To appear in Networks

The Power Edge Set Problem*

Pierre-Louis Poirion Sonia Toubaline Claudia D'Ambrosio
Leo Liberti
CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France
{poirion,toubaline,dambrosio,liberti}@lix.polytechnique.fr

Abstract

The automated real time control of an electrical network is achieved through the estimation of its state using phasor measurement units. Given an undirected graph representing the network, we study the problem of finding the minimum number of phasor measurement units to place on the edges such that the graph is fully observed. This problem is also known as the Power Edge Set problem, a variant of the Power Dominating Set problem. It is naturally modelled using an iteration-indexed binary linear program, whose size turns out to be too large for practical purposes. We use a fixed-point argument to remove the iteration indices and obtain a more compact bilevel formulation. We then reformulate the latter to a single-level mixed-integer linear program, which performs better than the natural formulation. Lastly, we provide an algorithm that solves the bilevel program directly and much faster than a commercial solver can solve the previous models. We also discuss robust variants and extensions of the problem.
Keywords: Real time electrical network monitoring, Bilevel program, Mixed-integer linear program, PMU placement problem, Observability, Power Dominating Set.

1 Introduction

An electrical network is represented by an undirected graph where the edges are the transmission lines, and the nodes are

(i) their junctions, called *buses* or *zero injection* nodes;

(ii) the points of energy consumption, called *loads*;

(iii) the points of energy production, called *generators*.

The *state* of the network is defined to be the assignment of electrical current values to lines, of voltage values to buses, and of power values to loads and generators. One way to make the network "smart" is to automate the control of its state in real time, which requires continuous monitoring.

Monitoring the state of an electrical network can be done using control and measurement devices such as Phasor Measurement Units (PMUs). A PMU is a monitoring device designed to be placed at a certain bus in order to provide to some system monitoring centre:

- a synchronized phasor measurement (a phasor is a complex number that represents both the magnitude and phase angle of the sine waves in electrical signals)

*This work was carried out as part of the SO-grid project (www.so-grid.com), co-funded by the French agency for Environment and Energy Management (ADEME) and developed in collaboration between participating academic and industrial partners. A very preliminary version of the results presented in this paper has been accepted for publication [21].

- the branch current phasor of all its outgoing transmission lines [15].

PMUs are synchronized via GPS and send large bursts of data to the system monitoring centre. Due to their relatively high cost, their optimal placement constitutes an important challenge. The present study is motivated by the French smart grid prototype being constructed under the auspices of the SO-grid project `www.so-grid.com`.

The problem of finding the optimal placement of PMUs, called the PMU PLACEMENT problem (PMUP), consists in the determination of the minimum number of PMUs to place on the nodes of the graph so as to ensure its full observability. A node (resp. an edge) is said to be *observed* if its voltage (resp. its current) is known. A graph is then said to be *fully observed* if all the node voltages and edge currents are known. In [5], Brueni and Heath define the observability of a graph using two rules:

(i) if a PMU is installed on a node, then this node and all its neighbours are observed;

(ii) if all the neighbours of an observed node are known to be observed except one, then this latter is also observed.

In other words, deciding whether a given node or edge is observed implies a "propagation" of observability through the repeated application of these rules starting from the set of PMUs placed on the graph (see Fig. 1). We want to make it absolutely clear that observability is *not* propagated in time, namely: either a node/edge is observed, or it is not. What is propagated is simply the decision logic embedded in the above rules. As such, we are *not* saying that nodes or edges are initially unobserved and will become observed later in time. All we are saying is that, from an initial set of placed PMUs, the decision of whether any node or edge is observed can be made by iteratively applying the rules above until the set of observed nodes/edges no longer changes.

In this paper, we study the optimal placement of PMUs which can only observe the state of one incident transmission line on networks without loads and generators. This PMUP variant turns out to be **NP**-complete [20] by reduction from 3-REGULAR VERTEX COVER.

1.1 Literature review

Various solution methods have been proposed to solve the PMUP [15, 16]. To the best of our knowledge, all of the proposed approaches so far use observability rule (ii) only locally, i.e., limited to given propagation depths. A possible reason for this is that it makes optimization models smaller, and hence easier to solve, by limiting the propagation steps (at the cost of sub-optimality). In this paper we take a different approach: we apply rule (ii) globally, but replace it by its fixed-point condition, which vastly reduces the number of decision variables. Limited to network models involving only zero injection nodes, the PMUP is also known as the POWER DOMINATING SET (PDS) problem [11].

The PDS has been extensively studied in the literature: for example, it is **NP**-complete even for bipartite and chordal graphs, but polynomial for trees [11]; it is **NP**-complete for planar bipartite graphs, but polynomial for grids [7]; and there is an $O(\sqrt{n})$-approximation algorithm for planar graphs of n vertices, but it is **NP**-hard to approximate (on general graphs) within a factor $2^{\log^{1-\epsilon} n}$ [1]. By inclusion from the PDS, the PMUP is also **NP**-complete.

In practice, the observability rules above are not the only way to find information about the state of the network. For example, it is easier to estimate the state of nodes that are adjacent to (or not too far from) other observed nodes. By carefully leveraging a reduced number of observed nodes, it makes sense to propose fairly efficient relaxation-based heuristics which, however, often rely on the structure of the problem instance being considered rather than addressing the whole problem in general. Examples of this approach are [18], which proposes spanning tree search techniques; [6], which is based on solving an Integer Linear Program (ILP); [23], which exploits a nonlinear Integer Programming (IP) formulation including conventional power flow and injection measures (conventional measures are those that are provided by non-synchronized sensors) in addition to PMUs. Moreover, [23] also allows the determination of PMU locations that are guaranteed to be optimal: specifically, the observability rule (ii) is applied to zero injection nodes with no limited depth. Some work has been carried out on robust variants of the PMUP: for example, single line outages and single PMU failures have been addressed by means of some IP formulations discussed in [3].

An inherent hardware limitation of many PMU devices, namely the number ℓ of incident transmission lines (or *channels*) that PMUs can monitor, gives rise to a different set of restrictions of the PMUP, which we call ℓ-PMUP (for each ℓ). We remark that the instances of the ℓ-PMUP and the PMUP are the same, namely the graph representing the electrical network. The solution of the ℓ-PMUP is a set of couples $(v, \{e_1, \ldots, e_\ell\})$ such that we place PMUs at each node v monitoring the channels $e_1, \ldots, e_\ell$ adjacent to v. We also remark that, differently from the PMUP, a single node can host multiple PMUs in the solution of ℓ-PMUP instances. A Binary Linear Program (BLP), which considers all the possible combinations of ℓ channels incident to each node, is given in [12]. Another method based on node connectivity and edge selectivity matrices, where the number of channels is less than the minimum degree of the graph, is proposed in [13].

1.2 The problem studied in this paper

As mentioned above, this paper is concerned with the special case of PMUs that can observe at most one channel, i.e., $\ell = 1$, on networks that only involve zero injection nodes. Note that when $\ell = 1$, though PMUs are still actually placed at nodes and monitor a single adjacent channel, the output is equivalent to a placement on the edges. To see this, it is sufficient to notice that given an edge placement we can arbitrarily decide, for each edge, which of the two adjacent nodes hosts the PMU, since the solution cardinality is the same. Because it concerns a selection of edges, we call this the Power Edge Set (PES) problem. To the best of our knowledge, this is the first study concerning this problem variant.

In the PES problem, the first rule of observability must be re-written as: *if a PMU is installed on an edge, then both its adjacent nodes are observed* (see Fig. 1). While the PES problem is **NP**-hard, we note

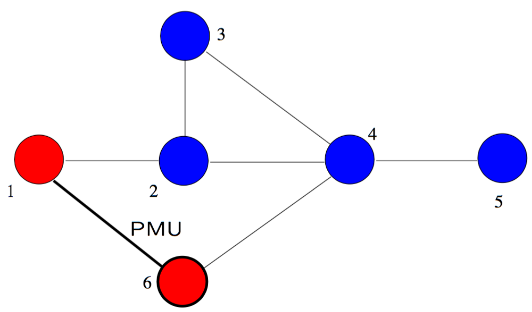

Figure 1: Observability propagation based on Rules (i)-(ii). The nodes adjacent to the PMU placed on edge $\{1, 6\}$ are observed by rule (i). The other nodes are observed by repeated application of rule (ii), e.g., node 2 is observed because 1 is observed, its star consists of 2 and 6, and 2 is the only yet unobserved node in the star.

in passing that if we relax rule (ii), the corresponding problem is equivalent to minimum edge cover [8], which can be solved in polynomial time. A localized version of the second observability rule which only considers a given depth (similar to [6]) is discussed for zero injection nodes in [9], together with some variants that are robust to line outages.

1.3 Contents of this paper

The rest of this paper is divided into two parts:

- In Section 2 we define the PES problem formally. We present a natural BLP model for the PES problem in Section 2.2, which explicitly makes use of variables indexed on the iteration of the observability propagation process based on rules (i)-(ii). We then model the fixed-point conditions (Section 2.3) of the propagation process as constraints and propose in Section 2.3.2 a bilevel BLP model with fewer decision variables. In Section 2.4, we further reformulate the latter to a Mixed-Integer Linear Program (MILP) with binary variables. In Section 2.5, we propose a cutting-plane algorithm to solve the bilevel BLP formulation directly and improve the convergence speed of this algorithm by means of a lower bound on the optimal objective function value (Section 2.6). In Section 2.7 we integrate conventional measures into our models. We discuss robust variants of the PES problem to protect against a single line outage in Section 2.8 and against a single measurement loss in Section 2.9. Computational results are presented in Section 2.10.
- In Section 3, we consider a generalization of the PES problem, called the GENERALIZED PES (GPES) problem, where we apply the second observability rule to a cut instead of a star. We adapt in Section 3.2 the bilevel modelproposed for the PES problem to solve the GPES problem and propose in Section 3.3 a polynomial-time algorithm to compute the observability propagation. Computational results are discussed in Section 3.4.

2 The Power Edge Set problem

Let $G = (V, E)$ be a graph modelling the electrical network where $V = \{1, \dots, n\}$ is the set of nodes representing the buses and E is the set of edges corresponding to channels. An edge linking nodes i and j is denoted $\{i, j\}$. For $i \in V$, $\Gamma(i) = \{j \mid \{i, j\} \in E\}$ is the set of neighbours (adjacent nodes) of i. For graph-theoretical notions, see [10, 22].

