Computers and Operations Research 96 (2018) 330–349

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Observability of power systems with optimal PMU placement

Margarida Carvalho[a,*], Xenia Klimentova[b], Ana Viana[b,c]

[a] IVADO Fellow, Canada Excellence Research Chair, École Polytechnique de Montréal, Montréal, Canada
[b] INESC TEC, Porto, Portugal
[c] School of Engineering, Polytechnic of Porto,Portugal

ARTICLE INFO

Article history:
Received 23 February 2017
Revised 12 September 2017
Accepted 31 October 2017
Available online 10 November 2017

Keywords:
Network observability
PMU
Combinatorial optimization
Bilevel programming
Cutting plane

ABSTRACT

Phasor Measurement Units (PMUs) are measuring devices that, when placed in electrical networks, observe their state by providing information on the currents in their branches (transmission lines) and voltages in their buses. Compared to other devices, PMUs have the capability of observing other nodes besides the ones they are placed on. Due to a set of observability rules, depending on the placement decisions, the same number of PMUs can monitor a higher or smaller percentage of a network. This leads to the optimization problem hereby addressed, the PMU Placement Problem (PPP) which aims at determining the minimum number and location of PMUs that guarantee full observability of a network at minimum cost.

In this paper we propose two general mathematical programming models for the PPP: a single-level and a bilevel integer programming model. To strengthen both formulations, we derive new valid inequalities and promote variable fixing. Furthermore, to tackle the bilevel model, we devise a cutting plane algorithm amended with particular features that improve its efficiency. The efficiency of the algorithm is validated through computational experiments. Results show that this new approach is more efficient than state-of-the-art proposals.

1. Introduction

Context. Phasor Measurement Units (PMUs) are measuring devices that, by providing time synchronized phasor measurements, allow monitoring of a electrical power network (Phadke, 1993). When installed in a bus (node) of a network, a PMU measures the voltage of that bus and the currents following across a given number of incident branches. The number of currents that can be measured depends on the number of channels of the device. Furthermore, voltages at the buses incident to those branches can be inferred from Ohm's law. For nodes with no associated load nor generation (zero-injection nodes), information about other nodes can be used by applying Kirchoff's current law. To the cascade process associated to the inference of values, we call propagation and say that a node (branch) is observed if we can infer its associated electric values.

Inference of values based on the above mentioned electrical laws allows for full monitoring (observability) of a network with a number of PMUs that is less than the number of buses. That value can be minimized if we optimize the PMU placement: depending on the placement decisions, the same number of PMUs can monitor a higher or smaller percentage of a network. The PMU Placement Problem (PPP) addresses this optimization goal: it aims at determining the minimum number of PMUs (and their location) that guarantee full observability of a network at minimum cost. The problem was proven to be NP-complete both for networks without zero-injection nodes, as it reduces to the dominating set problem (Garey and Johnson, 1979), and for the case where all nodes are zero-injection, where we have the power dominating set (Haynes et al., 2002).

Literature review. Both exact methods, based on Integer Programming (IP) and heuristics were proposed in the literature to address the PPP. In terms of IP models, Xu and Abur (2004) were the first proposing an integer nonlinear formulation to solve the problem. In their model the depth of propagation of the observability rules is limited. The model was later linearized by Dua et al. (2008) that studied the multiplicity of optimal PMU placements. Gou (2008) applied similar formulations to study the cases with redundancy and incomplete observability. Sodhi et al. (2010) also addressed the PPP. However, they only considered Ohm's law for value inference. This significantly simplifies the problem and results in solutions with a larger number of PMUs. Aminifar et al. (2010) proposed a linear IP model where

* Corresponding author.
E-mail addresses: margarida.carvalho@polymtl.ca (M. Carvalho), xenia.klimentova@inesctec.pt (X. Klimentova), aviana@inesctec.pt (A. Viana).

https://doi.org/10.1016/j.cor.2017.10.012

PMUs may have a different number of channels. Later their propagation model was incorporated by Fan and Watson (2015) into a novel integer linear formulation with an exponential number of variables. PMUs with varying costs were also considered in Fan and Watson (2015). In both papers this propagation model of the observability rules is limited as proven by their computational results that attain solutions with more PMUs than others already found in the literature. Finally, Poirion et al. (2016) studied the PPP restricted to 1-channel capacity (the PMU has capacity to observe two nodes and the respective transmission line). To the best of our knowledge, this is the first study that properly models the propagation of the observability rules. They propose an integer linear program and an equivalent bilevel program. The reader is addressed to Manousakis et al. (2011); 2012) for an updated state-of-the art on formulations and solution techniques.

Besides mathematical formulations, several heuristics with an emphasis on meta-heuristics, were presented in the literature, being thoroughly reviewed in Nazari-Heris and Mohammadi-Ivatloo (2015). Some of the most relevant work mentioned in that survey is referenced next. Genetic Algorithms were used by Mohammadi-Ivatloo (2009), Marín et al. (2003) and Aminifar et al. (2009). Nuqui and Phadke (2005) combine the so called tree search placement technique with Simulated Annealing, while in Mesgarnejad and Shahrtash (2008) and Koutsoukis et al. (2013) Tabu Search approaches are presented. Hajian et al. (2007) and Ahmadi et al. (2011) designed methods based on particle swarm optimization. Finally, a Chemical Reaction method is build by Xu et al. (2013). The drawback of heuristic methods is that there is no guarantee of optimality of the solutions computed.

The computational complexity of four variants of the PPP were investigated by Gyllstrom et al. (2012): (1) minimize the number of PMUs such that full observability is guaranteed, (2) maximize the number of observed buses for a fixed number of PMUs, (3) minimize the number of PMUs such that full observability is ensured, as well as redundancy, and (4) maximize the number of observed buses for a fixed number of PMUs and ensure redundancy. Their work generalizes the complexity results by Brueni and Heath (2005), which proved that even for planar bipartite graphs the decision version of the PPP is already NP-complete.

Paper contributions and organization. In this paper we extend the work by Poirion et al. (2016) in the following directions: we adapt their IP models for the general PPP, implementing observability rules for all types of nodes and considering unconstrained PMU capacity. To strengthen the mathematical models we derive new valid inequalities and promote variable fixing. Furthermore, another type of valid inequalities fully characterizing the PPP's feasible set is derived together with a polynomial time algorithm that improves those inequalities. This leads to dominant inequalities and, as a byproduct, to an upper bound on the solution. Finally, we incorporate these ingredients in a cutting plane fashion algorithm to solve the problem at hand. We also extend those results to two problem variants: the L-capacity PMU, with $L \geq 1$, and the variable cost PMU. In this last problem we consider the case where PMUs may have different capacities and associated costs.

The paper is organized as follows. Section 2 states the PMU Placement problem, establishes general notation and presents two mathematical formulations: the observability propagation model (OPM) and a bilevel programming model. Section 3 discusses valid inequalities for the set of feasible PMU placements (*i.e.*, decision plans that lead to full observability) and variable fixing for optimal solutions. To solve the bilevel mathematical programming formulation, these theoretical results plus extra crucial enhancements are gathered to build the algorithm presented in Section 4. We validate the efficiency of our approach through computational results presented in Section 5. Section 6 summarizes our contributions and provides future research lines in this context. Flowcharts of our algorithms can be found in Appendix A.

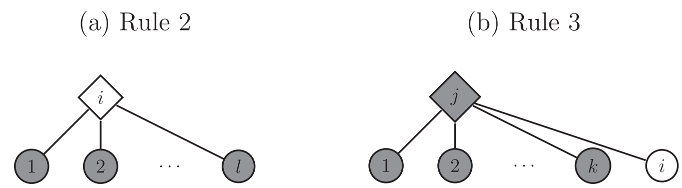

Fig. 1. Diamonds represent zero-injection nodes and circles are any node in V (zero- or non zero-injection). The gray nodes are observed; the white nodes become observed by (a) Rule 2, (b) Rule 3.

2. Problem statement and formulations

Consider a power system network (PSN) that can be represented by an undirected graph $G = (V, E)$, where V and E respectively represent sets of buses (nodes) and transmission lines (edges). We assume that if $(i, j) \in E$ then $(j, i) \in E$. The neighborhood $N(i)$ for $i \in V$ is the set of buses adjacent to i: $N(i) = \{j|(i, j) \in E\}$, and define $N[i] = N(i) \cup \{i\}$. Furthermore, the set of nodes V in G can be formally partitioned into two subsets: the set of nodes with no associated generation or load, so called zero-injection nodes, and the set of nodes with associated generation or load, non zero-injection nodes.

A phasor measurement unit (PMU) is a device that allows to monitor the state of a PSN. It can measure the voltage of the bus it is placed in and the current in all the adjacent lines. A bus is said to be observed if its voltage is know, while a transmission line is observed if its current is known. A PSN is fully observed if the voltage and currents in all its buses and transmission lines are known. Furthermore, based on two fundamental electrical circuit laws, Ohm's law and Kirchoff's current law, additional voltages and currents can be obtained.

