of (6) that reduce the number of false positives. Additionally, in Section III-G, we describe a method for post-processing the sensor alarm thresholds to further reduce the occurrences of false positives.

B. Reformulation using KKT constraints

With a linear lower-level problem, we can apply standard techniques for reformulating the bilevel problem (6) as a (single-level) mixed-integer linear program (MILP). These techniques replace the lower-level problem (6a)–(6d) with its KKT conditions that are both necessary and sufficient for optimality of this problem [13] and also apply McCormick envelopes [15] to convert the bilinear product of the continuous and discrete variables in (1c) to an equivalent linear form. The resulting single-level problem still involves bilinear constraints associated with the complementarity conditions. These bilinear constraints are traditionally addressed using binary variables in a "Big-M" formulation. Commercial MILP solvers are applicable to this traditional reformulation, which we denote throughout the paper as the "KKT formulation". This formulation is obtained by defining the lower-level coupling quantities $\mathbb{L}_i$ and $\mathbb{U}_i$ using the KKT conditions given below:

$$\mathbb{L}_i = \underline{a}_{i,0} + \underline{\boldsymbol{a}}_{i,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \quad \left(\mathbb{U}_i = \ \overline{a}_{i,0} + \overline{\boldsymbol{a}}_{i,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \right), \tag{7a}$$

$$(\forall j \in N_{PQ} \setminus \{i\})$$

$$\underline{a}_{i,1} - \sum_{k \in \mathcal{N}_{PQ}} \lambda_k^i \overline{\boldsymbol{a}}_{k,1} + \sum_{k \in \mathcal{N}_{PQ}} \mu_k^i \underline{\boldsymbol{a}}_{k,1} - \sum_{k \in \mathcal{N}_{PQ}} \gamma_k^i e_k + \sum_{k \in \mathcal{N}_{PQ}} \pi_k^i e_k = 0$$

$$\left(\text{for } \mathbb{U}_i : \overline{a}_{i,1} - \sum_{k \in \mathcal{N}_{PQ}} \lambda_k^i \overline{\boldsymbol{a}}_{k,1} + \sum_{k \in \mathcal{N}_{PQ}} \mu_k^i \underline{\boldsymbol{a}}_{k,1} - \sum_{k \in \mathcal{N}_{PQ}} \gamma_k^i e_k + \sum_{k \in \mathcal{N}_{PQ}} \pi_k^i e_k = 0\right), \tag{7b}$$

$$\overline{a}_{j,0} + \overline{\boldsymbol{a}}_{j,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \geq \underset{\sim}{U}_j, \tag{7c}$$

$$\underline{a}_{j,0} + \underline{\boldsymbol{a}}_{j,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \leq \widetilde{U}_j, \tag{7d}$$

$$\begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^{\min} \leq \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \leq \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^{\max}, \tag{7e}$$

$$\lambda_j^i \cdot \left(\underset{\sim}{U}_j - \overline{a}_{j,0} - \overline{\boldsymbol{a}}_{j,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \right) = 0, \tag{7f}$$

$$\mu_j^i \cdot \left(-\widetilde{U}_j + \underline{a}_{j,0} + \underline{\boldsymbol{a}}_{j,1}^T \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \right) = 0, \tag{7g}$$

$$\boldsymbol{\gamma}^i \odot \left(\begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^{\min} - \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i \right) = \mathbf{0}, \tag{7h}$$

$$\boldsymbol{\pi}^i \odot \left(\begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^i - \begin{pmatrix} \mathbf{P} \\ \mathbf{Q} \end{pmatrix}^{\max} \right) = \mathbf{0}, \tag{7i}$$

$$\boldsymbol{\gamma}^i, \boldsymbol{\pi}^i \geq \mathbf{0}; \lambda_j^i, \mu_j^i \geq 0, \ \forall j \in N_{PQ} \setminus \{i\}, \tag{7j}$$

where the operator $\odot$ is the element-wise multiplication; e_i is the i^{th} column of the identity matrix; $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\boldsymbol{\gamma}$ $:= (\gamma_1, \gamma_2, \ldots, \gamma_{2m})^T$, and $\boldsymbol{\pi}$ $:= (\pi_1, \pi_2, \ldots, \pi_{2m})^T$ are dual variables associated with the voltage and power injection bounds in the constraints (6b)–(6d). Note that the solution to the set of equations in $\mathbb{L}_i$ is completely decoupled from that in $\mathbb{U}_i$. Equations (7b)–(7i) are the KKT conditions of the lower-level problem. Equation (7b) is the stationarity condition. The primal feasibility conditions in (7c)–(7e) are similar to the constraints (6b)–(6d) in the original problem with conservative linear approximations of the power flow equations. The complementary slackness conditions are (7f)–(7i) and the dual feasibility condition is (7j). Observe that the complementary slackness conditions give rise to nonlinear

functions due to the multiplication of the dual variables $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\pi}$ with the primal variables $\mathbf{P}$ and $\mathbf{Q}$. To handle these nonlinearities, traditional methods for bilevel optimization replace these products using additional binary variables and a big-M reformulation. This requires bounds on the dual variables that are difficult to determine, and bad choices for these bounds can result in either infeasibility or poor computational performance [14].