In this paper, we are interested in the optimal placement of PMUs with one channel, so as to ensure a full observability of G. We first consider that no conventional measures exist, which reduces the number of PMUs to install. We show how to take them into account in the model in Section 2.7.

A PMU is placed on an edge $\{i, j\}$ close to node i, for $i \in V$ and $\{i, j\} \in E$. The fact that PMU placement occurs closer to one of the adjacent node is relevant for physical reasons, but irrelevant for our abstract modelling purposes. Henceforth, we shall simply assume that placement occurs on an edge $\{i, j\} \in E$. A graph is said to be observable if all node voltages and current edges are known either measured by a PMU or estimated using electrical laws. The two electrical laws we use are Ohm's law and Kirchhoff 's current law. Ohm's law states that for a given channel $\{u, v\}$,

$$\Delta\mathbf{V}(u, v) = R\,\mathbf{I}(u, v),$$

where

- $\Delta\mathbf{V}(u, v) = \mathbf{V}(u) - \mathbf{V}(v)$;
- $\mathbf{V}(u)$ and $\mathbf{V}(v)$ are the voltages at nodes u and $v \in V$ respectively;
- $\mathbf{I}(u, v)$ is the current flowing between u and v;
- R is the resistance of the channel $\{u, v\}$.

Kirchhoff's Current Law (KCL) applied at a given node $v \in V$ states that the sum of all currents on edges incident to v is zero. If all the currents are known, save one, then this latter can be computed.

The problem of interest is formally defined as follows:

> Power Edge Set (PES) problem. Given a graph $G = (V, E)$, find an edge subset $\Pi \subseteq E$ of minimum cardinality such that a placement of PMUs on edges in Π yields the full observation of G.

We describe in the following the mathematical models and solution methods we propose to solve the PMU placement problem with and without conventional measures. We also consider the case of a single line outage and a single PMU failure.

2.1 Observability rules

Let Π be a given PMU placement on E and let Ω be the set of observed nodes in V. The observability propagation rules (i)-(ii) above can be formally reformulated in the PES setting as follows.

R1: If a PMU is placed on an edge $\{i, j\}$, then nodes i and j are observed:

$$\{i, j\} \in \Pi \Rightarrow i, j \in \Omega.$$

R2: If an observed node i has all its neighbour nodes observed, except one, then this latter node is observed :

$$i \in \Omega \ \wedge \ |\Gamma(i) \smallsetminus \Omega| \leq 1 \Rightarrow \Gamma(i) \subseteq \Omega.$$

By rule **R1**, the PMU placed at $\{i, j\}$ measures $\mathbf{V}(i)$ and $\mathbf{I}(i, j)$. Using Ohm's law, we can deduce $\mathbf{V}(j)$. Then i and j are both observed. By rule **R2**, if a node i and all its neighbours $k \in \Gamma(i) \setminus \{j\}$ are observed, except a single node j, then using Ohm's law we can determine $\mathbf{I}(i, k)$ for $k \in \Gamma(i) \setminus \{j\}$; knowing all $\mathbf{I}(i, k)$ we can deduce $\mathbf{I}(i, j)$ using KCL. Then, knowing $\mathbf{V}(i)$ and $\mathbf{I}(i, j)$, we determine $\mathbf{V}(j)$ using Ohm's law. Hence, j is observed.

We remark that these rules define an "observability propagation" dynamics, strictly in an algorithmic sense. Moreover, the rules imply one further "minimality" rule, namely that no other nodes are observed aside from those which can be observed with rules **R1** and **R2**.

With the notation Π and Ω in place, the PES problem consists in finding $\min |\Pi|$ such that $\Omega = V$, where **R1** and **R2** represent the link between Π and Ω.

2.2 Iteration-indexed binary linear programming

In this section we describe a "natural" Mathematical Programming (MP) formulation for the PES problem. This formulation is not particularly computationally efficient, as we shall see in Section 2.10, and we only use it as a starting point for a reformulation that will lead to a bilevel formulation (see Section 2.3). Our motivation in explicitly stating this MP formulation is to actually showcase the reformulation process, which we believe has some degree of generality.

MP is a formal language for describing optimization problems, and its main purpose is that, by providing many different "generic solvers" for solving MP formulations with varying mathematical properties, shifts the focus from "solving" to "modelling" a problem, which can be achieved faster than inventing a solution algorithm. An MP formulation consists of parameters (which encode the instance), decision variables (which encode the solution), an objective function, and constraints. Modelling a logical process is a way of transforming an informal natural language description of the process into a formal one, given in the language of MP. Moreover, many possible formalizations are possible for any given informal problem description. In the vast majority of cases, the first modelling effort produces what is often called a *natural formulation*, meaning that the decision variables are linked to concepts referred to in the natural language description of the problem. Natural formulations are often reformulated to achieve improved computational efficiency of the solvers. Unlike modelling, reformulation methods are entirely formal. Researchers in MP are usually interested in reformulations that can be applied to some degree of generality, which we believe is the case here.

The iteration-indexed formulation we present here is based on the iterative process used to determine the observability of the nodes of G and given by rules **R1** and **R2**. Assuming the problem instance to be a feasible one, Ω can be found in at most $n-1$ iterations. This worst-case bound is given by disregarding **R2**: we choose an arbitrary order on the nodes, and then we place a PMU on an edge incident to each vertex in the order; but since the first edge allows the observation of two vertices, we see that $n-1$ PMUs are needed at most. A better worst-case bound is given by the observation that, without **R2**, the problem can be solved in polynomial-time by finding a minimum sized edge cover: so the number of iterations is at most equal to the size of a minimum edge cover. Accordingly, we set ι to either of these upper bounds.

The parameters of the formulation are: the network graph $G = (V, E)$, given as the set V and the set of neighbourhoods $\Gamma(i)$ for each $i \in V$, and the set $I = \{0, 1, \dots, \iota\}$ of iteration indices, where ι is a worst-case bound on the maximum number of iterations. We also let $I' = I \smallsetminus \{\iota\}$. The decision variable tensors have the following components:

(i) s_{ij}, a binary variable equal to one if and only if we place a PMU on the edge $\{i, j\} \in E$, for each $i \in V$ and $j \in \Gamma(i)$;

(ii) ω_{id}, a binary variable equal to one if and only if the propagation rules prove that i is in the observed set Ω at iteration $d \in I$;

(iii) y_{ijd}, a binary variable equal to one if and only if rule **R2** is used at iteration $d \in I$ to observe node $j \in \Gamma(i)$ using node $i \in V$ as already observed.

Here follows the formulation:

$$
(P_{\mathsf{IT}}) \left\{
\begin{array}{rll}
\min\limits_{s,\omega,y} & \sum\limits_{i\in V} \sum\limits_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} & (1)\\
\forall i \in V, j \in \Gamma(i) & s_{ij} = s_{ji} & (2)\\
\forall i \in V & \omega_{i\iota} = 1 & (3)\\
\forall i \in V & \sum\limits_{j\in\Gamma(i)} s_{ij} \ge \omega_{i0} & (4)\\
\forall j \in V, d \in I' & \omega_{j,d+1} - \omega_{jd} \le \sum\limits_{\substack{i\in V\\ j\in\Gamma(i)}} y_{ijd} & (5)\\
\forall i \in V, j \in \Gamma(i), d \in I' & \omega_{j,d+1} - \omega_{jd} \le \omega_{id} - y_{ijd} + 1 & (6)\\
\forall i \in V, j, k \in \Gamma(i) \smallsetminus \{j\}, d \in I' & \omega_{j,d+1} - \omega_{jd} \le \omega_{kd} - y_{ijd} + 1 & (7)\\
\forall i \in V, d \in I' & \omega_{id} \le \omega_{i,d+1} & (8)\\
\forall i \in V, j \in \Gamma(i) & s_{ij} \in \{0,1\} & (9)\\
\forall i \in V, d \in I & \omega_{id} \in \{0,1\} & (10)\\
\forall i \in V, j \in \Gamma(i), d \in I & y_{ijd} \in \{0,1\}. & (11)
\end{array}
\right.
$$

The objective function Eq. (1) aims at minimizing the number of installed PMUs. Constraints (2) enforce the undirectedness of the graph. Constraints (3) state that every node must be observed by the last iteration. Constraints (4) essentially state that **R1** is enforced at the zero-th iteration: if a node is observed at the outset, it is because there must be a PMU installed at some incident edge. Constraints (5)-(7) describe the dynamics of rule **R2**: the left-hand sides can only be zero, one, or minus one; the latter possibility is excluded by Constraints (8). If the left-hand sides are zero, it is because the node in question has not entered Ω at iteration d; and, as a consequence, since the right-hand sides are always non-negative, the constraints are inactive. The only way these constraints can be active is if a node enters Ω at iteration d, in which case the left-hand side has value one. Each of these left-hand sides forces a one in the right-hand sides, which means that the right-hand size condition is true: in Constraints (5), if node j is observed at iteration d, it must have been in consequence of rule **R2** being applied with some $i \in V$ as the center of the neighbourhood to which j belongs; Constraints (6) state that i is already observed at iteration d, and Constraints (7) that every other node k in the neighbourhood, aside from j, is also already observed. Constraints (8) state monotonicity of observability: once observed, a node stays observed (in other words, nodes can never be removed from Ω). Constraints (9)-(11) are the variable domains.

Many discrete dynamical processes have been previously formulated using MP and iteration indices. The authors of [19], for example, formulated the problem of minimizing the number of agents placed in the nodes of a given graph in order to clean node contamination in a similar way. The propagation dynamics described in [19], however, are non-monotonic both in terms of moving agents and contamination evolution, whereas our observability propagation is monotonic. This allows us to propose a fixed-point reformulation as detailed in Section 2.3, which would be impossible for the problem described in [19].

2.3 Bilevel programming via fixed-point reformulation

As shown in Table 1, the iteration-indexed formulation can only be used in practice for rather small instances. We proceed to reformulate it using an innovative fixed-point technique which, though not completely general, can be applied in many other settings, given a monotonic propagation algorithm on a graph.

For any $d \in I$, let $\omega^d = (\omega_{id} \mid i \in V)$ be the characteristic vector describing the observability of the nodes at iteration d. Formulation (P_{IT}) computes the vector values for $d \in \{1, \dots, \iota\}$. With a small abuse of notation, we denote by $\omega = (\omega_i \mid i \in V)$ the characteristic vector of Ω (i.e., $\omega_i = 1$ if and only if $i \in \Omega$). We remark that $\omega_i = 1$ if and only if i enters Ω at some iteration d, which implies that ω_i is the projection of the tensor ω^d along the dimension denoted d. In particular, by the monotonicity constraints (8), $\omega = \omega^\iota$. This suggests that the tensor ω^d actually carries "too much information", in the sense that only its slice ω^ι matters. In the following, we exploit this feature to derive a non-iterative formulation, where the observability variables ω only depend on $i \in V$.