- **Ohm's law** states that the current I_{ij} in line (i, j) is $I_{ij} = \frac{V_i - V_j}{R_{ij}}$, where R_{ij} is the resistance of the line, and V_i, V_j are the potentials (voltages) of nodes i and j, respectively.
- **Kirchhoff's current law** states that for a zero-injection bus i, $\sum_{j \in N(i)} I_{ij} = 0$.

These laws support the following set of observability rules where, for simplicity of explanation, we assume that PMUs have unlimited capacity and can observe an unlimited number of elements of the network.

Rule 1: A node i is observed if node i or one of its neighbors has a PMU;
Rule 2: A zero-injection node i is observed if all its neighbors are observed (see Fig. 1 (a));
Rule 3: If a zero-injection node j and all its neighbors, except for node i, are observed then i is observed (see Fig. 1(b)).

These rules have a cascade propagation nature. Fig. 2 illustrates observability propagation, for a given PSN with PMUs placed in nodes 1 and 9. Initially (step 0), due to rule 1 nodes 2, 3, 7 and 8 are observed. Then, in a second step, as 1,2, and 3 are observed, by rule 3 node 4 becomes observed. Further, Rule 2 is applied to node 6 as 4, 7 and 8 are observed. Finally, in the last step, again by rule 3, node 5 becomes observed.

Definition 2.1. The PMU placement problem (PPP) can be defined as follows: find a placement for the minimum number of PMUs that ensures full observability.

In the remaining of this section we propose generalizations of the two formulations presented in Poirion et al. (2016) and discuss

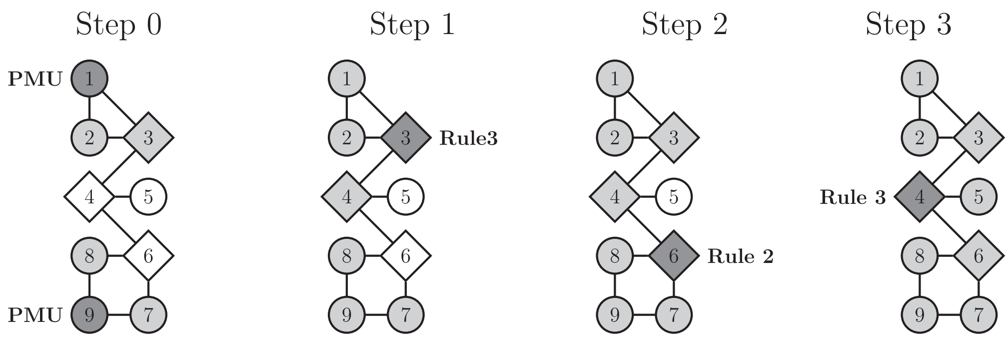

Fig. 2. Example of propagation of observability rules. Diamonds represent zero-injection nodes, circles are non zero-injection nodes. Light gray nodes are observed at the current step d; with dark gray we mark a node where one of rules 2 or 3 is applied.

their extensions to relevant problem variants. The first one, hereby called observability propagation model, is a linear integer program (IP), and the second one is a bi-level IP.

2.1. Observability propagation model (OPM)

This model is constructed based on the observability propagation derived from rules 2 and 3. Let $S \subseteq V$ be the set of zero-injection nodes. Denote by η a parameter that sets the maximum number of propagation steps of rules 2 and 3, use an index $d \in D = \{0, 1 \ldots, \eta\}$ to refer to each step of the propagation and consider $D' = D \backslash \{\eta\}$. It is enough to set $\eta = |S|$ to guarantee that rules 2 and 3 are not constrained in terms of propagation. The decision variables are the following:

- x_i is a binary variable which takes value 1 if a PMU is placed in node $i \in V$, otherwise $x_i = 0$;
- w_{id} is a binary variable which takes value 1 if node $i \in V$ is observed in step $d \in \{0, 1, \ldots, \eta\}$, otherwise $w_{id} = 0$; variables $w_{i,\eta} = 1$ represent observed nodes for a given placement of PMUs;
- s_{id} is a binary variable which takes value 1 if rule 2 is used in step d to observe node $i \in S$, otherwise $s_{id} = 0$;
- r_{ijd} is a binary variable which takes value 1 if rule 3 is used in step d to observed node $i \in N(j)$ with $j \in S$, otherwise $r_{ijd} = 0$.

In what follows we will omit indices of variables to refer to the full vector of that variable (*e.g.*, $x = (x_1, \ldots, x_{|V|})$).

The observability propagation model (PPP_{It}) can be formulated as follows:

$$\text{PPP}_{It} = \min_{\substack{x, w, s, r \\ binary}} \sum_{i \in V} x_i \tag{1}$$

$$\text{s.t.} \quad w_{i,\eta} = 1, \qquad \forall i \in V \tag{2}$$

$$x_i + \sum_{j \in N(i)} x_j \geq w_{i,0}, \qquad \forall i \in V \tag{3}$$

$$w_{i,d+1} \geq w_{i,d}, \qquad \forall i \in V, d \in D' \tag{4}$$

$$w_{i,d+1} - w_{i,d} \leq s_{i,d} + \sum_{j \in N(i) \cap S} r_{i,j,d}, \quad \forall i \in S, d \in D' \tag{5}$$

$$w_{i,d+1} - w_{i,d} \leq \sum_{j \in N(i) \cap S} r_{i,j,d}, \quad \forall i \in V \setminus S, d \in D' \tag{6}$$

$$s_{i,d} \leq w_{j,d}, \qquad \forall i \in S, \forall j \in N(i), d \in D' \tag{7}$$

$$r_{i,j,d} \leq w_{j,d}, \qquad \forall i \in V, \forall j \in N(i) \cap S, d \in D' \tag{8}$$

$$r_{i,j,d} \leq w_{k,d}, \qquad \forall i \in V, \forall j \in N(i) \cap S, \forall k \in N(j) \setminus \{i\}, d \in D' \tag{9}$$

Constraints (2) ensure that all nodes are observed and constraints (3) model rule 1. Constraints (4) imply that if node i is observed at propagation step d it remains observed for $d < \bar{d} \leq \eta$. Constraints (5) apply rules 2 and 3 to zero-injection nodes. Constraints (6) apply rule 3 to non zero-injection nodes. Constraints (7) state that rule 2 can be applied to node i in step d if all nodes in $N(i)$ are observed in step d. Constraints (8) ensure that if rule 3 is used to observe node i, then one of its adjacent zero-injection nodes, say node j, must have been observed. Constraints (9) mimic constraints (8) for the remaining neighbors of node j.

Our formulation generalizes the one in Poirion et al. (2016) to PMUs with unlimited capacity. By capacity we mean the number of adjacent transmission lines where the current can be measured (PMUs capacity will be discussed in detail in Section 2.3). Furthermore, our model incorporates rule 2.

2.2. Bilevel model

A direct adaptation of the reasoning in Poirion et al. (2016) for 1-channel PMUs, allows the binary programming problem (1)-(9) to be written as a bilevel programming problem, where first the so-called leader (upper level) selects the PMU placement, *i.e.*, fixes the variables x_i. After, the follower (lower level) observes the leader's strategy and finds the nodes observed. Notice that the lower level model mimics the propagation of the observability rules according to the PMU placement x that was selected (see Poirion et al., 2016 for more detail). Therefore, the observability variables w_{id} can let the subscript d fall in the lower level, and $w_i = 1$ if node i is observed, 0 otherwise. The bilevel model (PPP_{Bi}) is written as follows:

$$\text{PPP}_{Bi} = \min_{x \in \{0,1\}^{|V|}} \sum_{i \in V} x_i \tag{10}$$

$$\text{s.t.} \quad \sum_{i \in V} w_i \geq |V| \tag{11}$$

$$\min_{w \in \{0,1\}^{|V|}} \sum_{i \in V} w_i \tag{12}$$

$$\text{s.t.} \quad w_i \geq x_i, \quad \forall i \in V \tag{13}$$

$$w_i \geq x_j, \quad \forall i \in V, \forall j \in N(i) \tag{14}$$

$$w_i \geq 1 - |N(i)| + \sum_{j \in N(i)} w_j, \quad \forall i \in S \tag{15}$$

$$w_i \geq 1 - |N(j)| + w_j + \sum_{k \in N(j) \setminus \{i\}} w_k, \quad \forall i \in V, \forall j \in N(i) \cap S \tag{16}$$

Constraints (13) and (14) model rule 1, while constrains (15) and (16) apply rules 2 and 3, respectively. It should be noted that there is a single solution to the lower level problem: once the PMU placement is known, the nodes observed are unique. Therefore, the lower level has a single optimal solution and the distinction between pessimistic and optimistic approaches (see Colson et al., 2005; Dempe, 2002 for more details) is not necessary. We denote by $w(x)$ the solution of the lower level problem (13)–(16) for a given vector x.

2.3. Problem variants

The models introduced in Sections 2.1 and 2.2 are general enough to incorporate different variants of the PPP. In this section we address two problem variants. First, we present the case where the PMUs have limited capacity, measuring the current of at most L adjacent lines and, second, the case where each PMU can have a different capacity and different associated cost.