C. Duality of the lower-level problem

As an alternative to the KKT formulation, one can reformulate a bilevel problem into a single-level problem by *dualizing* the lower-level problem. This technique can only be usefully applied to problems with specific structure where the optimal objective value of the lower-level problem is constrained in the upper-level problem in the appropriate sense ($\max \leq \cdot$ or $\min \geq \cdot$). In this special case, we can significantly improve tractability compared to the KKT formulation.

Let $\underset{\sim}{\mathbf{y}}^i$ be the vector of all dual variables associated with the lower-level problem $\mathbb{L}_i$ and $\widetilde{\mathbf{y}}^i$ be the vector of all dual variables associated with lower-level problem $\mathbb{U}_i$. Let I be the identity matrix of appropriate dimension. By dualizing the lower-level problem (6), we obtain the following:

$$\mathbb{L}_i = \max_{\underset{\sim}{\mathbf{y}}^i} \ \mathbf{b}^T \underset{\sim}{\mathbf{y}}^i + \underline{a}_{i,0} \quad (8a) \qquad \text{s.t. } A\underset{\sim}{\mathbf{y}}^i = \underline{\boldsymbol{a}}_{i,1}, \quad (8b) \qquad \underset{\sim}{\mathbf{y}}^i \geq \mathbf{0}, \quad (8c)$$

$$\mathbb{U}_i = \min_{\widetilde{\mathbf{y}}^i} \ \mathbf{b}^T \widetilde{\mathbf{y}}^i + \overline{a}_{i,0} \quad (9a) \qquad \text{s.t. } A\widetilde{\mathbf{y}}^i = \overline{\boldsymbol{a}}_{i,1}, \quad (9b) \qquad \widetilde{\mathbf{y}}^i \leq \mathbf{0}, \quad (9c)$$

where

$$A = \left[-I, I, \overline{\boldsymbol{a}}_{1,1}, \ldots, \overline{\boldsymbol{a}}_{n,1}, -\underline{\boldsymbol{a}}_{1,1}, \ldots, -\underline{\boldsymbol{a}}_{n,1}\right],$$
$$\mathbf{b} = [(-\mathbf{P}^{\max})^T, (-\mathbf{Q}^{\max})^T, (\mathbf{P}^{\min})^T, (\mathbf{Q}^{\min})^T, \underset{\sim}{U}_1 - \overline{a}_{1,0}, \ldots, \underset{\sim}{U}_n - \overline{a}_{n,0}, -\widetilde{U}_1 + \underline{a}_{1,0}, \ldots, -\widetilde{U}_n + \underline{a}_{n,0}]^T.$$

Due to strong duality of the linear lower-level problem, the dual (8a) (and (9a)) has the same objective value as (6a) and does not directly provide any advantages. However, the problem has a specific structure where objectives from each lower-level problem (8a) and (9a) only appear in a single inequality constraint (1b). Hence, we *only need to show that there exists some choice of duals* $\underset{\sim}{\mathbf{y}}^i$ *and* $\widetilde{\mathbf{y}}^i$ *for which* (1b) *is feasible*. This allows us to obtain a single-level formulation by defining the lower-level coupling quantities via the following set of constraints:

$$\mathbb{L}_i = \mathbf{b}^T \underset{\sim}{\mathbf{y}}^i + \underline{a}_{i,0}, \quad (10a) \qquad A\,\underset{\sim}{\mathbf{y}}^i = \underline{\boldsymbol{a}}_{i,1}, \quad (10b) \qquad \underset{\sim}{\mathbf{y}}^i \geq \mathbf{0}. \quad (10c)$$

$$\mathbb{U}_i = \mathbf{b}^T \widetilde{\mathbf{y}}^i + \overline{a}_{i,0}, \quad (11a) \qquad A\,\widetilde{\mathbf{y}}^i = \overline{\boldsymbol{a}}_{i,1}, \quad (11b) \qquad \widetilde{\mathbf{y}}^i \leq \mathbf{0}. \quad (11c)$$

We refer to the formulation using (10) and (11) as the "bilinear formulation" due to the bilinear product of the sensor threshold variables ($\underset{\sim}{\mathbf{U}}$ and $\widetilde{\mathbf{U}}$) and the dual variables $\underset{\sim}{\mathbf{y}}^i$ and $\widetilde{\mathbf{y}}^i$ in (10a) and (11a). Similar to the KKT formulation, using (10) and (11) leads to a single-level optimization problem. However, the latter has the major advantage that no additional binary variables are required (beyond those associated with the sensor locations in the upper-level problem) since there are no analogous equations to the complementarity condition. Our numerical results in Section IV show that modern mixed-integer programming packages like Gurobi can solve the bilinear formulation for much larger systems than are possible with the standard KKT formulation. However, if required, the bilinear constraints (10a) and (11a) can be further simplified with one additional reformulation that is discussed next.