For $d \in I$ and a node in $j \in V$, $\omega_{j,d+1}$ can be expressed as a function of ω_{jd} as follows:

$$\omega_{j,d+1} = \max\left(\omega_{jd}, \max_{\substack{i \in V \\ j \in \Gamma(i)}} \left(1 - |\Gamma(i)| + \omega_{id} + \sum_{k \in \Gamma(i) \smallsetminus \{j\}} \omega_{kd}\right)\right). \quad (12)$$

In other words, j is observed at iteration $d+1$ either if it was already observed at iteration d or there exists a neighbourhood $N(i)$ of a node $i \in V$ such that all the other neighbours $k \neq j$ of i are observed at iteration d. Note that Eq. (12) is a recursive relation on ω that defines the observability propagation dynamics embedded in rule **R2**. For later reference, we remark that the minimality rule referred to in Section 2.1 also holds here and is implicitly enforced by the equality relation in Eq. (12).

We represent the right-hand side of Eq. (12) by means of a function $\theta : \{0,1\}^n \to \{0,1\}^n$:

$$\forall j \in V \quad \theta_j(x) = \max\left(x_j, \max_{\substack{i \in V \\ j \in \Gamma(i)}} \left(1 - |\Gamma(i)| + x_i + \sum_{k \in \Gamma(i) \smallsetminus \{j\}} x_k\right)\right),$$

where $x = (x_j \mid j \in V)$. We extend this definition to $\theta(x) = (\theta_j(x) \mid j \in V)$, so that we can summarize the observability propagation dynamics to $\forall d \in I' \quad \omega^{d+1} = \theta(\omega^d)$.

Recall rule **R1**: given $\Pi \subseteq E$, Ω contains all nodes adjacent to edges in Π. This can be formalized as follows: given the support vector $s \in \{0,1\}^m$ of Π,

$$\forall \{i,j\} \in \Pi \quad \omega_{j0} \geq s_{ij}.$$

Now we can write the application of both **R1** and **R2** to Π as follows:

$$\forall j \in V, d \in I' \quad \omega_{j,d+1} = \max\left(\bigvee_{i \in \Gamma(j)} s_{ij}\,,\, \theta_j(\omega^d) \right), \tag{13}$$

which we can naturally extend to a fixed-point equation for the whole vector ω^{d+1}:

$$\forall d \in I' \quad \omega^{d+1} = \vartheta_s(\omega^d), \tag{14}$$

where ϑ_s is an n-vector function, each component of which is given by the right-hand side of Eq. (13).

2.3.1 A mathematical program for ω

With s, ω and ϑ_s defined as above, we have the following result.

Theorem 2.1 *There exists $\eta \in \mathbb{N}$ such that ω^η is the unique least fixed-point of ϑ_s. Moreover, ω^η is the support vector of Ω.*

Proof. By definition, for each $d \in \mathbb{N}$, ω^d is the support vector of the nodes that are observed after d iterations of the observability propagation algorithm. The zero-th iteration of this algorithm consists in applying rule **R1** to s to yield an initial set Ω^0 of observed nodes, encoded by a support vector ω^0. At iteration $d+1$, the algorithm applies rule **R2** to ω^d. By monotonicity, $|\Omega^d|$ can never decrease. Moreover, since $\Omega^d \subseteq V$ and V is finite, this algorithm terminates. Let η be its last iteration. This means that $\omega^{\eta+h} = \omega^\eta$ for each $h \in \mathbb{N}$, so $\omega^\eta = \omega^{\eta+1} = \vartheta_s(\omega^\eta)$, which implies that ω^η is a fixed-point of ϑ_s.

Now suppose there is another fixed-point y of ϑ_s such that the cardinality of the support of y is smaller than that of ω^η. Then there must be an iteration β such that $y = \omega^{\beta+1} = \vartheta_s(\omega^\beta) = y$. By monotonicity, we must have $\beta < \eta$, contradicting the fact that η is the last iteration. Uniqueness of the least fixed point follows because the algorithm is deterministic. Now recall that the support vector of Ω is denoted by ω. Since $\omega = \omega^\iota$ as observed in Section 2.3 and $\omega^{\eta+h} = \omega^\eta$ for each $h \in \mathbb{N}$, it follows that $\omega = \omega^\eta$. □

Next, define an MILP for determining ω given s, independently of the iteration index, by describing the least fixed-point of ϑ_s in Eq. (14):

$$(P_{\mathsf{LL}}) \begin{cases} \min\limits_{\omega \in \{0,1\}^n} \quad \sum\limits_{j \in V} \omega_j & (15)\\ \forall i \in V, j \in \Gamma(i) \qquad \omega_j \geq s_{ij} & (16)\\ \forall i \in V, j \in \Gamma(i) \qquad \omega_j \geq 1 - |\Gamma(i)| + \omega_i + \sum\limits_{k \in \Gamma(i) \smallsetminus \{j\}} \omega_k. & (17) \end{cases}$$

Since (P_{LL}) links Ω to Π through their support vectors s and ω, we denote the observed node set Ω as $\Omega(s)$.

2.3.2 The bilevel formulation

The PES problem is modelled by the bilevel programming formulation described below:

$$
(P_{\mathsf{PES}}) \left\{ \begin{array}{ll}
\min\limits_{s\in\{0,1\}^m} & \sum\limits_{i\in V} \sum\limits_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} \\
\forall i\in V, j\in\Gamma(i) & s_{ij} = s_{ji} \\
 & n \le \left\{ \begin{array}{lll}
 \min\limits_{\omega\in\{0,1\}^n} \sum\limits_{j\in V} \omega_j & & \\
 \forall i\in V, j\in\Gamma(i)\ \omega_j & \ge & s_{ij} \\
 \forall i\in V, j\in\Gamma(i)\ \omega_j & \ge & 1-|\Gamma(i)|+\omega_i + \sum\limits_{k\in\Gamma(i)\smallsetminus\{j\}} \omega_k
 \end{array}\right.
\end{array}\right.
$$

In the upper-level problem the objective is to minimize the number of PMUs to install such that the number of observed nodes given by function $\Omega(s)$ is at least n. The lower-level problem encodes the fixed-point of Eq. (14), which is the support vector of $\Omega(s)$ as shown in Thm. 2.1. More precisely, the constraints on the lower level states that the optimum of the lower-level problem should be at least as large as n, which is equivalent to requiring that the propagation dynamics ensures observability of all nodes.

2.4 MILP reformulation

We show in this section that the bilevel program (P_{PES}) can be reformulated exactly as an MILP. We first prove that the integrality of the binary decision variables ω can be relaxed in the lower-level subproblem.

Lemma 2.2 *If the constraints $\omega \in \{0,1\}^n$ are replaced by $\omega \ge 0$ in the lower-level subproblem, there is at least one optimal solution to the relaxed lower-level subproblem which is binary.*

Proof. We need to show that, for each $s \in \{0,1\}^m$, the optimal solution of the lower-level subproblem is equal to that of its relaxation obtained by replacing $\omega \in \{0,1\}^n$ by $\omega \ge 0$. So let $s \in \{0,1\}^m$ and let $\bar\omega$ be an optimal solution of the relaxed lower-level subproblem (P_{LL}). By Constraints (16), there is a set S of nodes, namely those which are adjacent to edges $\{i,j\} \in \Pi$ such that $s_{ij} = 1$, such that $\bar\omega_j = 1$ for all $j \in S$. Now let $U \subseteq V$ be the set of all those node indices such that $\bar\omega_\ell \in (0,1)$, and suppose $Z \neq \varnothing$: it is impossible for any $\ell \in U$ to be incident to an edge of Π, since otherwise by Constraints (16) we would have $\bar\omega_\ell = 1$. By the constraint sense, we can set $\bar\omega_\ell = 0$ for each $\ell \in U$ and still satisfy the Constraints (17) (this can be seen by induction on the node indices occurring in ρ), as claimed. A similar argument applies in the case $\bar\omega_\ell > 0$. □

This means that the lower-level subproblem can be relaxed to an LP. The classical theory of bilevel programming states that any bilevel program involving an LP subproblem can be reformulated exactly to a single-level problem by replacing the lower-level subproblem by its Karush-Kuhn-Tucker (KKT) conditions [4]. Hence we obtain the Mixed-Integer Nonlinear Program (MINLP):

$$
(P_{\mathsf{MINLP}}) \begin{cases}
\min_{s,\lambda,\mu} \sum_{i\in V} \sum_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} & (18)\\
\forall i \in V, j \in \Gamma(i) \quad s_{ij} = s_{ji} & (19)\\
\sum_{i\in V} \sum_{j\in\Gamma(i)} s_{ij}\mu_{ij} + (1 - |\Gamma(j)|)\lambda_{ij} \geq n & (20)\\
\forall i \in V \quad \sum_{j\in\Gamma(i)} (\mu_{ij} + \lambda_{ij} - \lambda_{ji} - \sum_{\substack{k\in\Gamma(j)\\ k\neq i}} \lambda_{kj}) \leq 1 & (21)\\
s \in \{0,1\}^m, \lambda, \mu \geq 0. & (22)
\end{cases}
$$

We now prove that, $\forall i \in V, j \in \Gamma(i)$, the dual variables μ_{ij} are bounded.