The following additional variables are necessary:

$$y_{ij} = \begin{cases} 1, & \text{if a PMU placed in node } i \text{ observes node } j, \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V, \forall j \in N(i)$$

L-channel PMU variant. This problem variant considers that all PMUs have equal capacity L. The following constraints need to be added to the observability propagation model and to the upper level of the bilevel model:

$$\sum_{j \in N(j)} y_{ij} \leqslant L x_i, \quad \forall i \in V \tag{17}$$

Moreover, because rule 1 changes (if node i has a L-channel PMU, only L nodes of $N(i)$ can be observed), for the observability propagation model constraint (3) must be replaced by:

$$x_i + \sum_{j \in N(i)} y_{ji} \geq w_{i,0}, \quad \forall i \in V \tag{18}$$

In the bilevel model constraints (14) are replaced by:

$$w_i \geq y_{ji}, \quad \forall i \in V, \forall j \in N(i). \tag{19}$$

We will denote the models associated to the L-channel PMU variant by PPP^L_{OPM} , for the observability propagation model, and PPP^L_{Bi} , for the bilevel model, .

PMU variable cost variant. In this problem variant we are given a set of types of PMUs, that may have different capacity and associated cost. The list of available types is presented by pairs $[(L_1, c_1), \dots, (L_n, c_n)]$, where L_k is the capacity of PMU of type k and c_k the associated cost. We also consider the variables:

$$t_{ik} = \begin{cases} 1 & \text{if a PMU with capacity } L_k \text{ and cost } c_k \text{is placed in } i, \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V, k = 1, \dots, n$$

In order to satisfy PMUs' capacity and restrict each node to at most one PMU, in this variant both the observability propagation model and the upper level of the bilevel model need the following additional constraints:

$$\sum_{j \in N(j)} y_{ij} \leqslant \sum_{k \in M} L_k t_{ik}, \quad \forall i \in V \tag{20}$$

$$\sum_{k=1}^{n} t_{ik} \leq 1, \quad \forall i \in V \tag{21}$$

The objective function for both models transforms into:

$$\min \sum_{i \in V} \sum_{k=1}^{n} c_k t_{ik} \tag{22}$$

Notice that similarly to the L-channel PMU variant, constraints (3) and (14) are modified into (18) and (19) respectively. We will denote the models associated to the PMU variable cost variant by PPP^t_{OPM} and PPP^t_{Bi} .

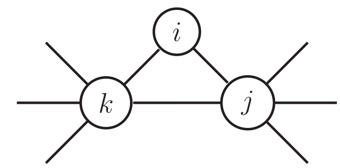

Fig. 3. Variable fixing: if a PMU is placed in node i, it can be moved to one of its adjacent nodes and the observability does not decrease.

3. Valid inequalities and variable fixing

The formulations introduced in Section 2 can be improved if additional (valid) inequalities are incorporated in the models to strengthen them. In this section we propose valid inequalities and variable fixing constraints for the PPP. These enhancements to the formulations are based on topology of the PSN.

Valid inequalities. The first improvement we propose is based on the simple observation that by applying rule 1 to a vertex i that is a non zero-injection ($i \in V \setminus S$) and that does not have zero-injection neighbors ($N(i) \cap S = \emptyset$), then for such i the following inequalities holds:

$$x_i + \sum_{j \in N(i)} x_j \geq 1. \tag{23}$$

In the remaining of this document we will refer to these inequalities as *Rule 1 inequalities*.

For the two variants of the problem modeled in Section 2.3 the Rule 1 inequalities are written as:

$$\sum_{j \in N(i)} \left(y_{ij} + y_{ji} \right) \geq 1, \quad \forall i \in V \setminus S \text{ and } N(i) \cap S = \emptyset. \tag{24}$$

Variable fixing for non-zero injection nodes. Consider the case of a non-zero injection node i with degree two that belongs to a cycle of length three, say cycle $c = (i, j, k)$ (nodes j and k can have degree greater than 2); see Fig. 3 for an illustration. If $x_i = 1$ then, by rule 1, nodes j and k become observed. If, instead, we move this PMU to node k by rule 1, the nodes in $N(k)$ become all observed, including nodes i and j. Therefore, there is no decrease in observability.

Proposition 1. *If a non-zero injection node i has degree two and belongs to a non-isolated cycle of length three, then there exists an optimal solution with $x_i = 0$.*

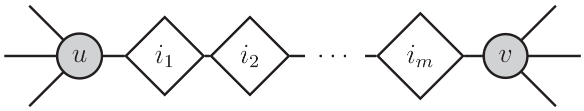

Fig. 4. A zero injection path $p = (i_1, \dots, i_m)$ with nodes u and v as supporting nodes.

Variable fixing for zero-injection nodes. We first introduce some useful concepts for a good understanding of the remaining text. For the sake of simplicity, we use vector w to refer to observed nodes. In the case of the observability propagation model w corresponds to the vector of variables $w_{i,\eta}$, $i \in V$.

Definition 3.1. A zero-injection path $p = (i_1, \dots, i_m)$ is a path in a graph G such that $\forall i_q \in p$, $q = 1, \dots, m$ we have $i_q \in S$ and $|N(i_q)| = 2$

Denote by $\mathcal{P}$ all the zero-injection paths in graph G.

Definition 3.2. The supporting nodes of a zero-injection path $p = (i_1, \dots, i_m)$ are $u \in N(i_1)\backslash\{i_2\}$ and $v \in N(i_m) \setminus \{i_{m-1}\}$. It is possible to have $u = v$.

See Fig. 4 for an illustration of the above definitions.

Lemma 3.3. *Let p be a zero-injection path having u and v as supporting nodes. If $x_q = 1$ for a node q in p, then all the nodes in the path become observed, i.e., $w_i = 1$ for all $i \in p$ and also $w_u = w_v = 1$.*

Proof. In this case, both for the observability propagation and the bilevel models, the constraints associated to rule 3 become active for the nodes of the path and also for the supporting nodes. □

If instead of placing a PMU in a node of a zero-injection path, it is placed in one of its supporting nodes, the same or a greater number of nodes is observed. This is true because rule 3 can be applied, propagating the observability.

Lemma 3.4. *If $x_u = 1 (x_v = 1)$, then $\forall i \in p$ $w_i = 1$ and $w_v = 1$ $(w_u = 1)$.*

Proposition 2. *There exists an optimal solution for the PPP such that $x_i = 0$ for all $i \in p$ of all zero-injection paths $p \in \mathcal{P}$.*

Proof. We will prove this proposition by contradiction. Suppose that there is no optimal solution that satisfies the conditions of Proposition 2. Hence, for any optimal vector x^*, there exists a path $p' \in \mathcal{P}$ with $q \in p'$ such that $x^*_q = 1$. By Lemma 3.3 $w_i = 1$ for all nodes $i \in p'$. Let us construct a vector $\bar{x}$ such that $\bar{x}_i = x^*_i$ for all $i \in V$, $i \neq q$, and $\bar{x}_q = 0$.

i) If $x^*_u = 1$ or $x^*_v = 1$ then, by Lemma 3.4, $w_i = 1$ for all $i \in p'$. Hence the vector $\bar{x}$ provides full observability of the system and $\sum_{i \in V} \bar{x}_i < \sum_{i \in V} x^*_i$, contradicting the fact that x^* is an optimum.

ii) If $x^*_u = x^*_v = 0$, we put $\bar{x}_u = 1$. Then, again by Lemma 3.4, $\bar{x}$ provides full observability and it is an optimal solution as the number of PMUs is equal to x^*. This contradicts the initial assumption that such an optimal solution does not exist. □

Proposition 2 is not valid for the variants of the PPP addressed in Section 2.3, as exemplified by Fig. 5: according to Proposition 2, it would be enough to place a PMU in the single gray node. However, that is not sufficient to attain full observability if, for example, the PMUs' capacity is $L = 1$. Nevertheless, the idea behind the proposition can be extended for cases with capacitated PMUs, as follows.

First we need to reformulate the Lemmas 3.3 and 3.4 for the L-channel and variable costs PPP. The reformulated lemmas are presented below. Text in between parenthesis refers to the variable cost variant.

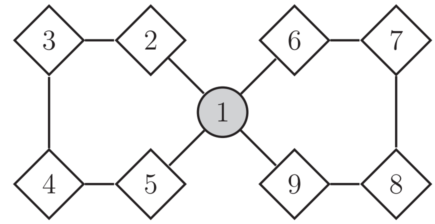

Fig. 5. Evaluation of network observability for 1-channel PMUs. Diamond nodes are zero-injection nodes and grey nodes can be any (zero- or non zero-injection). For 1-channel PMUs, one of the paths (2, 3, 4, 5) or (6, 7, 8, 9) will be not observed if Proposition 2 is used.

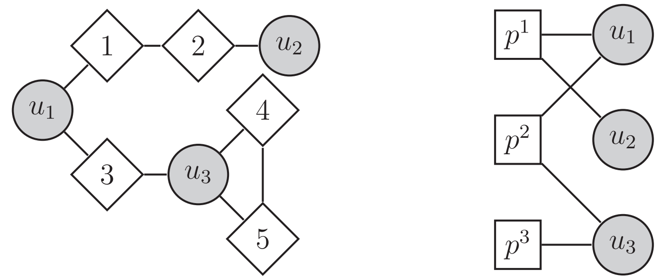

Fig. 6. Left: example of a PSN graph. Right: corresponding bipartite graph. Diamonds represent zero-injection nodes. Gray circles are supporting nodes for three zero-injection paths: $p^1 = (1, 2)$, $p^2 = (3)$ and $p^3 = (4, 5)$.