D. Bilinear to mixed-integer linear programming

The bilinear formulation can be further converted into an MILP by discretizing the continuous-valued sensor thresholds. This formulation has the advantage of being within the scope of a larger range of mixed-integer programming solvers since not all of them can handle billinear forms. We partition the sensor threshold ranges into d discrete steps with size ϵ and define the vectors of threshold variables, $\underset{\sim}{\mathbf{U}}$ and $\widetilde{\mathbf{U}}$, as

$$\underset{\sim}{\mathbf{U}} = \eta^T \underset{\sim}{v}_t \quad (\widetilde{\mathbf{U}} = \eta^T \widetilde{v}_t), \tag{12}$$

where

$$\eta = \begin{bmatrix} \eta_{0,1} & \eta_{0,2} & \cdots & \eta_{0,n} \\ \eta_{1,1} & \eta_{1,2} & \cdots & \eta_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{d,1} & \eta_{d,2} & \cdots & \eta_{d,n} \end{bmatrix}, \tag{13a}$$

$$\underset{\sim}{v}_t = \left[0, V^{\min}, V^{\min} + \epsilon, \ldots, V^{\min} + (d-1)\epsilon\right]^T$$
$$(\widetilde{v}_t = \left[0, V^{\max} - \epsilon, V^{\max} - 2\epsilon, \ldots, V^{\max}\right]^T), \tag{13b}$$

$$\sum_{i=1}^{d} \eta_{i,k} = 1, \quad \forall k \in \mathcal{N}_{PQ}. \tag{13c}$$

Note that this discretization exploits the fact that any sensor threshold will necessarily be above the lower voltage limit $V_i^{\min}$ and below the upper voltage limit $V_i^{\max}$. Equations (13a)–(13c) imply that when $\eta_{0,i} = 0$, no sensor is placed (i.e., $s_i = 0$). Using this discretization, the constraints (10a) and (11a) now contain bilinear products of binary variables. These products can be equivalently transformed into a mixed-integer *linear* (as opposed to bilinear) programming formulation using McCormick Envelopes [15]. With McCormick Envelopes and discrete sensor thresholds, the problems (10) and (11) become a MILP that can be computed using any MILP solver. We refer to the reformulation of the lower-level problems (10) and (11) using the discretization (12) as the "MILP formulation".

To further improve tractability, we can remove unnecessary binary variables by inspecting data from the samples of power injections used to compute the conservative linear approximations of the power flow equations. Let b be a bus where the voltage magnitude never reaches the highest discretized sensor threshold value (i.e., $V^{\min} + (d-1)\epsilon$) in any of the sampled power injections. Given a sufficiently comprehensive sampling of the range of possible power injections, we can then simplify the discretized representation of the sensor alarm threshold as:

$$\underset{\sim}{U}_b = 0 \cdot \eta_{0,b} + V^{\min} \cdot \eta_{1,b}, \tag{14a}$$
$$\eta_{0,b} + \eta_{1,b} = 1. \tag{14b}$$

A similar simplification can be used for the upper sensor thresholds. This pre-screening thus eliminates binary variables associated with sensor thresholds at buses that will never violate their voltage limits. We henceforth call this data-driven simplification technique "binary variable removal" (BVR).

E. A sensor placement heuristic for benchmarking

For comparison with our proposed sensor placement approach, we describe a heuristic alternative that exploits the traditional behavior of distribution systems where voltage magnitudes typically drop as one moves down a feeder away from the substation. This behavior suggests that the ends of each feeder may be good locations for locating sensors since the voltage magnitude limits are likely to be violated here first.

This heuristic technique avoids the computational burden from computing the conservative linear power flow approximations and solving some reformulation of the bilevel optimization problem. However, it can fail to eliminate false negatives and produce sub-optimal results in terms of the number of required sensors. This is especially true in systems with (i) DERs towards the ends of some branches, (ii) multiple operating topologies, and (iii) a large number of branches. Illustrative examples of systems with such characteristics are provided in Section IV-D.

F. Comparisons of each formulation

The previous subsections present several problem reformulations that convert the bilevel sensor placement problem (1) into various single-level problems that can be solved with mixed-integer solvers like Gurobi. Each reformulation has different computational characteristics and yields solutions with differing accuracy. We next compare the KKT formulation (7) described in Section III-B with the duality-based bilinear formulation (10) and (11) described in Section III-C according to the numbers of decision variables and constraints.

Both formulations involve bilinear terms but the bilinear formulation is more compact. Consider a system where there are b PQ buses, of which there are r buses where the voltage magnitudes may violate their limits after the pre-screening described in Section III-D. The total number of decision variables in the KKT formulation is $5 \cdot b \cdot r + 2 \cdot b$ ($2 \cdot b \cdot r$ from power injections, $3 \cdot b \cdot r$ from dual variables, b from the voltage thresholds, and b from the sensor locations). Our proposed duality-based bilinear formulation involves only $3 \cdot b \cdot r + 2 \cdot b$ decision variables. The reduction happens because the variables corresponding to power injections are entirely removed by duality.

Regarding the number of constraints, the duality-based bilinear formulation does not have the stationarity conditions, primal feasibility, or power injections directly involved, resulting in a reduction of $2 \cdot b \cdot r + b \cdot r + 2 \cdot b \cdot r = 5 \cdot b \cdot r$ constraints. The implications of these differences on tractability is assessed via the solution times presented in Section IV.