Proposition 2.3 *There exists a constant $M > 0$ such that $\mu_{ij} \leq M$ for each $i \in V$ and $j \in \Gamma(i)$.*

Proof. Let (s^*, μ^*, λ^*) be an optimal solution of (P_{MINLP}) and let (s^*, ω^*) be the corresponding optimal solution of the bilevel formulation (P_{PES}). In particular, we consider (s^*, μ^*, λ^*) such that (μ^*, λ^*) is a basis solution of the dual LP which defines $|\Omega(s)|$, that is:

$$
\begin{cases}
\max_{\lambda,\mu\geq 0} \quad \sum_{i\in V} \sum_{j\in\Gamma(i)} s_{ij}\mu_{ij} + (1 - |\Gamma(j)|)\lambda_{ij}\\
\forall i \in V \quad \sum_{j\in\Gamma(i)} (\mu_{ij} + \lambda_{ij} - \lambda_{ji} - \sum_{k\in\Gamma(j),k\neq i} \lambda_{kj}) \leq 1
\end{cases}
$$

Necessarily at most n dual variables are non-zero. Let $I = \{(i,j) \mid \mu_{ij} \neq 0\}$ and $J = \{(i,j) \mid \lambda_{ij} \neq 0\}$. We have $|I| + |J| \leq n$. Let $i \leq n$ such that $\omega_i^* = 1$. By complementary slackness we have

$$
\sum_{j\in\Gamma(i)} (\mu^*_{ij} + \lambda^*_{ij} - \lambda^*_{ji} - \sum_{\substack{k\in\Gamma(j)\\ k\neq i}} \lambda^*_{kj}) = 1. \tag{23}
$$

Let $A_B \in \mathbb{R}^{n\times n}$ be the basis matrix corresponding to the optimal solution (μ^*, λ^*). By Eq. (23), $X = (\mu^*, \lambda^*, \beta^*)$ is a solution of the system $A_B\, X = \mathbf{1}$, where β^* denotes the slack variables used to write the above dual program in standard form, $\mathbf{1}$ is a vector in $\mathbb{R}^n$ where each component is one, and all elements of A_B are in $\{-1, 0, 1\}$. Since $A_B^{-1} = \frac{1}{\det(A_B)} \mathsf{adj}(A_B)$, where $\mathsf{adj}(A_B)$ is the adjoint matrix of A_B and $\det(A_B)$ is the determinant of A_B, using the Hadamard inequality for the determinant, we obtain that the dual variables μ_{ij} are all bounded by $M = n^{\frac{n}{2}}$, for each $i \in V$ and $j \in \Gamma(i)$. □

By Proposition 2.3, we can linearize (P_{MINLP}) exactly by replacing the variable products by $p_{ij} = s_{ij}\, \mu_{ij}$ for all $i \in V$ and $j \in \Gamma(i)$, and then relaxing these equations by using McCormick's convex envelopes [17]:

$$
\begin{aligned}
p_{ij} &\leq \min(\mu_{ij}, M\, s_{ij})\\
p_{ij} &\geq \max(0, \mu_{ij} - M(1 - s_{ij})),
\end{aligned}
$$

for each $i \in V$ and $j \in \Gamma(i)$.

Finally, we obtain the single-level MILP (P_{MILP}) below, without iteration indices.

$$
(P_{\mathsf{MILP}}) \quad \left\{ \begin{array}{lrcl}
\min\limits_{s,p,\lambda,\mu} & \sum\limits_{i\in V} \sum\limits_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} & & \\
\forall i \in V, j \in \Gamma(i) & s_{ij} & = & s_{ji} \\
 & \sum\limits_{i\in V} \sum\limits_{j\in\Gamma(i)} p_{ij} + (1 - |\Gamma(j)|)\lambda_{ij} & \geq & n \\
\forall i \in V & \sum\limits_{j\in\Gamma(i)} (\mu_{ij} + \lambda_{ij} - \lambda_{ji} - \sum\limits_{\substack{k\in\Gamma(j)\\ k\neq i}} \lambda_{kj}) & \leq & 1 \\
\forall i \in V, j \in \Gamma(i) & p_{ij} & \leq & M\, s_{ij} \\
\forall i \in V, j \in \Gamma(i) & p_{ij} & \leq & \mu_{ij} \\
\forall i \in V, j \in \Gamma(i) & p_{ij} & \geq & \mu_{ij} - M(1 - s_{ij}) \\
 & s \in \{0,1\}^m, \quad \lambda, \mu, p & \geq & 0.
\end{array} \right.
$$

Computationally, solving this formulation with off-the-shelf solvers yields solutions much faster than with (P_{IT}). On the other hand, neither formulation is sufficient to solve the large scale instances occurring in the SO-grid project. This motivates us to investigate an algorithm for solving (P_{PES}) directly.

2.5 Cutting-plane algorithm for the bilevel problem

As mentioned above, the two MP formulations we derived above can hardly be used for solving large-scale PES instances. Another motivation for seeking an alternative method is that Lemma 2.2, which is necessary for reformulating the bilevel formulation to a single-level one, fails to hold in general if one were to add constraints to the lower-level subproblem — something which we shall need to do in order to model features such as conventional measures, robustness, and so on.

We propose a cutting plane algorithm, called BilevelSolve, to solve the bilevel program (P_{PES}). BilevelSolve iteratively solves a modified version of the upper-level problem as a *master problem*, adding a new cut at each iteration. The cuts are generated by means of the combinatorial procedure GenerateCut on the lower level *slave problem*.

Let us consider the feasible region of the upper level:

$$\mathcal{F} = \{s \in \{0,1\}^m \mid |\Omega(s)| \geq n\}.$$

Since the variables of the upper-level subproblem are binary, we can rewrite the bilevel program as

$$\min\{ \sum_{\{i,j\}\in E} s_{ij} \mid s \in \mathsf{conv}(\mathcal{F}) \cap \{0,1\}^m\},$$

where $m = |E|$. We now look for a polyhedron $\mathcal{P}$ such that $\mathcal{P} \cap \{0,1\}^m = \mathcal{F}$ and then solve the following problem

$$\min\{ \sum_{\{i,j\}\in E} s_{ij} \mid s \in \mathcal{P} \cap \{0,1\}^m\}.$$

Let $\bar{\mathcal{F}} = \{0,1\}^m \smallsetminus \mathcal{F}$. Notice that for all $s \in \{0,1\}^m$ and $s' \in \bar{\mathcal{F}}$ such that $s < s'$, we have $s \in \bar{\mathcal{F}}$, i.e., if a placement is infeasible and PMUs are removed from this placement then the new placement is also infeasible. For all $s \in \{0,1\}^m$, let $\zeta(s) = \{\{i,j\} \in E \mid s_{ij} = 0\}$.

Lemma 2.4 *For all $\bar{s} \in \bar{\mathcal{F}}$,*

$$\sum_{\{i,j\}\in\zeta(\bar{s})} s_{ij} \geq 1$$

is a valid inequality for $\mathcal{F}$.

Proof. Let $\bar{s} \in \bar{\mathcal{F}}$, and assume that there exists $s \in \mathcal{F}$ such that $\sum_{\{i,j\}\in\zeta(\bar{s})} s_{ij} < 1$. Since $s \in \{0,1\}^m$, we deduce that $\sum_{\{i,j\}\in\zeta(\bar{s})} s_{ij} = 0$, hence $s \leq \bar{s}$ by definition of $\zeta(s)$. Since $\bar{s} \in \bar{\mathcal{F}}$, we have $s \in \bar{\mathcal{F}}$ too, and this contradicts the initial assumption. □

Let $\bar{\mathcal{F}}_{\mathsf{max}}$ be the set of $\leq$-maximal elements in $\bar{\mathcal{F}}$, i.e.,

$$\bar{\mathcal{F}}_{\mathsf{max}} = \{s \in \mathcal{F} \mid \text{If there exists } s' \in \mathcal{F},\ s' \geq s \text{ then } s' = s\}.$$

Let $\mathcal{P} = \{s \in [0,1]^m \mid \forall \bar{s} \in \bar{\mathcal{F}}_{\mathsf{max}} \sum_{e\in\zeta(\bar{s})} s_e \geq 1\}$ and $\mathcal{P}_I = \mathcal{P} \cap \{0,1\}^m$.

Proposition 2.5 *We have $\mathcal{P}_I = \mathcal{F}$.*

Proof. Suppose first that $\mathcal{P}_I \nsubseteq \mathcal{F}$ and let $s' \in \mathcal{P}_I$ such that $s' \in \bar{\mathcal{F}}$. By Lemma 2.4,

$$\sum_{\{i,j\}\in\zeta(s')} s_{ij} \geq 1 \quad (*)$$

is valid for $\mathcal{F}$. The left-hand side of $(*)$ is 0 when $s \leftarrow s'$, so $(*)$ separates s' from $\mathcal{F}$. For any $\bar{s} \geq s'$ with $\bar{s} \in \bar{\mathcal{F}}$ we have $\zeta(\bar{s}) \subseteq \zeta(s')$, so $\sum_{\{i,j\}\in\bar{s}} s_{ij} \geq 1$ dominates $(*)$ and also separates s' from $\mathcal{F}$. In particular, this holds for $\bar{s} \in \bar{\mathcal{F}}_{\mathsf{max}}$, which contradicts the assumption $s' \in \mathcal{P}_I$. Hence $\mathcal{P}_I \subseteq \mathcal{F}$. Let $s \in \mathcal{F}$ and suppose there exists $\bar{s} \in \bar{\mathcal{F}}_{\mathsf{max}}$ such that $\sum_{\{i,j\}\in\zeta(\bar{s})} s_{ij} = 0$, i.e., $s \notin \mathcal{P}$. Then $s \leq \bar{s}$ implies $s \notin \mathcal{F}$, a contradiction: so $s \in \mathcal{P}_I$. □

Proposition 2.5 states that polyhedron $\mathcal{P}$ is entirely described by the points $\bar{s}$ in $\bar{\mathcal{F}}_{\mathsf{max}}$. Therefore at each iteration, either the current solution s belongs to $\mathcal{F}$ and is optimal or $s \notin \mathcal{F}$ and we look for a point $s' \in \bar{\mathcal{F}}_{\mathsf{max}},\ s' \geq s$ to deduce a new cut.

Consider the following MILP P^k:

$$[P^k] \quad \left\{ \begin{array}{lccc} \min\limits_{s\in\{0,1\}^{|E|}} & \sum\limits_{i\in V}\sum\limits_{j\in\Gamma(i)} s_{ij} & & \\ \forall h \leq k & \alpha^h s & \geq & 1, \end{array} \right. \tag{24}$$

where $\alpha^h \in \{0,1\}^{|E|}$ for each $h \leq k$, and k is the main algorithm iteration counter: at iteration k, P^k has k linear covering constraints, starting with $\alpha^1 = (1, \ldots, 1)$.

Algorithm 1: BilevelSolve

```
1: k ← 1
2: termination ← 0
3: while termination = 0 do
4:    s ← MILPSolve(P^k)
5:    k ← k + 1
6:    α^k = GenerateCut(s, termination)
7:    P^k ← [P^{k−1} s.t. α^k s ≥ 1]
8: end while
```

Although BilevelSolve needs exponentially many cuts in the worst case, as we will discuss in Section 2.10, it performs very well empirically.