Lemma 3.5. *Let $p = (i_1, \dots, i_m)$ be a zero-injection path with supporting nodes u and v. If $x_{i_q} = 1$ ($t_{i_q,k} = 1$ for any $k = 1, \dots, n$) and $y_{i_q,i_{q+1}} = 1$, then if $L = 1$ ($L_k = 1$) all the nodes i_s in path p with $s = q+1, \dots m$ become observed as well as supporting node v. If $L > 1$ ($L_k > 1$) then all nodes in p, as well as supporting nodes u and v are observed.*

Lemma 3.6. *If $x_u = 1$ ($t_{u,k} = 1$ for any $k = 1, \dots, n$) and $y_{u,i_1} = 1$, then $\forall i \in p$ $w_i = 1$ and $w_v = 1$. Symmetrically the result is written for v and variable y_{v,i_m}.*

The contradiction reflected in Fig. 5 intuitively leads to an idea on how to deal with the case of L-channel PMUs. Let us consider the set of all zero-injection paths, $\mathcal{P}$, its supporting nodes, $U \subseteq V$, and a bipartite graph that goes from $\mathcal{P}$ to U: there is an edge $e = (p, u)$ between path $p \in \mathcal{P}$ and node $u \in U$, if u is a supporting node of path p (see Fig. 6). Denote the constructed bipartite graph by $G' = (\mathcal{P} \cup U, E')$.

We will solve a capacitated matching problem on G', taking into account the PMUs capacity L, *i.e.* the nodes in U may have at most L matched edges. For the variable cost variant the types of PMUs to be placed in nodes cannot be predefined, hence we may only assume the smallest capacity in the list of available capacities L_k for all the nodes. We set $L = \min_{k=1,\dots,n} L_k$ and solve the matching problem, presented below.

The IP model to solve considers the following variables:

$$z_e = \begin{cases} 1, & \text{if edge } e \text{ is selected,} \\ 0 & \text{otherwise} \end{cases} \quad \forall e \in E';$$

and can be written as follows:

$$\max_{z \in \{0,1\}^{|E'|}} \quad \sum_{e \in E'} z_e \tag{25}$$

$$\text{s.t.} \quad \sum_{e \in E' : u \in e} z_e \leq L, \forall u \in U \tag{26}$$

$$\sum_{e \in E' : p \in e} z_e \leq 1, \forall p \in \mathcal{P} \tag{27}$$

Constraints (26) reflect PMU's capacity. Furthermore, constraints (27) force each path to be supported only by one node (otherwise it could be counted twice in the objective). This optimization problem is totally unimodular and thus, binary requirements can be removed.

Let z^* be an optimal solution for problem (25)–(27). Consider the set of paths $\mathcal{P}' \subseteq \mathcal{P}$, supported by the optimal solution z^*, *i.e.*, such that $z_e^* = 1\ \forall p \in \mathcal{P}'$ where node p in G' is incident to edge e.

Proposition 3. *There is an optimal solution for the* L*-channel PPP where $x_i = 0$ for all $i \in p$, $p \in \mathcal{P}'$.*

Proof. By contradiction, suppose that no optimal solution satisfies the conditions of the proposition. In other words, for any optimal solution (x^*, y^*), there is a zero-injection path $\bar{p} \in \mathcal{P}'$ with extremes $\{u, v\}$, such that $\exists i_q \in \bar{p}$ with $x_{i_q}^* = 1$, $y_{i_q,i_{q+1}}^* = 1$ (also $y_{i_{q-1},i_q}^* = 1$, if $L > 1$).

Let us construct a new solution $(\bar{x}, \bar{y})$, such that $\bar{x}_i = x_i^*$ for $i \neq i_q$, $\bar{y}_{ij} = y_{ij}^*$, $(i, j) \neq (i_q, i_{q+1}), (i_{q-1}, i_q)$. Assume $\bar{x}_{i_q} = \bar{y}_{i_q,i_{q+1}} = \bar{y}_{i_{q-1},i_q} = 0$, *i.e.*, the PMU in node i_q is now excluded, and try to place it in supporting nodes u or v. There are the following possibilities for the placement:

(i) If one of the supporting nodes of $\bar{p}$, say u, do not have a PMU ($\bar{x}_u = 0$), then we place there the PMU, *i.e.*, put $\bar{x}_u = 1$, and use its capacity to observe neighborhood node i_1 of path $\bar{p}$. By Lemma 3.6 for all $i \in \bar{p}$ $w_i = 1$ and $w_u = w_v = 1$. Hence $\bar{x}$ provides full observability of the network and $\sum_{i \in V} \bar{x}_i = \sum_{i \in V} x_i^*$. Thus $\bar{x}$ is an optimal solution with no PMUs on path $\bar{p} \in \mathcal{P}'$.

(ii) If $\bar{x}_u = \bar{x}_v = 1$ and at least one of this PMUs, say u, uses its capacity to observe the path ($\bar{y}_{u,i_1} = 1$), then by Lemma 3.6 all the nodes in path $\bar{p}$ are observed as well as its supporting nodes. This guarantees the full observability of the network (by construction of $\bar{x}$). However, $\sum_{i \in V} \bar{x}_i < \sum_{i \in V} x_i^*$ which contradicts the optimality of x^*.

(iii) Finally, $\bar{x}_u = \bar{x}_v = 1$ and $\bar{y}_{u,i_1} = \bar{y}_{v,i_m} = 0$. Let u be the supporting node that was matched, *i.e.*, for edge $\bar{e} = (\bar{p}, u) \in E'$ $z_{\bar{e}}^* = 1$. Due to capacity constraints (26) there should exist an $h \in V$ with $\bar{y}_{u,h} = 1$, such that $h \notin p\ \forall p \in \mathcal{P}'$. Indeed, if this is not true, then constraints (26) are not active for u (and consequently constraints (17) are also not active). If we set $\bar{y}_{u,i_1} = 1$ the network becomes full observed by $\bar{x}$. However this contradicts the optimality of x^*. Hence $\exists h \in V$, $h \notin p$, for all $p \in \mathcal{P}'$. Let us set $\bar{y}_{u,h} = 0$, and $\bar{y}_{u,i_1} = 1$, if in this case node h is still observed then we again obtain a contradiction to the optimality of x^*. Otherwise, we may put the PMU in h: $\bar{x}_h = 1$, and use the capacity of the PMU in u to observe $\bar{p}$: $\bar{y}_{u,i_1} = 1$. This leads to an optimal solution $(\bar{x}, \bar{y})$ with no PMUs on path $\bar{p}$.

To conclude, if there is only one node i_q such that $x_{i_q}^* = 1$ and $y_{i_q,i_{q+1}}^* = 1$ (also $y_{i_{q-1},i_q}^* = 1$, if $L > 1$) in all the paths in $\mathcal{P}'$, then we constructed an optimal solution $(\bar{x}, \bar{y})$ with no PMUs on nodes of zero-injection paths, which contradicts the initial assumption. Otherwise we may repeat all the steps for the other nodes and obtain an equal contradiction. □

Remark 1. In order to maximize the number of fixed variables we may consider the weighted matching problem. Then objective function (25) is written as:

$$\max_{z \in \{0,1\}^{|E'|}} \sum_{e \in E'} z_e |w_e|$$

where $w_e = |p|$, such that p is incident to e, $p \in \mathcal{P}'$.

Notice that for $L = 1$ the result can be improved by fixing the use of the PMU's capacity from unmatched supporting nodes. Consider the set of unmatched nodes $U_0 \subseteq U$, *i.e.*, $u \in U_0$ if there is no $e \in E'$ with $z_e^* = 1$ and $u \in e$. For these nodes Proposition 4 holds.

Proposition 4. *There exists an optimal solution for the 1-channel PPP such that $y_{ui} = 0$ for all $u \in U_0$ and $i \in p$, $p \in \mathcal{P}'$.*

Proof. By contradiction, we assume that none of the optimal solutions satisfies the condition of the proposition. Hence for any optimal pair (x^*, y^*) there exists an $u \in U_0$ supporting node for path $\bar{p} = (i_1, \ldots, i_m) \in \mathcal{P}'$ such that $y_{u,i_m}^* = 1$. Notice that due to model constraints $x_u^* = 1$. To prove the proposition we will follow a strategy similar to the one used when proving Proposition 3.

We construct a pair $(\bar{x}, \bar{y})$ such that it coincides with (x^*, y^*) in all the indices except $\bar{x}_u = 0$ and $\bar{y}_{u,i_m} = 0$. Notice that, as we have $L = 1$, we are not losing any observability in nodes i that do not belong to the path. Now we place the excluded PMU in the matched supporting node of path $\bar{p}$, say v. Similarly to Proposition 3, there are several possible cases:

(i) If $\bar{x}_v = 0$, then we set it to 1, which leads to the contradiction of the initial assumption.