G. Approximate Gradient Descent

Solving any of the reformulated bilevel optimization problems may lead to *false positives*, i.e., alarms that occur in the absence of a voltage limit violation. This is both due to the limited number of sensors and the conservative nature of the linear power flow approximations used in the lower-level problem. These false positives are undesirable because they may lead system operators to take unnecessary corrective actions that could reduce the efficiency of system operations or put avoidable wear on system components.

To reduce the number of false positives, this section proposes a post-processing step that iteratively adjusts the sensor thresholds that result from the reformulated bilevel optimization problems. We refer to this post-processing step as the Approximate Gradient Descent (AGD) method. This method uses the voltage magnitudes associated with a large number of sampled power injections within the considered range of power injection variability. These samples can be the same as those used to compute the conservative linear approximations of the power flow equations.

In this section, let superscript k denote the k^{th} iteration of the AGD method. Let ϵ_{AGD} be a step size for adjusting the sensor thresholds and $\mathbf{f}^k$ be the vector of the number of false positives from the sampled power injections using the sensor thresholds in the k^{th} iteration. Using the sampled power injections, this method computes an "approximate gradient" indicating how small changes to the sensor alarm thresholds affect the number of false positives. The approximate gradient at iteration k is denoted as $\mathbf{g}^k$. We denote the set of buses with sensors as $\mathcal{N}_s$. Subscripts denote the bus number and superscripts denote the iteration number.

Let Δf_i^k represent the change in the number of false positives among the sampled power injections using the sensor thresholds in the k^{th} iteration when the sensor alarm threshold i is changed by ϵ_{AGD} (leaving all other sensor thresholds unchanged). We then compute an approximate gradient $\mathbf{g}^k$ by comparing the values of Δf_i^k across different buses i:

$$\mathbf{g}^k = \frac{\Delta \mathbf{f}^k}{\sqrt{\sum_{i \in \mathcal{N}_s} (\Delta f_i^k)^2}}. \tag{15}$$

The approximate gradient $\mathbf{g}^k$ thus points in the direction of the steepest reduction in the (empirically determined) number of false positives. Each iteration of the AGD method updates the sensor thresholds according to the update rule:

$$\underset{\sim}{\mathbf{V}}^{k+1} = \underset{\sim}{\mathbf{V}}^{k} + \epsilon_{AGD} \cdot \mathbf{g}^k \quad (\widetilde{\mathbf{V}}^{k+1} = \widetilde{\mathbf{V}}^{k} + \epsilon_{AGD} \cdot \mathbf{g}^k). \tag{16}$$

The AGD method stops when taking an additional step would result in the appearance of false *negatives*, i.e., undetected violations of voltage limits. The AGD method for the upper voltage limits is calculated separately from the lower voltage limits. Fig. 1 shows the overall process for computing the sensor locations and thresholds starting from computing the conservative linear approximations of the power flow equations (a pre-processing step), solving a reformulation of the bilevel problem, and adjusting thresholds via the approximate gradient descent method (a post-processing step).

H. A system with multiple possible network configurations

Sensors are placed once and operate for extended periods of time during which network conditions may change. In

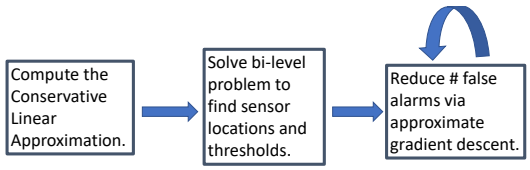


Fig. 1. A diagram showing the process of computing sensor locations and alarm thresholds. The final step is performed iteratively until encountering false negatives.

addition to the power injection variability considered in the bilevel problem (2f)–(2g), the network may be reconfigured using switches, resulting in multiple topologies. Network reconfiguration can significantly affect which locations in the network are best suited for sensor placement.

To address this, our sensor placement formulation can be extended to consider a set of possible network topologies. In this context, the goal of the sensor placement problem is to determine locations for sensors and sensor alarm thresholds that will reliably identify voltage limit violations for any topology within the considered set. The sensor locations must be selected once and remain consistent across all topologies, but the sensor alarm thresholds may vary between different topologies. With m different topologies for an n-bus system, the computational complexity is similar to solving a sensor placement problem for an $(n \times m)$-bus system with one configuration. For each topology, we compute different conservative linear approximations of the power flow equations and introduce additional lower-level problems (i.e., for a specific bus, each configuration introduces different constraints (10a)–(10c) and (11a)–(11c)). The sensor locations have one set of binary variables since they remain the same across topologies.

IV. Numerical Tests

In this section, we perform numerical experiments on a number of test cases to analyze the sensor locations and thresholds, demonstrate the advantages of our problem reformulations and the post-processing step, and compare results and computational efficiency from different problem formulations.