Algorithm 2: GenerateCut(s, termination)

termination $\leftarrow 0$
// observe nodes according to PMUs in s
place PMUs in G in all edges in the support of s
apply rules **R1** and **R2** to G, to obtain $\Omega \subseteq V$ (observed nodes)
if $\Omega = V$ **then**
  // if PMUs in s suffice to observe all nodes, terminate
  termination $\leftarrow 1$
  $\alpha \leftarrow (0, \ldots, 0)$
else
  // otherwise, apply more PMUs and aim to observe all nodes
  $\Theta \leftarrow \Omega$
  while $\Omega \subsetneq V$ **do**
    choose any $v \in V \smallsetminus \Omega$ and $\{u, v\} \in E$
    place PMU in $\{u, v\}$ and apply **R1**, **R2** to update Ω
    if $\Omega \neq V$ **then**
      $\Theta \leftarrow \Omega$
    end if
  end while
  // generate cut on edges not induced by nodes observed
  *// at **R2** application step before full observability*
  let F be the set of edges induced by Θ
  let α be the support of $E \smallsetminus F$
end if
return (α, termination)

2.6 A lower bound for the PES problem

We present in this section a lower bound L on the value of each globally optimal solution of the PES problem. We claim that the tighter the lower bound is, the faster the constraint generation algorithm will be. The reason for this is that, if such a bound $L > 0$ is known, at the first iteration we can replace the right-hand side of the cut $\alpha^1 s \geq 1$ by L, thereby making the constraint tighter. This makes the feasible region smaller, which is good evidence that we would need to generate fewer cuts in BilevelSolve.

We achieve our estimation for L by using some results on zero forcing sets and the minimum rank of a graph [2].

Definition 2.6 *The minimum rank of a graph $G = (V, E)$, $\mathsf{mr}(G)$, is the minimum rank over all $n \times n$ symmetric matrices S such that $S_{ij} \neq 0$ for each $\{i, j\}$, where $n = |V|$.*

Let A be the adjacency matrix of G and let $\mathsf{rank}(A)$ denotes its rank. By Definition 2.6, we have that $\mathsf{rank}(A) \geq \mathsf{mr}(G)$.

Definition 2.7 *Let $G = (V, E)$ be a graph. A zero forcing set of G is a set $Z \subset V$ of minimum size such that, if these nodes are initially observed, the graph G is fully observed using the propagation rule* **R2**.

The zero forcing set problem corresponds to a variant of the PDS problem where the problem consists of installing a minimum number of PMUs on the nodes such that G is fully observed. We call this problem the *Power Vertex Set* (PVS) problem. In [2], the authors present a relation linking the size of a zero forcing set and the minimum rank of the graph.

Proposition 2.8 ([2]) *Let Z be a zero forcing set of G. We have $|V| - \mathsf{mr}(G) \leq |Z|$.*

Lemma 2.9 *Let Z be a zero forcing set of G and let Π^* be an optimal solution of the PES problem. Then $|Z| \leq 2|\Pi^*|$.*

Proof. Let $Z' = \{i \in V \mid \{i,j\} \in \Pi^*\}$. Assume that all the nodes in Z' are initially observed. By applying the rule **R2**, G can be fully observed. Hence, Z' is a feasible solution of the zero forcing set. Therefore $|Z| \leq |Z'| \leq 2|\Pi^*|$. □

By Lemma 2.9 and Proposition 2.8, for any symmetric matrix $S \in \mathbb{R}^{|V| \times |V|}$ such that $S_{ij} \neq 0$ if and only if $\{i,j\} \in E$, we have that $\frac{|V| - \mathsf{rank}(S)}{2}$ is a lower bound for the PES problem. In particular, for $S = A$, we have $\frac{|V| - \mathsf{rank}(A)}{2}$. This proves the following:

Theorem 2.10 *Let $G = (V,E)$ be a graph and let A be its adjacency matrix. Then $\frac{|V| - \mathsf{rank}(A)}{2}$ is a lower bound for the PES problem.*

2.7 Modelling conventional measures

Additionally to PMUs, some non-synchronized sensors may also be installed in the electrical network. They can then provide either power flow measurements along given lines or voltage measures at some nodes. Considering these measurements in our observability model may reduce the number of PMUs to install. We present here how to take into account these conventional measurements.

- *Power flow measurements along given lines are known.* Having the flow measurement along a line allows us to calculate one of the terminal bus voltages when the other one is known using Ohm's law. Therefore, if the flow along an edge $\{i,j\}$ is known, node i is observed if node j is observed and inversely. Let M_{PF} be the set of all edges with known power flow measurements. Therefore, the following constraints are added to the lower level of (P_{PES})

$$\forall \{i,j\} \in M_{\mathsf{PF}} \quad \omega_i = \omega_j.$$

- *Bus voltages at some nodes are known.* If the bus voltage is known at a given node, then this node is observed. Let M_{BV} be the set of all the nodes with known bus voltages. Therefore, the following constraints are added to the lower level of (P_{PES})

$$\forall i \in M_{\mathsf{BV}} \quad \omega_i = 1.$$

2.8 Single line outage contingency

Some troubles may occur in the electrical network leading to some contingencies. Incorporating contingencies in the models for the PES problem would result in more reliable results. We consider in this section the single line outage contingency. Hence, the objective is to find the minimum number of PMUs to install on edges under the assumption that *any* single edge may fail and the graph should still be fully observed. We present the bilevel model corresponding to a single line outage contingency for the PES problem.

For any edge in E, let $G \setminus \{e\} = (V, E \setminus \{e\})$, and for all $i \in V$ let $\Gamma^e(i)$ be the neighbourhood of node i in $G \setminus \{e\}$. The binary bilevel linear program for the PES problem with a single line outage is given by

$$
(P_{\mathsf{LineOut}}) \begin{cases} \min\limits_{s\in\{0,1\}^m} & \sum\limits_{i\in V} \sum\limits_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} \\ \forall i\in V, j\in\Gamma(i) & s_{ij}=s_{ji} \\ \forall e\in E & n\leq \begin{cases} \min\limits_{\omega\in\{0,1\}^n} & \sum\limits_{i\in V}\omega_i \\ \forall i\in V, j\in\Gamma^e(i) & \omega_i\geq s_{ij} \\ \forall i\in V, j\in\Gamma^e(i) & \omega_i-\omega_j\geq \sum\limits_{\substack{k\in\Gamma^e(j)\\ k\neq i}}\omega_k - |\Gamma^e(j)|+1 \end{cases} \end{cases}
$$

Note that the optimal objective function value of (P_{LineOut}) is larger than that of (P_{PES}).

Similar to the bilevel model proposed for the PES problem and described in Section 2.3.2, (P_{LineOut}) can be reformulated as an MILP by applying the same process to each program of the lower level. The resulting MILP is as follows.

$$
\begin{cases} \min\limits_{\substack{s\in\{0,1\}^m\\ p,\lambda,\mu\geq 0}} & \sum\limits_{i\in V} \sum\limits_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} & & \\ \forall i\in V, j\in\Gamma(i) & s_{ij} & = & s_{ji} \\ \forall e\in E & \sum\limits_{i\in V}\sum\limits_{j\in\Gamma^e(i)} p^e_{ij} + (1-|\Gamma^e(j)|)\lambda^e_{ij} & \geq & n \\ \forall i\in V, e\in E & \sum\limits_{j\in\Gamma^e(i)} (\mu^e_{ij}+\lambda^e_{ij}-\lambda^e_{ji}-\sum\limits_{\substack{k\in\Gamma^e(j)\\ k\neq i}}\lambda^e_{kj}) & \leq & 1 \\ \forall i\in V, j\in\Gamma^e(i), e\in E & p^e_{ij} & \leq & M\, s_{ij} \\ \forall i\in V, j\in\Gamma^e(i), e\in E & p^e_{ij} & \leq & \mu^e_{ij} \\ \forall i\in V, j\in\Gamma^e(i), e\in E & p^e_{ij} & \geq & \mu^e_{ij} - M(1-s_{ij}), \end{cases}
$$

where the constant M can be taken to be the same as for (P_{MILP}), as the proof of Prop. 2.3 goes through essentially unchanged. Moreover, the cutting-plane algorithm proposed in Section 2.5 to solve (P_{PES}) can also be used to solve the program (P_{LineOut}). The fact that this is a bilevel program with $m = |E|$ lower-level subproblems simply has the effect of being able to generate multiple cuts. More precisely, cuts are generated for those slave programs for which $\Omega(s) < n$ in $G \setminus \{e\}$.

2.9 Measurement losses

Another type of contingency is the loss of measurements due to PMU failures. We consider in this section a single PMU contingency. Hence, the objective is to find the minimum number of PMUs to install on edges under the assumption that any single PMU may be lost and the graph should still be fully observed. This case is similar to the one in Section 2.8, but robustness with respect to edge removal only applies to rule **R1**; consequently, only the first constraint in the lower level is quantified over $j \in \Gamma^e(i)$, whereas the second is quantified in the standard way, i.e., over $j \in \Gamma(i)$. We refer to the resulting formulation as (P_{PMUOut}).

As in previous cases, (P_{PMUOut}) can also be reformulated as an MILP; moreover, the cutting-plane algorithm proposed in Section 2.5 to solve the model (P_{PES}) can also be used to solve the (P_{PMUOut}). The same provision holds as in (P_{LineOut}) about the m lower-level subproblems: cuts are generated for those slave programs for which $\Omega(s^e) < n$. Also, the value of an optimal solution of (P_{PMUOut}) is larger than one of an optimal solution of (P_{PES}).

2.10 Computational results

All the experiments presented here were performed on a 2.70GHz Intel i7 dual-core CPU with 16.0 GB RAM. All formulations and algorithms were implemented in Julia using JuMP [14] and solved using IBM ILOG CPLEX 12.6.

Our benchmark contains two types of graphs.

- Networks with topologies of standard IEEE n-bus systems, with $n \in \{5, 7, 14, 30, 57, 118\}$, see [24] — we call these instances Gr_n.
- Random graphs with n nodes and $m = 1.4 \times n$ for $n \in \{5i \mid 1 \leq i \leq 10\}$ — we call these instances Rnd. The constant 1.4 is the average ratio of edges over nodes in standard IEEE bus systems. These instances are allowed to be forests (hence, disconnected), but no node is isolated. For each value of n, 10 different instances were generated and tested.