(ii) If $\bar{x}_v = 1$ and $y_{v,i_1} = 1$, path $\bar{p}$ is observed due to the Lemma 3.6 and we may conclude contradiction to the optimality of x^*.

(iii) If, $\bar{x}_v = 1$ and $\bar{y}_{v,h} = 1$, for some $h \in V$, $h \notin \bar{p}$, and if we transfer the capacity of the PMU in v to observe path $\bar{p}$, keeping $w_h = 1$, then we come again to contradiction with optimality of x^*. Otherwise if we put $\bar{x}_h = 1$ and $y_{v,i_1} = 1$, we guarantee the full observability and optimality of $(\bar{x}, \bar{y})$ contradicts the initial assumption. □

For the PPP with variable costs Proposition 5 holds.

Proposition 5. *There is an optimal solution for the PPP with variable costs such that $t_{ik} = 0$ for all $k = 1, \ldots, n$, $i \in p$, $p \in \mathcal{P}'$, where $\mathcal{P}'$ is determined accordingly with $L = \min_{k=1,\ldots,m} L_k$.*

The proof of the proposition mimics the proof of Proposition 3.

4. Algorithmic approach for the bilevel problem

To the extent of our knowledge, MibS (Ralphs, 2017) is the only solver available to tackle general mixed integer bilevel programming problems (MIBPs). In general, MIBPs are hard to solve being NP-complete (even if all variables are continuous (Dempe, 2002)) or even Σ_2^p-complete (Caprara et al., 2014) and general purpose MIBP's solvers are not efficient. In this paper, we propose a problem tailored cutting plane algorithm to tackle the PPP_{Bi}, that exploits its particular structure. To this end, in Section 4.1 we replace the follower's problem in PPP_{Bi} by an exponential number of constraints and in Section 4.2, we propose a cutting plane approach to solve the new formulation. Section 4.3 provides a polynomial time algorithm to compute the observed nodes given a PMU placement. Sections 4.4 and 4.5 add extra ingredients to the basic cutting plane method. Section 4.6 summarizes our final method. For a clear understanding of our approach, this section is complemented with the flowcharts in Appendix A.

4.1. Single-level formulation

Analogously to Poirion et al. (2016), we re-write the PPP_{Bi} as a single-level problem. However, contrarily to Poirion et al. (2016) that addresses the 1-channel PMU problem, we consider a general model that can handle instances with PMU's of different capacity. Furthermore, the valid inequalities presented by us, focus in the information given by the nodes observed by a given PMU placement, in opposition to Poirion et al. (2016) that focus in the localization of the PMUs.

The PPP_{Bi} is equivalent to the following single-level optimization problem:

$$\text{PPP}_{\bar{B}i} = \min_{x \in \{0,1\}^{|V|}} \sum_{i \in V} x_i \tag{28}$$

$$\text{s.t.} \sum_{i \in V: w_i(\bar{x}) + \sum_{j \in N(i)} w_j(\bar{x}) \leq |N(i)|} x_i \geq 1, \ \forall \bar{x} \in \mathcal{I}(\text{PPP}_{Bi}) \tag{29}$$

where $w_i(\bar{x})$ takes value 1 if node i is observed under the PMU placement $\bar{x}$, otherwise it is 0, and $\mathcal{I}(\text{PPP}_{Bi}) = \{\bar{x} | \sum_{i \in V} w_i(\bar{x}) < |V|\}$, *i.e.*, it is the set of infeasible PMU placements (full observability is not attained). In the following text, we show the equivalence of PPP_{Bi} and $\text{PPP}_{\bar{B}i}$.

Let us define $\mathcal{I}(\bar{x}) = \{i \in V | w_i(\bar{x}) + \sum_{j \in N(i)} w_j(\bar{x}) \leq |N(i)|\}$ as the set of nodes i for which at least a node in $N[i]$ is not observed by $\bar{x}$.

Lemma 4.1. *For all* $i \in \mathcal{I}(\bar{x})$, $\bar{x}_i = 0$.

Proof. If $i \in \mathcal{I}(\bar{x})$, then $\bar{x}_i = 0$. Otherwise, if $\bar{x}_i = 1$, then rule 1 implies that node i and its neighbors, $N(i)$, are all observed, *i.e.*, $w_i(\bar{x}) = 1$ and $w_j(\bar{x}) = 1$ for all $j \in N(i)$. □

Theorem 1. *PPP_{Bi} is equivalent to the $\text{PPP}_{\bar{B}i}$.*

Proof. We show this result by proving that the set of infeasible PMU placements to PPP_{Bi}, $\mathcal{I}(\text{PPP}_{Bi})$, is equal to the set of infeasible solutions to $\text{PPP}_{\bar{B}i}$. If that is proven, since PPP_{Bi} and $\text{PPP}_{\bar{B}i}$ have the same objective function and space of solutions, $\{0, 1\}^{|V|}$, we are able to conclude that PPP_{Bi} is equivalent to $\text{PPP}_{\bar{B}i}$.

By definition, $\mathcal{I}(\text{PPP}_{Bi})$ is the set of infeasible solutions to PPP_{Bi}. For each $\bar{x} \in \mathcal{I}(\text{PPP}_{Bi})$ and for all $i \in \mathcal{I}(\bar{x})$, Lemma 4.1 implies that $\bar{x}_i = 0$. Then, the associated constraint (29) implies that $\bar{x}$ is infeasible also to $\text{PPP}_{\bar{B}i}$. This shows that the $\mathcal{I}(\text{PPP}_{Bi})$ is contained in the set of infeasible solutions of $\text{PPP}_{\bar{B}i}$.

If x' is infeasible to $\text{PPP}_{\bar{B}i}$ then, for a given $\bar{x} \in \mathcal{I}(\text{PPP}_{Bi})$ constraints (29) are not satisfied by x'. In other words, for all nodes $i \in \mathcal{I}(\bar{x})$, we have $x'_i = 0$. Thus, x' is contained in $\bar{x}$ and, consequently, $w(x') \subseteq w(\bar{x})$. Given that $\bar{x}$ did not guarantee full observability, we conclude that x' is also infeasible to PPP_{Bi}. □

Variants. For the PPP^L_{Bi} and PPP^t_{Bi} formulations, there is a totally analogous reasoning to re-write the models as single level formulations. It is simply necessary to replace their lower level optimization problem by:

$$\sum_{i \in V} \sum_{j \in N(i): w_i(\bar{y}) + w_j(\bar{y}) \leq 1} y_{ij} \geq 1, \quad \forall \bar{y} \in \mathcal{I}(\text{PPP}_{Bi})^y, \tag{30}$$

where $w_i(\bar{y})$ takes value 1 if node i is observed under the PMU placement $\bar{y}$, and 0 otherwise, and $\mathcal{I}(\text{PPP}_{Bi})^y = \{\bar{y} | \sum_{i \in V} w_i(\bar{y}) < |V|\}$, *i.e.*, it is the set of infeasible PMU placements.

Let us define $\mathcal{I}(\bar{y}) = \{(i, j) \in E | w_i(\bar{y}) + w_j(\bar{y}) \leq 1\}$. Essentially, if for a PMU placement y, $(i, j) \notin \mathcal{I}(\bar{y})$ (*i.e.*, $w_i(y) + w_j(y) = 2$), adding a PMU in node i to observe node j, resulting in $y_{ij} = 1$, will not increase the number of nodes observed. Thus, for any feasible PMU placement y and all $\bar{y} \in \mathcal{I}(\text{PPP}_{Bi})^y$, there is a $i \in V$ and a $j \in N(i)$ such that $w_i(\bar{y}) + w_j(\bar{y}) < 1$ and $y_{ij} = 1$ (*i.e.*, constraints (30) are satisfied).

In this way we get the following single-level formulations for the L-channel PPP and for the variable cost PPP. Respectively:

$$\text{PPP}^L_{\bar{B}i} = \min_{x \in \{0,1\}^{|V|}, y \in \{0,1\}^{|E|}} \sum_{i \in V} x_i \tag{31}$$

$$\text{s.t.} \sum_{j \in N(i)} y_{ij} \leq L x_i, \qquad \forall i \in V \tag{32}$$

and constraints (31)

and

$$\text{PPP}^t_{\bar{B}i} = \min_{t \in \{0,1\}^{n|V|}, y \in \{0,1\}^{|E|}} \sum_{i \in V} \sum_{k=1}^{n} t_{ik} \tag{33}$$

$$\text{s.t.} \sum_{j \in N(i)} y_{ij} \leq \sum_{k=1}^{n} L_k t_{ik}, \ \forall i \in V \tag{34}$$

$$\sum_{k=1}^{n} t_{ik} \leq 1, \qquad \forall i \in V \tag{35}$$

and constraints (31)

4.2. Basic scheme

As the size of $\mathcal{I}(\text{PPP}_{Bi})$ is $O(2^{|V|})$, a natural way to solve the problem is through a cutting plane approach. Algorithm 4.2.1 describes the initial cutting plane method implemented to obtain an optimal PMU placement (see its associated flowchart in Fig. A.11). Step 2 builds the initial subset of valid inequalities and $\text{PPP}_{\bar{B}i}$ is solved restricted to it (Step 3). Then, while the PMU placement of iteration k, x^k, is not feasible (Step 4), constraint (29) associated with the nodes observed by x^k is added (Step 5).