The test cases we use in these experiments are the 10-bus system *case10ba*, the 33-bus system *case33bw*, and the 141-bus system *case141* from MATPOWER [16]. For the CLAs, we minimize the ℓ_1 error with 1000 samples in the first iteration and 4000 additional samples in a sample selection step, and we choose a quadratic output function of voltage magnitude. (See [11] for a discussion on computationally efficient iterative methods for computing CLAs and variants of CLAs that approximate different quantities in order to improve their accuracy.) All power injections vary within 50% to 150% of the load demand values given in the MATPOWER files except for *case33bw* where we consider a variant with solar panels installed at buses 18 and 33. The loads at these two buses vary within -200% to 150% of the given values.

We implement the single-level reformulations of the sensor placement problem in MATLAB using YALMIP [17] and use Gurobi as a solver with a MIP gap tolerance of 0.005, unless otherwise stated. The AGD step size is $\epsilon_{AGD} = 2 \times 10^{-4}$ per unit. The value of δ in the objective (4) is 0.02. In *case10ba*, *case33bw*, and *case141*, the lower voltage limits are 0.90, 0.91, and 0.92, respectively, and the computation times for the CLAs are 58, 198, and 1415 seconds, respectively.

A. Sensor locations

We compare the quality of results and the computation time from the following reformulations: (i) the KKT formulation (7), (ii) the duality-based bilinear formulation (10) and (11), (iii) the MILP formulation without applying the binary variable removal (BVR) pre-processing, and (iv) the MILP formulation with the BVR pre-processing.

The first test case is the 10-bus system *case10ba*, which is a simple single-branch distribution network. We consider a variant where the loads are 60% of the values in the MATPOWER file. The results from each formulation place a sensor at the end of the branch (the furthest bus from the substation) with an alarm threshold of 0.9 per unit (at the voltage limit). Fig. 2a compares computation times from the four formulations. The the KKT formulation takes 26.7 seconds while the bilinear, the MILP without BVR, and the MILP formulations take 1.96, 1.95, and 1.54 seconds, respectively. Since the sensor threshold for the KKT and both MILP formulations is at the voltage limit, AGD is not needed. On the other hand, the bilinear formulation gives a higher alarm threshold. As a result, the AGD method is applied as a post-processing step to achieve the lowest possible threshold without introducing false alarms. The number of false positives reduces from 5.48% to 0%. Executing the AGD method takes 0.11 seconds.

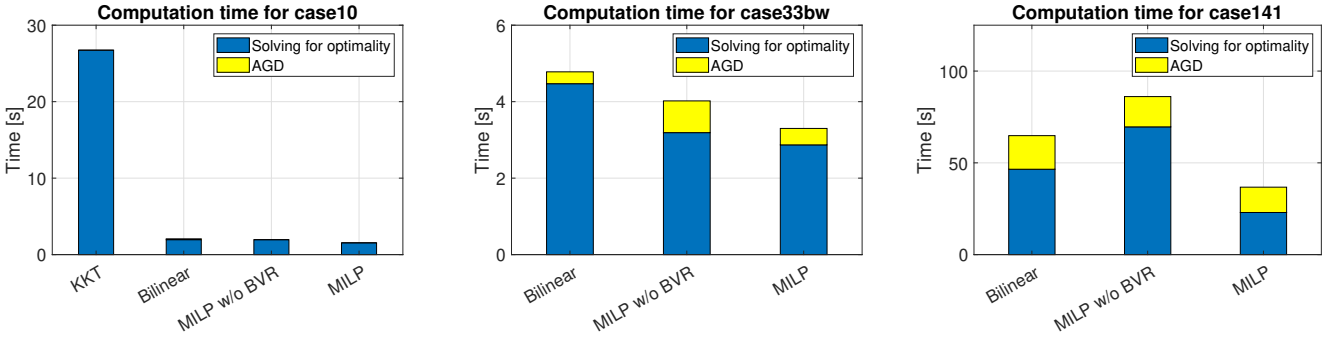
The second test case is the 33-bus system *case33bw*, which has multiple branches. Table I shows the computation times for the bilinear and the two MILP formulations. We exclude the computation time for the KKT formulation since the solver fails to find even a feasible (but potentially suboptimal) point within 55000 seconds (15 hours). Our final test case is the 141-bus system *case141*. Similar to the 33-bus system, the solver could not find the optimal solution for the KKT formulation within a time limit of 15 hours. Table I again shows the results for this test case, and Figs. 2b and 2c compare the computation times for the bilinear and MILP formulations.

Table I shows both the computation times and the results of randomly drawing sampled power injections within the specified range of variability, computing the associated voltages by solving the power flow equations, and finding the number of false positive alarms (i.e., the voltage at a bus with a sensor is outside the sensor's threshold but there are no voltage violations in the system). The results for the 33-bus and 141-bus test cases given in Table I illustrate the performance of the proposed reformulations. Whereas the KKT formulation is computationally intractable, our proposed reformulations find solutions within approximately one minute, where the MILP formulation with the BVR method typically exhibits the fastest performance. The solutions to the reformulated problems place a small number of sensors (two to four sensors in systems with an order of magnitude or more buses). No solutions suffer from false negatives since all samples where there is a voltage violation trigger an alarm. There are a number of false alarms prior to applying the AGD that after its application

TABLE I
RESULTS SHOWING SENSOR ALARM THRESHOLDS AND NUMBER OF FALSE POSITIVES FROM KKT, BILINEAR, AND MILP FORMULATIONS.