The project that motivated this research provided us with instances of similar size to those reported here. On the other hand, the project itself was devoted to the study of a prototype rather than of a full deployment. Although it is unlikely that any exponential algorithm will scale to truly large scale instances, the type of networks used in the French electricity distribution system are "almost trees", since they have relatively few cycles. In other words, they are on the easier side of the spectrum.

2.10.1 Methodological comparison

Our first test compares three different methodological approaches: solving (P_{IT}) and (P_{MILP}) with CPLEX, and using the BilevelSolve algorithm. The results are obtained for the case without conventional measures, and are reported in Table 1. Each given result for the randomly generated graphs is the average over the 10 generated instances. We limited the running time to 2 hours. For any instance not solved optimally within the time limit, the running time is set to this limit.

For each purely formulation-based method, we reported the following statistics.

(i) The average CPU time expressed in seconds.

(ii) The average optimality gap, expressed as a percentage. Gap averages are taken over ratios $\frac{LB - UB}{LB}$ computed on all instances returning at least one feasible solution, where LB is the final best lower bound and UB is the best solution value found.

(iii) The number of instances *#opt* solved optimally, and the number of instances that run out of memory (marked "mof" for "memory overflow").

For BilevelSolve, we reported: (i) average CPU time in seconds, (ii) number *#itn* of iterations taken before termination (for the Rnd set, this is an average over the 10 runs), and (iii) number of optimally solved instances.

We note that the iteration-indexed formulation cannot be used to solve medium and larger size instances. The fixed-point reformulation can solve larger instances than the iteration-indexed one but cannot solve large size instances. BilevelSolve can solve almost all the considered instances in few seconds. Only one instance of the random generated graphs could not be solved within the time limit. For small instances, formulation P_{MILP} performs somewhat better than BilevelSolve. This is due to the nature of our programming language system: Julia is a Just-in-Time (JiT) environment, where each function needs to be pre-compiled every time Julia is started: this overhead is $O(1)$ seconds of user time, and so it dominates the actual algorithmic execution time. Since the amount of code to be compiled is larger for the cutting plane algorithm BilevelSolve than it is to deploy CPLEX on formulations P_{IT} and P_{MILP}, the overhead of the cutting plane algorithm is larger. For larger instances, where the overhead is a minor fraction of the total time, it is clear that BilevelSolve is on average better in terms of running time and size of instances that can be solved.

	n	m	P_{IT}			P_{MILP}			BilevelSolve		
			Time (s)	Gap (%)	*#opt* (mof)	Time (s)	Gap (%)	*#opt* (mof)	Time (s)	*#itn*	*#opt* (mof)
Gr_n	5	6	1.30	0	1	**1.29**	0	1	2.01	1	1
	7	8	6.22	0	1	**1.33**	0	1	2.14	1	1
	14	20	19.30	0	1	**1.38**	0	1	2.17	5	1
	30	41	7200.00	100	0	40.93	0	1	**2.37**	18	1
	57	80	7200.00	100	0	7200.00	63.51	0	**4.26**	38	1
	118	176	7200.00	100	0	7200.00	100.00	0	**247.41**	367	1
Rnd	5	7	**1.30**	0	10	6.42	0	10	2.09	2.6	10
	10	14	2.20	0	10	**1.37**	0	10	2.16	5.7	10
	15	21	297.60	0	10	**1.48**	0	10	2.18	9.8	10
	20	28	7200.00	69.89	0	2.34	0	10	**2.26**	17.1	10
	25	35	7200.00	96.33	0	29.16	0	10	**2.64**	29.9	10
	30	42	7200.00	95.71	0	1820.58	5.31	8	**4.86**	57.0	10
	35	49	*7200.00*	*98.14*	0(1)	3789.77	16.81	6	**15.26**	90.9	10
	40	56	7200.00	93.11	0	*6316.36*	*33.56*	1(5)	**24.34**	121.4	10
	45	62	7200.00	98.57	0	7200.00	47.14	0	148.58	211.8	10
	50	70	7200.00	93.16	0	*7200.00*	*50.36*	0(4)	414.24	446.0	9

In *italics*: average over instances that did not run out of memory

Table 1: Computational results for the PES problem.

2.10.2 Variants: conventional measures and robustness

We compare now the results obtained considering conventional measures, single edge deletion and single measurement loss. The tests were only performed for BilevelSolve and for the same instances described above. For conventional measures, the additional power flow measurements and bus voltage are generated randomly. Moreover, only instances with $n \geq 30$ were considered.

The results are given in Table 2 where we reported: (i) the number of power flow measurement *#addPF*, resp. bus voltage *#addV*, chosen so as to make the variant actually differ from (P_{PES}); (ii) the optimal number *Opt. val.* of PMUs to install. The other values are as explained before. For conventional measures, note that, since the generation of additional measures is random, *#addPF* and *#addV* are not necessarily optimal.

For conventional measures, the running time is generally less than when BilevelSolve is applied to the standard PES (P_{PES}): the additional constraints allow BilevelSolve to converge more quickly to the optimal solution. The number of conventional measures needed to be known is in general between 1 to 5.

For single edge deletion, even if the number of slave problems to solve is m (while (P_{PES}) has just one), the running time has the same order of magnitude as BilevelSolve for (P_{PES}), and decreases for random graph instances with $n \geq 35$. This can be explained by the fact that at each iteration we add several valid inequalities (one for each edge/PMU) that help convergence speed. Single measurement loss results are similar.

2.11 Other extensions and variants

We assumed here that the installation cost is the same for every PMU location at a node along an edge. If not, the problem consists then in finding the placement of PMUs that ensures a full observability of the graph and minimize the total installation cost. Let, $\forall i \in V, j \in \Gamma(i)$, c_{ij} be the cost of installing a

	n	m	(P_{PES})		Conventional measurements				Single edge del.		Single PMU loss	
					Power flow		Voltage					
			Time (s)	*Opt. val.*	Time (s)	*#addPF*	Time (s)	*#addV*	Time (s)	*Opt. val.*	Time (s)	*Opt. val.*
Gr_n	5	6	1.90	1	-	-	0	2.1	2.10	2	2.00	2
	7	8	1.90	2	-	-	-	-	2.10	4	2.00	3
	14	20	1.95	2	-	-	-	-	2.20	4	2.10	4
	30	41	2.00	5	2.00	2	1.90	1	3.13	9	3.66	8
	57	80	6.90	5	2.10	5	2.50	4	46.00	9	15.00	10
	118	176	283.00	18	24.00	3	240.00	4	390.00	29	310.00	30
Rnd	5	7	1.90	1.30	-	-	-	-	2.80	2.30	2.60	2.20
	10	14	1.80	2.20	-	-	-	-	2.60	3.90	2.20	3.50
	15	21	1.80	2.70	2.45	1.50	2.55	1.10	2.70	5.50	2.60	4.50
	20	28	2.00	3.30	2.63	1.77	2.59	2.21	3.00	6.50	3.40	5.60
	25	35	2.10	4.40	2.55	1.64	3.06	2.90	3.30	8.00	4.00	6.50
	30	42	3.10	5.80	2.39	1.70	2.59	1.40	4.30	9.40	6.80	7.80
	35	49	7.30	5.80	5.78	2.70	2.69	2.80	5.70	10.00	22.00	8.40
	40	56	10.80	7.00	6.90	2.80	6.45	2.60	8.70	12.10	28.00	9.20
	45	62	60.30	7.20	37.03	2.85	17.80	2.60	7.50	12.50	66.00	10.00
	50	70	244.00	8.70	86.65	2.93	62.35	2.63	18.00	14.10	459.40	11.00

Table 2: Computational results for conventional measurements and robust variants of the PES problem.

PMU on $\{i, j\}$. The new objective function is then given by:

$$\min \sum_{i \in V} \sum_{\substack{j \in \Gamma(i) \\ j > i}} c_{ij}\, s_{ij}.$$

Our proposed models can easily be adapted to solve the PDS problem. It suffices to define the placement variables by $s_i = 1$ if a PMU is installed on node i and 0 otherwise for each $i \in V$. The binary bilevel linear program for solving the PDS problem is the following:

$$(P_{\mathsf{Nodes}}) \left\{ \begin{array}{ll} \min\limits_{s \in \{0,1\}^m} & \sum\limits_{i \in V} s_i \\ & n \leq \left\{ \begin{array}{rl} \min\limits_{\omega \in \{0,1\}^n} & \sum\limits_{i \in V} \omega_i \\ \forall i \in V, j \in \Gamma(i) \cup \{i\} & \omega_j \geq s_i \\ \forall i \in V, j \in \Gamma(i) & \omega_i - \omega_j \geq \sum\limits_{\substack{k \in \Gamma(j) \\ k \neq i}} \omega_k - |\Gamma(j)| + 1 \end{array} \right. \end{array} \right.$$

Similarly, all the above models and approaches described for the PES problem can be adapted to solve the PDS problem.

3 The Generalized Power Edge Set (GPES) problem

We present a generalization of the PES problem called the Generalized PES (GPES) problem. The GPES problem is based on a generalization of the observability rule presented in Section 2, which might arise in electrical networks. We first state the new rule and infer it from simple electrical considerations. Secondly, we derive a bilevel formulation of the GPES problem. Different from (P_{PES}), the lower-level subproblem of the GPES formulation has an exponential number of constraints; it is therefore not tractable in its original form. We therefore propose a (polynomial-time) combinatorial algorithm to solve the observability problem.

3.1 Observability generalization rules

Let $S \subseteq V$ and let $\bar{S} = V \setminus S$ be its complementary set. For any $i \in V$, we denote by $\Gamma_S(i)$ the neighbourhood of i belonging to S, i.e., $\Gamma_S(i) = \Gamma(i) \cap S$. The neighbourhood of S, denoted by $\Gamma(S)$, is defined by $\Gamma(S) = \left(\bigcup_{i \in S} \Gamma_{\bar{S}}(i)\right)$. For $S \subseteq V$, the *border* of S, denoted δS, is defined as $\delta S = \{i \in S \mid \Gamma_{\bar{S}}(i) \neq \varnothing\}$. The *interior* of S is the set $S^\circ = S \setminus \delta S$. Note that border and interior are only meaningful for nontrivial subsets of V — in a way, they are a relative notion (to V).

Let $\Pi \subseteq E$ be a placement of PMUs in G and let $\Omega \subseteq V$ be the set of observed nodes. We consider the two following observation rules.