Algorithm 4.2.1 Basic scheme.

Input: A graph $G = (V, E)$ and a set of zero-injection nodes S.
Output: Optimal PMU placement.
1: $k \leftarrow 1$
2: Initialize $\mathcal{I}(\text{PPP}_{Bi})$ with $\sum_{i \in V} x_i \geq 1$, *i.e.*, there must be at least one PMU
3: $x^k \leftarrow$ optimal solution to $\text{PPP}_{\bar{B}i}$ with $\mathcal{I}(\text{PPP}_{Bi})$.
4: **while** $\sum_{i \in V} w_i(x^k) < |V|$ **do**
5: Add Constraint (30) in x^k to $\text{PPP}_{\bar{B}i}$ /* update $\mathcal{I}(\text{PPP}_{Bi})$ */
6: $k \leftarrow k + 1$
7: Solve $\text{PPP}_{\bar{B}i}$ and let x^k be the optimal solution
8: **end while**
9: **return** x^k

Algorithm 4.2.1 is also valid to solve the problem variants $\text{PPP}^L_{\bar{B}i}$ and $\text{PPP}^t_{\bar{B}i}$, if $\text{PPP}_{\bar{B}i}$ is replaced by $\text{PPP}^L_{\bar{B}i}$ or $\text{PPP}^t_{\bar{B}i}$, and constraints (29) by constraints (30).

4.3. Computing observed nodes

To compute the nodes observed by a PMU placement, Poirion et al. (2016) show that in the bilevel formulation of their PPP, the binary requirement in the follower's variables, w, can be relaxed. Instead, given a PMU placement, we propose a polynomial time algorithm (Algorithm 4.3.1) in the input size of G to determine the observed nodes. Its flowchart is available in Appendix A, Fig. A.12.

The algorithm first applies rule 1 to nodes with a PMU (Steps 2–7). Then, since rule 2 and 3 can be used at most one time for each node in S, we create the set S' that saves the zero-injection nodes for which neither of those rules were applied yet (Step 8). Rules 2 and 3 are then applied, when adequate (Steps 13–17 and 19–23, respectively) and, whenever used in a zero-injection node i, the node is removed from S' (Steps 19 and 23, respectively). The process repeats while there is an update of the observed nodes (the variable *end* encodes the existence of an update).

Algorithm 4.3.1 runs in polynomial time as shown next. Steps 2–7 run in $O(|E|)$: note that we cycle in V and if rule 1 is applicable, we cycle in the neighbor nodes. In each iteration of Step 9,

Algorithm 4.3.1 Computation of the nodes observed.

Input: A graph $G = (V, E)$, a set of zero-injection nodes S and a PMU placement vector $x \in \{0, 1\}^{|V|}$.
Output: Vector $w \in \{0, 1\}^{|V|}$ of nodes observed

1: $w \leftarrow (0, 0, \ldots, 0)$
/* *Apply rule 1* */
2: **for** $i \in V$ **do**
3: **if** $x_i = 1$ **then**
4: $w_i \leftarrow 1$
5: $w_j \leftarrow 1$ for $j \in N(i)$
6: **end if**
7: **end for**
8: $end \leftarrow True$ and $S' = copy(S)$
9: **while** end **do**
10: $end \leftarrow False$ and $S'' = copy(S')$
11: **for** $i \in S''$ **do**
12: **if** $w_i = 0$ **then**
13: **if** $\sum_{j \in N(i)} w_i = |N(i)|$ **then**
14: $w_i \leftarrow 1$ /* *Apply rule 2* */
15: $end \leftarrow True$ /* *There was an update in the set of nodes observed* */
16: $S' \leftarrow S' \setminus \{i\}$
17: **end if**
18: **else**
19: **if** $\sum_{j \in N(i)} w_j = |N(i)| - 1$ **then**
20: $w_j = 1$ for $j \in N(i)$ /* *Apply rule 3* */
21: $end \leftarrow True$ /* *There was an update in the set of nodes observed* */
22: $S' \leftarrow S' \setminus \{i\}$
23: **end if**
24: **end if**
25: **end for**
26: **end while**
27: **return** w

we check the applicability of rules 2 and 3 for each zero-injection node in S', and this step continues to the next iteration if these rules where applied. In this way, we conclude that the algorithm runs in time $O(|E|^2)$.

Algorithm 4.3.1 is also valid for the variants described in Section 2.3 by adapting steps from 2 to 7 as follows: the cycle will run for edges $(i, j) \in E$ and not for nodes, and will verify if $y_{ij} = 1$ (instead of $x_i = 1$) to apply rule 1 for the observability of nodes i and j.

4.4. Improving the valid inequalities for the cutting plane algorithm

In this section, we present a polynomial algorithm that uses an intermediate solution x^k computed in Algorithm 4.2.1 to improve its efficiency. Succinctly, constraints (29) are improved and an upper bound to the optimal number of PMUs is computed.

With this purpose, we say that a PMU placement $\bar{x} \in \mathcal{I}(\text{PPP}_{Bi})$ is *maximal* if, for all $i \in \mathcal{I}(\bar{x})$ and $\bar{x}_i = 0$, the addition of a PMU in node i to $\bar{x}$ results in a feasible solution.

Lemma 4.2. *If $\bar{x} \in \mathcal{I}(\text{PPP}_{Bi})$ is not maximal then, there is a $\tilde{x}$ that is maximal such that $\mathcal{I}(\tilde{x}) \subseteq \mathcal{I}(\bar{x})$. Furthermore, constraint (29) in $\bar{x}$ is dominated by the one in $\tilde{x}$.*

Proof. If $\bar{x} \in \mathcal{I}(\text{PPP}_{Bi})$ is not maximal, in order to find its maximal PMUs must be added to $\bar{x}$ for nodes in $\mathcal{I}(\bar{x})$ until it becomes maximal; denote the resulting placement by $\tilde{x}$. By construction of $\tilde{x}$, $\mathcal{I}(\tilde{x}) \subseteq \mathcal{I}(\bar{x})$. Thus, constraint (29) in $\tilde{x}$ dominates the one in $\bar{x}$. □

We remark that if $i \notin \mathcal{I}(\bar{x})$ and $\bar{x}_i = 0$, since the nodes $\{i\} \cup N(i)$ are observed, the addition of a PMU in node i to $\bar{x}$ does not change the observability $w(\bar{x})$, and thus does not improve constraint (29).

The following example illustrates this idea.

Algorithm 4.4.1 Computation of a maximal PMU placement and of an upper bound. For the L-channel and variable cost variants, the instruction in gray should replace the ones in black, for the same step.

Input: A graph $G = (V, E)$, a set of zero-injection nodes S, the set S_0, the set U_0, the set $U \setminus U_0$ and a PMU placement x^k.
A graph $G = (V, E)$, a set of zero-injection nodes S, the set S_0, the set U_0, the set $U \setminus U_0$ and a PMU placement x^k and the associated use of capacities y^k.
Output: A maximal PMU placement for x^k if x^k is infeasible and an upper bound x^{UB}.
A maximal PMU placement for y^k if y^k is infeasible and an upper bound x^{UB}.

1: Sort the nodes $V = \{v_1, v_2, \ldots, v_{|V|}\}$ such that the ones with highest index are in S_0 and the ones with lowest index are in $U \setminus U_0$
Sort the edges $E = \{e_1, e_2, \ldots, e_{|E|}\}$ such that the edges incident with nodes in S_0 and in U_0 have the highest index, and the ones with lowest index are in $U \setminus U_0$
2: $aux_{UB} = True$
3: **if** $\sum_{i \in V} w_i(x^k) = |V|$ **then**
if $\sum_{i \in V} w_i(y^k) = |V|$ **then**
4: **return** x^k, x^k
return y^k, (y^k, x^k)
5: **else**
6: $i \leftarrow 1$
7: **while** $i \leq |V|$ **do** /* *check if* $i \in \mathcal{I}(x^k)$*/
while $e \leq |E|$ **do** /* *check if* $e \in \mathcal{I}(y^k)$*/
8: **while** $w_i(x^k) + \sum_{j \in N(i)} w_j(x^k) = |N(i)| + 1$ **do**
while $w_i(y^k) + w_j(x^k) = 2$ **do** /* $e = (i, j)$*/
9: $i \leftarrow i + 1$
$e \leftarrow e + 1$
10: **if** $i = |V| + 1$ **then**
if $e = |E| + 1$ **then**
11: **return** x^k, x^{UB}
return y^k, (y^{UB}, x^{UB})
12: **end if**
13: **end while**
14: $x_i^k \leftarrow 1$
Assume i is the node without a PMU or with capacity not fully used, make $y_{i,j}^k \leftarrow 1$ and $x_i^k \leftarrow 1$
15: **if** $\sum_{i \in V} w_i(x^k) = |V|$ **then**
if $\sum_{i \in V} w_i(y^k) = |V|$ **then**
16: **if** aux_{UB} **then**
17: $x^{UB} \leftarrow x^k$ and $aux_{UB} \leftarrow False$
$y^{UB} \leftarrow y^k$, $x^{UB} \leftarrow x^k$ and $aux_{UB} \leftarrow False$
18: **end if**
19: $x_i^k \leftarrow 0$
$y_{ij}^k \leftarrow 0$; if a PMU was added to node i step 14 $x_i^k \leftarrow 0$
20: **end if**
21: $i \leftarrow i + 1$
$e \leftarrow e + 1$
22: **end while**
23: **return** x^k, x^{UB}
24: **end if**

Example 1. Consider the graph instance of Fig. 7 and the infeasible solution with only one PMU in node 4. The associated con-

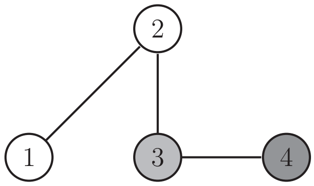

Fig. 7. PMU placement $\bar{x} = (0, 0, 0, 1)$. The dark gray node is observed. The light gray node becomes observed by rule 1. The remaining nodes are not observed.

straint (29) would be $x_1 + x_2 + x_3 \geq 1$ since $\mathcal{I}(\bar{x}) = \{1, 2, 3\}$. If a PMU is added to nodes 1 or 2, full observability is attained. On the other hand, adding a PMU to node 3 maintains infeasibility. Thus, by considering solution $\tilde{x} = (0, 0, 1, 1)$, we improve the constraint (29) which becomes $x_1 + x_2 \geq 1$.