		KKT[§]	Bilinear			MILP w/o BVR			MILP		
		case10ba	*case10ba*	*case33bw*	*case141*	*case10ba*	*case33bw*	*case141*	*case10ba*	*case33bw*	*case141*
Computation time [s]	Optimality	26.7	1.96	4.47	46.52	1.95	3.19	69.6	1.54	2.87	22.95
	AGD	—	0.11	0.31	18.3	—	0.83	16.5	—	0.43	13.8
Sensor location(s)		10	10	12, 16, 32	79, 80, 82, 85	10	14, 31	76, 80, 86	10	14, 30	80, 86
Sensor threshold(s)		0.9	0.9017	0.9119, 0.9167, 0.9113	0.92, 0.9213, 0.93, 0.93	0.9	0.919 0.9195	0.92, 0.9295, 0.9295	0.9	0.9195 0.9190	0.929 0.9295
with AGD		—	0.9	0.9119, 0.9139, 0.9103	0.92, 0.9212, 0.9218, 0.9201	—	0.9167 0.9142	0.92, 0.9211, 0.9201	—	0.9167 0.9168	0.9213 0.9201
# feasible points		7317	7317	9560	9955	7317	9560	9955	7317	9560	9955
# false positive(s)		0	401	674	7207	0	2131	6664	0	1033	6664
with AGD		—	0	147	3	—	453	1	—	289	1
# false negatives		0	0	0	0	0	0	0	0	0	0

[§]The solver does not find a solution to *case33bw* and *case141* within 55000 seconds.



(a) *case10* (b) *case33bw* (c) *case141*

Fig. 2. Bar graphs showing the computation times for solving various problem formulations and executing the approximate gradient descent method.

decrease dramatically to a small fraction of the total number of samples (3.02% and 0.01% in the 33-bus and the 141-bus systems, respectively). These observations suggest that our sensor placement formulations provide a computationally efficient method for identifying a small number of sensor locations and associated alarm thresholds that reliably identify voltage constraint violations with no false negatives (missed alarms) and few false positives (spurious alarms).

B. MIP gap tolerance, solver times, and solution quality

Each of the problem formulations involves a mix of continuous and binary variables, thus requiring solution from mixed-integer programming solvers like Gurobi. These solvers are based on branch-and-bound algorithms that iteratively update upper and lower bounds on the optimal solution, terminating when the difference between these bounds (i.e., the "MIP Gap") converges to a value less than a specified tolerance. The choice of MIP gap tolerance can directly affect the computation time and the quality of the solution (e.g., sensor thresholds and number of sensors).

We investigate the effect of two different choices for the MIP gap tolerance, 0.3% and 0.1%, on the bilinear and MILP formulations for the test cases *case33bw* and *case141*. The results are shown in Table II and Table III, respectively. As expected, these results suggest that decreasing the MIP gap tolerance can increase the computational times in some cases,

TABLE II
RESULTS FROM DIFFERENT MIP GAP TOLERANCES FOR *case33bw*

		Bilinear		MILP	
MIP Gap Tolerance		**0.3%**	**0.1%**	**0.3%**	**0.1%**
Computation time [s]	Optimality	4.81	5.51	3.01	5.35
	AGD	0.37	0.22	0.75	0.13
Sensor location(s)		15, 31	18, 31	14, 32	17, 33
Sensor threshold(s)		0.9169 0.9126	0.9136 0.9126	0.9195 0.918	0.9125 0.912
with AGD		0.9152 0.9116	0.9119 0.9115	0.9167 0.9132	0.9109 0.9105
# feasible points		9560	9560	9560	9560
# false positive(s)		516	537	1941	334
with AGD		178	208	427	57
# false negatives		0	0	0	0

with approximately twice as much time required for the MILP formulation with *case33bw* and the bilinear formulation with *case141*. Nevertheless, the computation times with the smaller MIP gaps are still reasonable in all cases (less than two minutes), and the MIP gap tolerance has little impact on the computation times for other cases. Regarding solution quality, tighter MIP gap tolerances can lead to fewer sensors (as in the bilinear formulation with *case141*) and fewer false positives (as in the MILP formulation with *case33bw*). However, the results are not uniformly improved by tightening the MIP gap tolerance, as the 0.1% tolerance leads to the same number of

TABLE III
RESULTS FROM DIFFERENT MIP GAP TOLERANCES FOR *case141*

		Bilinear		MILP	
MIP Gap Tolerance		**0.3%**	**0.1%**	**0.3%**	**0.1%**
Computation time [s]	Optimality	49.9	94.3	27.6	27.7
	AGD	18.3	12.6	10.5	10.5
Sensor location(s)		79, 80, 82, 85	79, 80, 85	80, 86	80, 86
Sensor threshold(s)		0.92, 0.9213 0.93, 0.93	0.9297, 0.92, 0.93	0.929, 0.9295	0.929, 0.9295
with AGD		0.92, 0.9212 0.9218, 0.9201	0.922, 0.92, 0.9202	0.9213, 0.9201	0.9213 0.9201
# feasible points		9955	9955	9955	9955
# false positives		7207	7174	6664	6664
with AGD		3	4	1	1
# false negatives		0	0	0	0

sensors and *more* false positives than the 0.3% tolerance after applying the AGD method. This suggests a potential benefit of assessing the performance of multiple "nearly optimal" solutions obtained with different MIP gap tolerances.