R1: If a PMU is placed on an edge $\{i, j\}$, then nodes i and j are observed:

$$\{i, j\} \in \Pi \Rightarrow i, j \in \Omega.$$

(Note that **R1** is defined exactly as in Section 2.1.)

R2′: If a subset S has all its border set observed as well as all its neighbour nodes observed, except one, then this neighbour node is observed:

$$\delta S \subseteq \Omega \ \wedge \ |\Gamma(S) \smallsetminus \Omega| \leq 1 \Rightarrow \Gamma(S) \subseteq \Omega.$$

For subsets S having one element, the rule **R2′** is exactly the same as the rule **R2**. For a subset S with $|S| \geq 2$, an example of the applicability of rule **R2′** is given in Figure 2 where $\delta S = \{j, j_1, j_2, j_3, j_4\}$ and $\Gamma(S) = \{i, j'_1, j'_2, j'_3\}$.

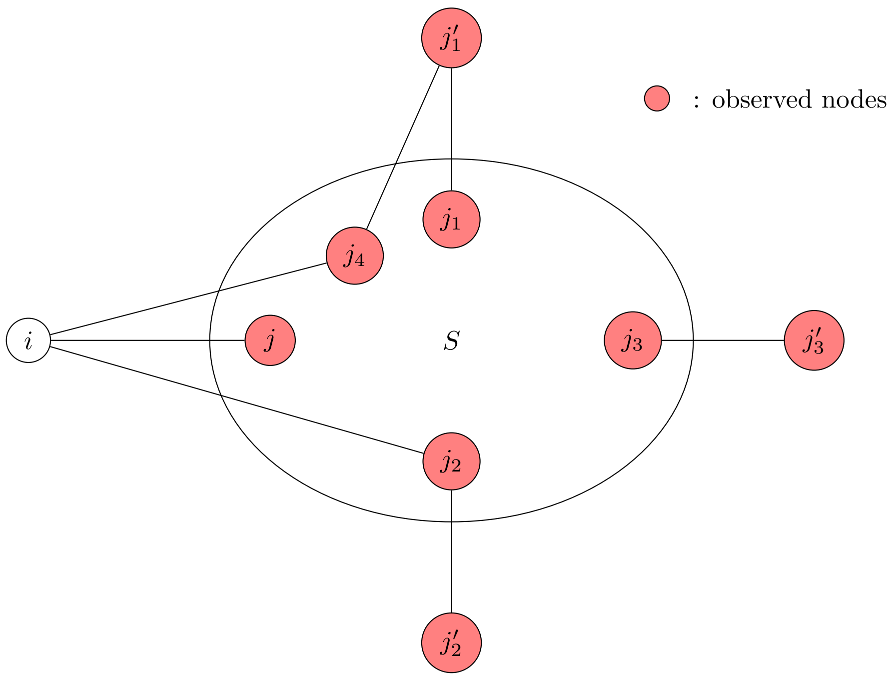

Figure 2: The node i can be observed through the subset S.

Formally speaking, then, the GPES problem is: given $G = (V, E)$, find $\Pi \subseteq E$ of minimum cardinality such that $\Omega = V$, where the relationship between Π and Ω is given by **R1** and **R2′**.

Corollary 3.1 *For $S \subseteq V$,*

$$\sum_{i \in \delta S} \sum_{j \in \Gamma_{\bar{S}}(v)} \mathbf{I}(i,j) = 0.$$

Proof. By Kirchhoff's law we have $\sum_{i \in S} \sum_{j \in \Gamma(i)} I(i,j) = 0$. Notice that

$$\begin{aligned}
\sum_{i \in S} \sum_{j \in \Gamma(i)} \mathbf{I}(i,j) &= \sum_{i \in \delta S} \sum_{j \in \Gamma(i)} \mathbf{I}(i,j) + \sum_{i \in S^\circ} \sum_{j \in \Gamma(i)} \mathbf{I}(i,j) \\
&= \sum_{i \in \delta S} \sum_{j \in \Gamma_{\bar{S}}(i)} \mathbf{I}(i,j) + \sum_{i \in \delta S} \sum_{j \in \Gamma_S(v)} \mathbf{I}(i,j) + \sum_{i \in S^\circ} \sum_{j \in \Gamma(i)} \mathbf{I}(i,j) \\
&= \sum_{i \in \delta S} \sum_{j \in \Gamma_{\bar{S}}(i)} \mathbf{I}(i,j) + \sum_{i \in S} \sum_{j \in \Gamma_S(i)} \mathbf{I}(i,j)
\end{aligned}$$

For all $i, j \in V$ we have $\mathbf{I}(i,j) = -\mathbf{I}(j,i)$, so we conclude that $\sum\limits_{i \in S} \sum\limits_{j \in \Gamma_S(i)} \mathbf{I}(i,j) = 0$. Therefore, we have $\sum\limits_{i \in \delta S} \sum\limits_{j \in \Gamma_{\bar{S}}(i)} \mathbf{I}(i,j) = 0$. □

Proposition 3.2 *The rules* **R1** *and* **R2′** *are consistent with Ohm's and Kirchhoff's laws.*

Proof. As in Section 2.1, if $\{i,j\} \in \Pi$, then i and $j \in \Omega$ using Ohm's law.

Now let us consider $S \subset V$ and $i \in \Gamma(S)$ such that $\delta S \subseteq \Omega$ and $\Gamma(S) \setminus \{i\} \subseteq \Omega$. By Corollary 3.1 we have

$$\sum_{\substack{j \in \Gamma(S) \\ j \neq i}} \sum_{k \in \Gamma_S(j)} \mathbf{I}(k,j) + \sum_{h \in \Gamma_S(i)} \mathbf{I}(h,i) = 0.$$

By Ohm's law, we deduce that

$$\sum_{h \in \Gamma_S(i)} \frac{\mathbf{V}(h) - \mathbf{V}(i)}{R(h,i)} = \sum_{\substack{j \in \Gamma(S) \\ j \neq i}} \sum_{k \in \Gamma_S(j)} \frac{\mathbf{V}(k) - \mathbf{V}(j)}{R(k,j)}$$

After rearranging we obtain

$$\mathbf{V}(i) = \frac{\sum\limits_{\substack{j \in \Gamma(S) \\ j \neq i}} \sum\limits_{k \in \Gamma_S(j)} \frac{\mathbf{V}(k) - \mathbf{V}(j)}{R(k,j)} + \sum\limits_{h \in \Gamma_S(i)} \frac{\mathbf{V}(h)}{R(h,i)}}{\sum\limits_{h \in \Gamma_S(i)} \frac{1}{R(h,i)}}.$$

By hypothesis, all the quantities in the right-hand side of the equation above are known. Thus $\mathbf{V}(i)$ can be computed. Therefore $i \in \Omega$. □

3.2 Bilevel model

Similar to Section 2.3.2, the GPES problem can be formulated by a binary bilevel model as follows:

$$(P_{\mathsf{GPES}})\left\{\begin{array}{ll} \min\limits_{s\in\{0,1\}^m} & \sum\limits_{i\in V}\sum\limits_{\substack{j\in\Gamma(i)\\ j>i}} s_{ij} \\ \forall i\in V, j\in\Gamma(i) & s_{ij}=s_{ji} \\ & n\leq \left\{\begin{array}{ll} \min\limits_{\omega\in\{0,1\}^n} & \sum\limits_{j\in V}\omega_j \\ \forall i\in V, j\in\Gamma(i) & \omega_j\geq s_{ij} \\ \forall S\subset V, i\in\Gamma(S) & \omega_i\geq \sum\limits_{\substack{j\in\Gamma(S)\cup\delta S\\ j\neq i}}\omega_j-|\Gamma(S)\cup\delta S|+2. \end{array}\right. \end{array}\right.$$

Note that, if the whole graph is observed, then for all $i \in V$ either there exists a PMU installed on $\{i,j\} \in E$ or, using $\mathbf{R2}'$, we can find a subset S such that $i \in \Gamma(S)$ and $\sum\limits_{\substack{j\in\Gamma(S)\cup\delta S\\ j\neq i}} \omega_j = |\Gamma(S)\cup\delta S|-1$. In the latter case, we have $\omega_i \geq 1$ and therefore $\omega_i = 1$. Conversely, if there exists $i \in V$ that cannot be observed from any subset S where $i \in \Gamma(S)$, then none of the constraints $\omega_i - \sum\limits_{\substack{j\in\Gamma(S)\cup\delta S\\ j\neq i}} \omega_j \geq 2-|\Gamma(S)\cup\delta S|$ will be active for any set S, and therefore $\omega_i = 0$.

Unfortunately (P_{GPES}) is not tractable in this form, since the lower-level subproblem has exponentially many constraints. We show in the following that the lower-level subproblem can nonetheless be solved in polynomial time.

3.3 Determining the observability in polynomial time

In this section we present a polynomial-time algorithm to detect if, given a placement $\Pi \subseteq E$, the graph is fully observed or not. Similar to the PES problem, we define an observability propagation dynamics. For each $d \in I = \{0,\ldots,\iota\}$ with $\iota = n-1$, $\Omega^d \subseteq \Omega$ is defined recursively by $\Omega^0 = \{i \in V \mid \{i,j\} \in \Pi\}$ and

$$\Omega^{d+1} = \{i \in V \mid i \text{ can be observed from nodes in } \Omega^d \text{ using } R2' \text{ at most one time}\}.$$

The algorithm framework is the following: assume that for some $d \in I' = I \smallsetminus \{\iota\}$, the set Ω^d is known. For each $i \notin \Omega^d$, we determine if there exists a subset $S \subseteq V$ such that $i \in \Gamma(S)$ and $\delta S \cup \Gamma(S) \setminus \{i\} \in \Omega^d$. In this case we say that i can be observed through the set S at iteration $d+1$.

Lemma 3.3 *Let $i \in V$ and let $S \subset V$ be such that i can be observed through S at iteration $d+1 \in I'$. Then:*

- *for all $j \in S$, $j \notin \Omega^d$, $\Gamma(j) \subset S$*
- *for all $j \in S \cap \Omega^d$, if there exists $k \neq i \in \Gamma(j)$ such that $k \notin \Omega^d$, then $k \in S$.*

Proof. Let $j \in S$ and $j \notin \Omega^d$. If there exists $k \in \Gamma(j)$ such that $k \notin S$ then $j \in \delta S$. However $j \notin \Omega^d$, a contradiction. Similarly if there exists $j \in S \cap \Omega^d$, and $k \neq i \in \Gamma(i)$ such that $k \notin \Omega^d$, then k must belong to S otherwise at least two neighbour nodes of S, i and k, are not observed and then i cannot be observed through S at iteration $d+1$. □

At iteration $d+1 \in I'$, for all $u \in \Gamma(\Omega^d)$ we test, by examining all the neighbour nodes of i in Ω^d, whether there exists a subset S such that i can be observed through S. If such a subset S does not exist then i cannot be observed using the placement Π. Algorithm 3 describes the polynomial-time algorithm for determining the observability of a graph given a placement Π of PMUs. For each $i \notin \Omega^d$, there are at most n^2 iterations to check if $i \in \Omega^{d+1}$. Therefore Algorithm 3 runs in time $O(n^3)$.