To consider this extra ingredient, in step 5 of Algorithm 4.2.1, we introduce the computation of a maximal PMU placement for x^k. To that end, we use Algorithm 4.4.1 (see its flowchart in Fig. A.13) which makes x^k maximal: step 8 adds PMUs in nodes $i \in \mathcal{I}(x^k)$ and step 15 determines if the new x^k is feasible. If yes, step 19 removes the last added PMU. In a completely analogous way, (see instructions in grey in Algorithm 4.4.1) we improve constraints (30) for the variants of the problem by adding to x^k the information in y^k and to x^{UB} the information in y^{UB}. For doing so we replace step 1 of Algorithm 4.4.1 in order to sort the edges $E = \{e_1, e_2, \dots, e_{|E|}\}$ such that the edges incident with nodes in S_0 and in U_0 have the highest index, and the ones with lowest index are in $U \backslash U_0$. We also change the condition in $|V|$ in step 8 by $|E|$ and the condition in step 8 by $w_i(y^k) + w_j(y^k) = 2$. In step 14 we add a PMU to node i, $x_i^k \leftarrow 1$, and use the capacity to observe node j, $y_{ij}^k \leftarrow 1$, if node i had no PMU installed or still had capacity to be used (otherwise, do that in node j); we update in step 19 also the variable y^k.

Poirion et al. (2016) also improve their inequalities by adding to an infeasible solution PMUs in non-observed nodes. Our algorithms consider a larger set of nodes to add a PMU: not only non-observed nodes, but also observed nodes with at least a neighbor not observed. Thus, our approach can lead to dominant constraints.

4.5. Additional stopping criterion

While making an infeasible solution x^k maximal, at least one feasible solution is determined which gives us an upper bound : step 15 of Algorithm 4.4.1 identifies feasible solutions. Moreover, we would like to save the best upper bound during the algorithm execution; that is done through Step 16 of Algorithm 4.4.1, since the first feasible solution computed by the algorithm is saved. According with Step 1 of Algorithm 4.4.1, nodes in $U \backslash U_0$ are more likely to have a PMU than the ones in S_0. This step is also an attempt to have a better upper bound (recall that supporting nodes with a PMU eliminate the need of PMUs in the associated zero-injection paths).

This upper bound together with the infeasible solution (lower bound) provide information on how far we are from attaining the optimum. Moreover, once we compute an upper bound in iteration k that uses the same number of PMUs as x^k (lower bound), we can stop the algorithm, because the upper bound is an optimal solution.

4.6. Final algorithm description

The new ingredients introduced in Sections 4.4 and 4.5 are incorporated in Algorithm 4.2.1, leading to a final finely tuned algorithm, whose pseudo-code is displayed in Algorithm 4.6.1. The corresponding flowchart is in Fig. A.14. The algorithm is initialized in step 2 with the upper bound given by the trivial feasible solution where all nodes have a PMU. Then, as in Algorithm 4.2.1, while x^k does not lead to full observability (step 8), which is computed through Algorithm 4.3.1, run Algorithm 4.4.1 which computes an upper bound, x^{UB}, and makes x^k maximal (step 9). If the new upper bound x^{UB} is better than the incumbent x^{best}, update it (step 10). Next, in step 13, the stopping criterion discussed in Section 4.5 is applied.

Algorithm 4.6.1 Improved cutting plane.

Input: A graph $G = (V, E)$ and a set of zero-injection nodes S.
Output: Optimal PMU placement.
1: $k \leftarrow 1$
2: $x^{best} \leftarrow (1, 1, \dots, 1)$
3: Compute S_0, U_0 and U.
4: Initialize a set of infeasible solutions $\mathcal{I}(\text{PPP}_{Bi})$ (*e.g.*, there must be at least one PMU, *i.e.*, $\sum_{i \in V} x_i \geq 1$.)
5: ${\text{PPP}_{\tilde{B}i}}^k \leftarrow \text{PPP}_{\tilde{B}i}$ restricted to constraints (30) in $\mathcal{I}(\text{PPP}_{Bi})$ and including rule 1 initial cuts and variable fixing of Section 3
6: $x^k \leftarrow$ optimal solution to ${\text{PPP}_{\tilde{B}i}}^k$
7: $LB \leftarrow \sum_{i \in V} x_i^k$
8: **while** $\sum_{i \in V} w_i(x^k) < |V|$ **do** /* *run algorithm 4.3.1*/
9: $x^k, x^{UB} \leftarrow$ algorithm 4.4.1
10: **if** $\sum_{i \in V} x_i^{UB} < \sum_{i \in V} x_i^{best}$ **then**
11: $x^{best} \leftarrow x^{UB}$
12: **end if**
13: **if** $\sum_{i \in V} x_i^{best} \leq LB$ **then**
14: **return** x^{best}
15: **end if**
16: Add constraint (30) in x^k to ${\text{PPP}_{\tilde{B}i}}^k$ // update $\mathcal{I}(\text{PPP}_{Bi})$
17: $k \leftarrow k + 1$
18: $x^k \leftarrow$ optimal solution to ${\text{PPP}_{\tilde{B}i}}^k$
19: $LB \leftarrow \sum_{i \in V} x_i^k$
20: **end while**
21: **return** x^k

5. Computational results

In this section we summarize the computational experiments that were carried out to validate the effectiveness of the models proposed, and of the algorithms designed. First we compare our results with the ones known from the literature. To the best of our knowledge the only work that properly deals with rule 3 is Poirion et al. (2016). Some approaches consider only Ohm's law when proposing observability rules (Sodhi et al., 2010), or do not propagate the rules correctly (Dua et al., 2008; Fan and Watson, 2015; Xu and Abur, 2004), thus we are not able to compare with them. Moreover, although our focus is on the comparison of exact approaches, we also refer to the extensive literature in meta-heuristics considering zero-injection nodes. Second, we present the results for the general models and validate the improvements proposed in Section 3 and Sections 4.3–4.6.

5.1. Test instances and implementation details

The computational analysis will be done in two sets of instances available in the literature. We will refer to them as *Random* and *IEEE instances*.

Table 1
Comparison of different approaches for the random instances from Poirion et al. (2016), 1-channel PPP.

		Model (Poirion et al., 2016)		PPP^{L}_{OPM}		PPP^{L}_{OPM} +VIfix		CP (Poirion et al., 2016)		BS 4.2.1			Alg 4.6.1			Alg 4.6.1 & Poirion et al. (2016)		
$\|V\|$	$\|S_0\|$	*gap*	*T*	*gap*	*T*	*gap*	*T*	*T*	*it*	*T*	*it*	it_{un}	*T*	*it*	it_{UB}	*T*	*it*	it_{UB}
5	1.6		0.09		0.15		0.05	0.00	3.1	0.00	3.3		0.00	2.4	1.1	0.00	2.3	1.1
10	2.3		1.48		11.88		1.16	0.01	7.9	0.01	10.3		0.01	6.6	1.6	0.00	7.1	1.5
15	3.9		11.18	92.8	(1) 671.4		23.41	0.01	10.1	0.01	17.4		0.01	7.6	4.3	0.01	6.7	2.2
20	4.7	33.3	(9) 257.99	100.0	tl	54.2	(8) 507.91	0.04	17.3	0.19	47.3		0.02	9.7	4.8	0.02	9.6	5.3
25	6.3	37.5	(6) 115.18	97.5	tl	93.7	(1) 1405.22	0.12	29.3	1.11	94.4		0.12	19.6	7.5	0.08	20.4	7.0
30	8.2	57.9	tl	97.5	tl	97.5	tl	0.56	78.2	8.09	269.0		0.30	32.0	4.2	0.20	32.4	4.6
35	10.3	65.8	tl	100.0	tl	95.0	tl	0.60	65.5	21.13	393.1		0.31	27.8	17.8	0.26	30.1	18.9
40	10.3	-	–	–	–	–	–	4.07	114.7	129.30	836.4		0.96	44.9	34.6	0.58	44.0	33.0
45	14	-	–	–	–	–	–	12.73	197.7	(6) 372.62	1251.7	2056.5	1.79	65.4	27.0	1.16	65.9	26.8
50	13.4	-	–	–	–	–	–	170.37	448.0	(1) 618.15	1872.0	2094.8	9.75	117.1	36.4	5.74	113.4	30.5

Table 2
Comparison of different approaches for IEEE instances, 1-channel PPP.