C. Multiple configurations

The previous results described the sensor placements for the *case10ba*, *case33bw*, and *case141* systems in their nominal network topologies. We next demonstrate the effectiveness of our problem reformulations for variants of these systems with multiple network configurations. We consider a variant of the *case33bw* system with three different network configurations and two solar PV generators installed at buses 18 and 33. The first configuration is the nominal topology given in the MATPOWER version of the test case. In the second configuration, the line connecting buses 6 and 7 is removed and a new line connecting buses 4 and 18 is added. The third configuration removes the line connecting buses 6 and 26 and adds a new line connecting buses 25 and 33. All network configurations are displayed in Fig. 3.

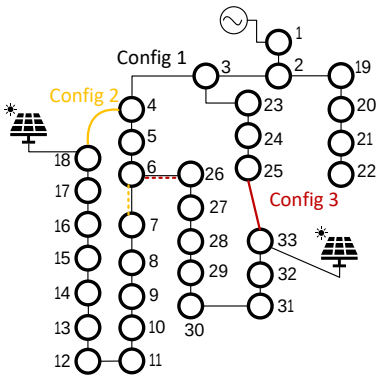


Fig. 3. The *case33bw* system with three different network configurations. A dash line means that line is removed when a solid line with the same color is added such that the system has a radial topology in each configuration.

Table IV shows the results from using the bilinear and MILP formulations to solve the multiple-configuration problem for this case. The results generally mirror those from the single-network-configuration test cases shown earlier in that computation times are still reasonable (approximately a factor of four larger) and there are no false negatives and a small number of false positives after applying the AGD method.

We note that some configurations may not need to utilize all available sensors. To show this, we describe an experiment that considers each configuration separately, possibly leading to different sensor locations for each configuration. In this experiment, we compare the number of sensors and the locations of the sensors with those in the previous experiment. As Table V shows, configurations 1 and 2 require only two sensors while configuration 3 requires only one sensor as opposed to three-sensor solution obtained from the multiple-configuration problem. This demonstrates the need to jointly consider network topologies in one problem for such situations.

D. Comparison to a heuristic sensor placement technique

We next demonstrate the performance of the heuristic sensor placement technique described in Section III-E in the context of multiple network configurations for the three-configuration variant of the test case *case33bw*. We study the effects of two versions of this heuristic: (i) place sensors at the end of all branches based on configuration 1 and (ii) place sensors at the ends of branches considering all configurations. The results are shown in Table VI. For the first version of this heuristic, we place four sensors. This technique works well in configuration 1; however, it introduces false negatives (failures to alarm in cases with voltage limit violations) in approximately 10% to 20% of the power injection samples for configurations 2 and 3 since the sensors are instead located in the middle of some branches and thus do not capture all possible violations. To reduce the number of false negatives, we next consider sensor locations based on all configurations, i.e., the second version of the heuristic. This results in six sensors being placed and only one occurrence of a false negative. Comparing to our approach (refer to Table IV), we only need three sensors (as opposed to six sensors) to detect all violations. This shows the necessity of using an optimization formulation to obtain *sparse* sensor placement solutions, since a naive approach where sensors are placed at all buses will always avoid false positives.

V. CONCLUSION

This paper has formulated a bilevel optimization problem that seeks to minimize the number of sensors needed to detect violations of voltage magnitude limits in an electric distribution system. We first addressed the power flow nonlinearities in the lower-level problem via previously developed conservative linear approximations of the power flow equations. Due to computational challenge from the bilevel nature, we therefore exploited structure specific to this problem to obtain single-level mixed-integer programming formulations that avoid introducing unnecessary additional discrete variables. We also demonstrated how to obtain a mixed-integer-linear programming formulation by discretizing the sensor thresholds. Furthermore, we developed extensions to these reformulations that consider the possibility of multiple network topologies. Our proposed sensor placement reformulations require substantially less computation time than

TABLE IV
RESULTS FOR THE *case33bw* SYSTEM WITH THREE NETWORK CONFIGURATIONS.

		Bilinear			**MILP**		
		Config 1	**Config 2**	**Config 3**	**Config 1**	**Config 2**	**Config 3**
Computation time [s]	Optimality	——— 20.1 ———			——— 7.93 ———		
	AGD	0.53	0.55	0.10	0.86	0.93	0.40
Sensor location(s)		——— 8, 14, 26, 33 ———			——— 9, 14, 30 ———		
Sensor thresholds		0.91, 0.919, 0.91, 0.9113	0.9126, 0.91, 0.91, 0.9111	0.91, 0.91, 0.9106, 0.9137	0.91, 0.919, 0.919	0.9185, 0.91, 0.919	0.91, 0.91, 0.9165
with AGD		0.91, 0.9167, 0.91, 0.9104	0.9117, 0.91, 0.91, 0.9109	0.91, 0.91, 0.9101, 0.9136	0.91, 0.9167, 0.9168	0.9166, 0.91, 0.9189	0.91, 0.91, 0.9151
# feasible points		9560	8292	9112	9560	8292	9112
# false positives		619	486	128	919	1238	655
with AGD		186	121	5	287	338	238
# false negatives		0	0	0	0	0	0

TABLE V
RESULTS FOR *case33bw* WITH THREE NETWORK CONFIGURATIONS WHERE SENSOR LOCATIONS ARE NOT NECESSARY THE SAME.