Algorithm 3: Observability Algorithm

Let Π be a PMU placement
$d \leftarrow 0$
$\Omega^d \leftarrow \{i \in V \mid \{i,j\} \text{ or } \{j,i\} \in \Pi\}$
repeat
  $d \leftarrow d + 1$
  $\Omega^d \leftarrow \Omega^{d-1}$
  for each $i \in \Gamma(\Omega^{d-1})$ **do**
    $J \leftarrow \Gamma(i) \cap \Omega^{d-1}$
    for $j \in J$ **do**
      Determine if i can observed through a subset S containing j
      if such an S exists **then**
        $\Omega^d \leftarrow \Omega^d \cup \{i\}$
      end if
    end for
  end for
until $\Omega^d = \Omega^{d-1}$ or $\Omega^d = V$
return Ω^d

3.4 Computational results

The experiments were performed on the same computer and the same instances as in Section 2.10. As above, the algorithm was implemented in Julia and solved using IBM ILOG CPLEX 12.6. We implemented BilevelSolve, proposed in Section 2.5, to solve (P_{GPES}) by using Algorithm 3 to solve the slave problem. The tests were performed on the same previous instances considered in Section 2.10. The results obtained are reported in Table 3. Each given value for the randomly generated graphs is the average over the 10 generated instances. We limited the running time to 2 hours. For any instance which is not solved optimally within the time limit, the running time is set to this limit. We reported: (i) the average CPU time expressed in seconds; (ii) the value of an optimal solution; and (iii) the number of instances *#opt* solved optimally.

We note that for graph instances $\mathsf{Gr\_}n$, for $n \in \{5, 14, 30, 57\}$, the optimal value of the GPES problem is the same as the PES one but the running time is much longer due to the running time of Algorithm 3, which needs to consider many subset combinations in order to apply $\mathbf{R2'}$. Furthermore, the GPES problem cannot be solved for $n = 118$, as shown by the absence of statistics in the table. However, the bilevel algorithm for the GPES problem returned better optimal values for $\mathsf{Gr\_7}$ and all the random generated graphs with, in general, a better running time. This can be explained by the fact that although the algorithm to solve the observability problem is slower in the case of the GPES problem, we notice that the value of the optimal solution is smaller than for the PES problem. Therefore the algorithm will converge "faster" to an optimal solution.

As a result, the model for the GPES problem can be used for small and medium size instances.

4 Conclusion

In this paper, we study a variant of the Phasor Measurement Unit placement problem on an undirected graph representing an electrical network. The PMU placement problem, also known as the Power Dominating Set problem, consists of finding the minimum number of PMUs to place on the nodes such that the graph is fully observed, i.e., the network state is known. In the variant we consider, called the Power Edge Set problem, the PMUs are placed on edges instead of nodes. We propose extended propagation rules based on Ohm's and Kirchoff's laws and two alternative mathematical programming

	n	m	GPES			PES Bilevel		
			Time (s)	*#Opt. val.*	*#Opt.* (mof)	Time (s)	*#Opt. val.*	*#Opt.* (mof)
Gr_n	5	6	1.90	1	1	1.87	1	1
	7	8	1.74	1	1	1.90	2	1
	14	20	1.70	2	1	1.95	2	1
	30	41	2.35	5	1	2.00	5	1
	57	80	93.38	5	1	6.90	5	1
	118	176	7200.00	-	-	283.00	18	1
Rnd	5	7	1.61	1.02	10	1.90	1.30	10
	10	14	1.63	2.00	10	1.80	2.20	10
	15	21	1.74	2.19	10	1.80	2.70	10
	20	28	1.88	2.32	10	2.00	3.30	10
	25	35	2.10	2.65	10	2.10	4.40	10
	30	42	2.92	3.21	10	3.10	5.80	10
	35	49	7.47	3.50	10	7.30	5.80	10
	40	56	11.90	4.10	10	10.80	7.00	10
	45	62	18.56	3.42	10	60.30	7.20	10
	50	70	26.94	4.13	10	244.00	8.70	9

Table 3: Computational results for the GPES problem.

formulations based on these rules, namely the iteration-indexed formulation and the bilevel formulation.

The former is a binary linear programming problem which can be solved through any standard (Mixed) Integer Linear Programming (MILP) solver. The latter can be derived from the former by means of a fixed-point argument. In order to solve it, we propose two approaches. First, we show that it can be reformulated as an MILP problem that can also be solved via a standard solver. The second approach is a cutting plane algorithm which natively solves the bilevel formulation. These two formulations and the solution algorithm were implemented and tested on six IEEE bus systems and 100 randomly generated instances. The cutting plane algorithm turns out to be the only method capable of solving large instances.

We then discuss how to adapt our models to tackle robust variants of the PES problem to protect against contingencies. In the second part of the paper, we presented the Generalized Power Edge Set problem, i.e., a generalization of the PES problem in which the propagation rules apply to a subset of nodes instead of just one node. The bilevel formulation is adapted to this case and a polynomial-time algorithm for solving the GPES problem is proposed. This bilevel algorithm is tested on the same instances as the ones for the PES problem. The results showed that this general model can only be used for small and medium size instances.

A future direction for further work is to generalize our formulations and methods to the case of PMUs with more than one limited channel. Also, due to maintenance or repairs the electrical network topology is not fixed. Hence, another interesting perspective is to study the PES problem under these conditions by proposing a robust model and a solution method to solve it. For very large scale instances, we believe we shall have to resort to a heuristic or approximate solution approach.

References

[1] A. Aazami and M. Stilp, Approximation algorithms and hardness for domination with propagation, Approximation, Randomization, Combinatorial Optim: Algorithms Techniques, Springer, LNCS, vol. 4627, New York, 2007, pp. 1–15.

[2] AIM Minimum Rank–Special Graphs Work Group, Zero forcing sets and the minimum rank of graphs, Linear Algebra its Appl 428 (2008), 1628–1648.

[3] F. Aminifar, A. Khodaei, M. Fotuhi-Firuzabad, and M. Shahidehpour, Contingency-constrained PMU placement in power networks, IEEE Trans Power Syst 25 (2010), 516–523.

[4] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge University Press, Cambridge, 2004.

[5] D.J. Brueni and L.S. Heath, The PMU placement problem, SIAM J Discr Math 19 (2005), 744–761.

[6] S. Dambhare, D. Dua, R.K. Gajbhiye, and S.A. Soman, Optimal zero injection considerations in PMU placement: An ILP approach, 16th Power Syst Computation Conference, Piscataway, IEEE, 2008, pp. 1812–1820.

[7] M. Dorfling and M.A. Henning, A note on power domination in grid graphs, Discr Appl Math 154 (2006), 1023–1027.

[8] R. Emami and A. Abur, Robust measurement design by placing synchronized phasor measurements on network branches, IEEE Trans Power Syst 25 (2010), 38–43.

[9] R. Emami, A. Abur, and F. Galvan, Optimal placement of phasor measurements for enhanced state estimation: A case study, 16th Power Syst Computation Conference, Piscataway, IEEE, 2008, pp. 923–928.

[10] J.L. Gross and J. Yellen, Handbook of graph theory, Taylor and Francis, Abingdon, 2003.

[11] T.W. Haynes, S.M. Hedetniemi, S.T. Hedetniemi, and M.A. Henning, Domination in graphs applied to electric power networks, SIAM J Discr Math 15 (2002), 519–529.

[12] M. Korkali and A. Abur, Placement of PMUs with channel limits, IEEE Power & Energy Soc General Meeting, Piscataway, IEEE, 2009, pp. 3054–3057.

[13] R. Kumar and V.S. Rao, Optimal placement of PMUs with limited number of channels, North American Power Symp (NAPS), Piscataway, IEEE, 2011, pp. 64–71.

[14] M. Lubin and I. Dunning, Computing in operations research using Julia, INFORMS J Comput 27 (2015), 238–248.

[15] N.M. Manousakis, G.N. Korre, and P.S. Georgilakis, Optimal placement of phasor measurement units: A literature review, 16th Int Conference Intelligent System Application to Power Syst (ISAP), Piscataway, IEEE, 2011, pp. 416–421.

[16] N.M. Manousakis, G.N. Korre, and P.S. Georgilakis, Taxonomy of PMU placement methodologies, IEEE Trans Power Syst 27 (2012), 1070–1077.

[17] G.P. McCormick, Computability of global solutions to factorable nonconvex programs: Part I Convex underestimating problems, Math Program 10 (1976), 146–175.

[18] R.F. Nuqui and A.G. Phadke, Phasor measurement unit placement techniques for complete and incomplete observability, IEEE Trans Power Delivery 20 (2005), 2381–2388.

[19] J. Penuel, J.C. Smith, and S. Shen, Integer programming models and algorithms for the graph decontamination problem with mobile agents, Networks 61 (2013), 1–19.

[20] S. Toubaline, C. D'Ambrosio, L. Liberti, P.L. Poirion, B. Schieber, and H. Schachnai, Complexité du Power Edge Set problem, Proc Ann Meeting, Paris, ROADEF, 2016, pp. .

[21] S. Toubaline, P.L. Poirion, C. D'Ambrosio, and L. Liberti, Observing the state of a smart grid using bilevel programming, Combinatorial Optim Appl, Springer, LNCS, vol. 9486, New York, 2015, pp. 364–376.

[22] West, D.B., Introduction to graph theory, Prentice Hall, Upper Saddle River, 2000.

[23] B. Xu and A. Abur, Observability analysis and measurement placement for systems with PMUs, Power Syst Conference Exposition, Piscataway, Vol. 2, IEEE, 2004, pp. 943–946.

[24] R. Zimmerman and C.E. Murillo-Sánchez, *Matpower 5.1-User's Manual*, Power Systems Engineering Research Center (PSERC) 2015.