	Config 1	**Config 2**	**Config 3**
Sensor location(s)	14, 30	9, 31	30
Sensor threshold(s)	0.9195 0.9190	0.9185 0.9185	0.9185
with AGD	0.9167 0.9168	0.9164 0.9186	0.9151
# feasible points	9560	8292	9112
# false positives	1048	1245	1453
with AGD	289	306	238
# false negatives	0	0	0

TABLE VI
RESULTS FOR *case33bw* WITH THREE NETWORK CONFIGURATIONS USING TWO VARIANTS OF THE HEURISTIC TECHNIQUE

Case		**Config 1**	**Config 2**	**Config 3**
version 1	Sensor location(s)	——— 18, 22, 25, 33 ———		
	Sensor thresholds	——— 0.91, 0.91, 0.91, 0.91 ———		
	# feasible points	9560	8292	9112
	# false positives	0	0	0
	# false negatives	0	1626	885
version 2	Sensor location(s)	——— 7, 18, 22, 25, 26, 33 ———		
	Sensor thresholds	0.91, 0.91, 0.91, 0.91, 0.91, 0.91		
	# feasible points	9560	8292	9112
	# false positives	0	0	0
	# false negatives	1	0	0

standard reformulation techniques. We also developed a post-processing technique that reduces the number of false alarms via an approximate gradient descent method. The combination of the bilevel problem reformulation and this post-processing technique allows us to compute sensor locations and alarm thresholds that result in few false alarms and no missed alarms, as validated numerically using out-of-sample testing.

In our future work, we seek to identify where the violations occur using the information obtained from CLAs and solutions from the sensor placement problem. Furthermore, we intend to use the sensor locations and thresholds resulting from the proposed formulations to design corrective control actions which ensure that all voltages remain within safety limits.

REFERENCES

[1] A. N. Samudrala, M. H. Amini, S. Kar, and R. S. Blum, "Optimal sensor placement for topology identification in smart power grids," in *53rd Annual Conference on Information Sciences and Systems (CISS)*, 2019.

[2] ——, "Sensor placement for outage identifiability in power distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 1996–2013, 2020.

[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, "Optimal placement of PMUs by integer linear programming," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1525–1526, 2008.

[5] R. J. Albuquerque and V. L. Paucar, "Evaluation of the PMUs measurement channels availability for observability analysis," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2536–2544, 2013.

[6] M. S. Thomas, S. Ranjan, and N. Bhaskar, "Optimization of PMU placement by performing observability analysis," in *IEEE 6th India International Conference on Power Electronics (IICPE)*, 2014.

[7] F. Aminifar, M. Fotuhi-Firuzabad, M. Shahidehpour, and A. Khodaei, "Probabilistic multistage PMU placement in electric power systems," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 841–849, 2011.

[8] H. Mehrjerdi, S. Lefebvre, D. Asber, and M. Saad, "Eliminating voltage violations in power systems using secondary voltage control and decentralized neural network," in *IEEE Power & Energy Society General Meeting*, 2013.

[9] A. M. Nour, A. Y. Hatata, A. A. Helal, and M. M. El-Saadawi, "Review on voltage-violation mitigation techniques of distribution networks with distributed rooftop PV systems," *IET Generation, Transmission & Distribution*, vol. 14, no. 3, pp. 349–361, 2020.

[10] D. K. Mohanta, C. Murthy, and D. S. Roy, "A brief review of phasor measurement units as sensors for smart grid," *Electric Power Components and Systems*, vol. 44, no. 4, pp. 411–425, 2016.

[11] P. Buason, S. Misra, and D. K. Molzahn, "A sample-based approach for computing conservative linear power flow approximations," *Electric Power Systems Research*, vol. 212, p. 108579, 2022, presented at the *22nd Power Systems Computation Conference (PSCC 2022)*.

[12] U.-P. Wen and S.-T. Hsu, "Linear bi-level programming problems – A review," *The Journal of the Operational Research Society*, vol. 42, no. 2, pp. 125–133, 1991.

[13] S. Dempe and A. Zemkoho, "On the Karush–Kuhn–Tucker reformulation of the bilevel optimization problem," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 75, no. 3, pp. 1202–1218, 2012.

[14] S. Pineda and J. M. Morales, "Solving linear bilevel problems using big-Ms: Not all that glitters is gold," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2469–2471, 2019.

[15] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I–Convex underestimating problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.

[16] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[17] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *IEEE International Symposium on Computer Aided Control Systems Design (CACSD)*, September 2004, pp. 284–289.