				Model (Poirion et al., 2016)		PPP^{L}_{OPM}		PPP^{L}_{OPM} +VIfix		CP (Poirion et al., 2016)			BS 4.2.1			Alg 4.6.1				Alg. 4.6.1 & Poirion et al. (2016)			
$\|V\|$	$\|S\|$	$\|S_0\|$	*opt*	*gap*	*T*	*gap*	*T*	*gap*	*T*	*gap*	*T*	*it*	*LB*	*T*	*it*	*gap*	*T*	*it*	it_{UB}	*gap*	*T*	*it*	it_{UB}
9	3	0	3		0.07		0.05		0.05		0.01	11	3	0.05	32		0.01	7	3		0	7	2
14	1	0	7		0.02		0.01		0.01		0.01	19	7	774.73	1776		0.01	5	1		0	5	1
24	4	1	10		0.45		1.17		0.15		0.05	38	3	tl	1714		0.02	14	13		0.03	22	22
30	6	0	13		2.89		6.38		1.86		0.07	46	3	tl	1853		0.05	21	14		0.03	25	13
39	12	2	14	7.14*	tl		70.29		13.5		0.11	51	3	tl	2116		0.05	23	20		0.07	34	31
57	15	7	21		194.11	14.28*	tl		13.65		0.61	92	2	tl	1896		0.38	53	53		0.11	43	43
73	13	4	30		598.46	10*	tl		1418.06		3.48	142	2	tl	1701		8.09	83	83		0.95	70	69
118	10	3	56		9.85		7.2		7.37		1.87	179	2	tl	1387		0.52	40	40		0.27	39	39
150	33	1	66	49.76*	tl	54.93	tl	27.06	tl	21.74	tl	1648	2	tl	1364	15.71	tl	647	254	7.35	tl	1059	888
300	68	24	127	26.7*	tl	29.55	tl	24.43	tl		95.65	754	2	tl	1276		14.33	210	187		7.8	198	159
2007	308	30	1002	–	–	–	–	–	–	91.37	tl	391	1	tl	841	21.91	tl	855	554	21.46*	tl	996	897

- **Random instances** were generated by Poirion et al. (2016) for networks with 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 buses, 10 instances for each network size. All nodes are zero-injection.
- **IEEE instances** are standard power network benchmarks instances of size 9, 14, 24, 30, 39, 57, 73, 118, 150, 300, 2007[1] available from Illinois center for a smarter electric grid. (2017) . The set of zero-injection nodes is known for each instance.

All algorithms have been coded in Python 2.7.2 and Gurobi 7.0.1. was used for solving the integer programming problems. The experiments were conducted on a Quad-Core Intel Xeon processor at 2.66 GHz, running under Mac OS X 10.8.4. Only one thread was assigned to any run of any approach. Unless explicitly stated we consider a CPU time limit of 1800 s for any run.

The following general notation is used in the tables that report the computational results.

- T – CPU time (in seconds); "tl" means that the time limit was reached;
- *opt* – optimal value or best found upper bound;
- *gap* – gap between the lower and the upper bounds (*LB* and *UB*, respectively) at the end of the computations, $gap = \frac{UB-LB}{UB} \times 100\%$; an empty entry means the instance (or all instances) was solved ($gap = 0$), and "-" means that the value is not available; for the IEEE instances, when *UB* coincides with the value in column *opt*, an asterisk is added (*e.g.* 21.46*);
- *LB* – final lower bound for the IEEE instances for the algorithmic approaches that do not provide the upper bound at the end of the computations;
- *it* – number of iterations performed;
- it_{UB} – iteration in which the final upper bound or optimum was found;
- it_{un} — number of iterations performed in average for the random instances when the instances where not solved to optimality within the time limit;
- $|S_0|$ – number of nodes in supported zero-injection paths (see Propositions 2, 3 and 5);
- $|V_0|$ – number of fixed non zero-injection nodes according to Proposition 1.

Noteworthy is that for all the tests done the preprocessing time for finding zero-injection paths and solving problem (25)–(27) was negligible (at most 0.2 s for IEEE instance with $|V| = 2007$). Therefore it is not presented in the corresponding tables.

For the random instances we present average values for each 10 instances of the same size. Moreover, when necessary, we present in column T in parenthesis the number of instances out of 10 solved to optimality withing the time limit (when omitted, then all the instances of a set where solved); 'tl' means that none of the 10 instances was solved).

5.2. Results for the 1-channel PPP

In this section we compare our results with the ones presented in Poirion et al. (2016). Since they solved the 1-channel PMU problem considering only rules 1 and 3, for a fair comparison we implemented their observability propagation model and cutting plane approach for the bilevel model including propagation rule 2.

We compare different observability propagation models: the model in Poirion et al. (2016), our model (PPP^{L}_{OPM}) for $L = 1$, and the same model strengthened by the valid inequalities and variables fixing (Section 3) (we will refer to it as PPP_{It}+VIfix). For the bilevel model four algorithmic approaches were compared: the cutting plane method from Poirion et al. (2016) (designated by CP Poirion et al., 2016), the basic scheme described in Algorithm 4.2.1 (from now on referred to as BS 4.2.1), Algorithm 4.6.1 (Alg 4.6.1), and Algorithm 4.6.1 adapted to the bilevel model in Poirion et al. (2016) (Alg 4.6.1 & Poirion et al., 2016).

Tables 1 and 2 provide the computational results for the class of randomly generated and IEEE instances, respectively. In what concerns the observability propagation models for random instances (Table 1), they are not competitive with any of the algorithmic approaches presented: none of the models was able to solve any instance of 30 buses. For bigger instances with 40 buses and more the models have not been run, as it would go over the time limit for all instances. This poor results for the observability propagation models were somehow expected due to the fact that the size of the models grows dramatically with the size of the set of zero-injection nodes, which for this instances is $S = V$.

For the cutting plane approaches, all methods except for BS 4.2.1 were able to solve all the instances. Despite some failures (see instances with 45 and 50 buses) the BS 4.2.1 still performs significantly better than the propagation models. Notice that in the computational experiments presented in Poirion et al. (2016) some of the instances were not solved. One of the explanations for this to happen could be the fact that in our implementation of CP (Poirion et al., 2016) we introduced rule 2, that contributes to the potential reduction on the number of PMUs needed and thus, since in each iteration of the CP (Poirion et al., 2016) lower bounds are being computed, these bounds are more likely to be close to the optimal solution. The impact of our approach is strongly highlighted by the results for the fully tuned version of our algorithmic approach. Algorithm 4.6.1 uses in average less time (column T) and needs less iterations (*it*) to solve an instance, than CP (Poirion et al., 2016) (*e.g.*, for $|V| = 50$ the reduction in CPU time is around 18 times and the number of iterations is 3 times less than for CP Poirion et al., 2016).

For the IEEE instances (see Table 2), if we compare the results for PPP_{It} and the model in Poirion et al. (2016), our model looses both in number of solved instances (see instances with 57 and 73 buses) and in the quality of the solutions (compare *gap* for instances 150 and 300). Again, these results are not unexpected since we are comparing a general model with another that, being an instantiation of ours, can be better for the particular case of $L = 1$, but not applicable for other problem variants. The only exception is instance 39, that was not solved by the observability propagation model in Poirion et al. (2016), and was solved by our model. Still, using the valid inequalities and variable fixing proposed in Section 3 of this paper, it was possible to improve the performance of our model. It slightly improved the results for instances 118, and was significantly faster for instances 39, 57. It also presented better gaps for unsolved instances 150, 300. The only exception is instance 73 where the time is significantly bigger than for model (Poirion et al., 2016). Note that due to the size of the propagation models, instance 2007 was not considered. The models were not even loaded to the solver within the time limit.

When comparing the observability propagation models with the algorithmic approaches used to solve the bilevel problem, the models perform worse than three of the four algorithmic approaches. The exception is the basic scheme that is extremely ineffective and solved only instances 9 and 14, and provided utterly bad lower bounds for other instances. When comparing the three remaining algorithms, both Alg 4.6.1 and Alg 4.6.1 & Poirion et al. (2016) improve the number of iterations needed to find the optimum, in comparison with CP (Poirion et al., 2016). Furthermore, even for the two non-solved instances, instances 150 and 2007, our methods terminate with a better *gap* in less iterations. Noteworthy is that, although Alg 4.6.1 needs less iterations to find the optimum when compared to CP (Poirion et al., 2016), it is more time consuming. However, Alg 4.6.1 & Poirion et al. (2016) is

[1] Exceptionally, the instance with $|V| = 73$ is known by IEEE 96-bus.

Table 3
Comparison of methods from the literature addressing the uncapacitated PPP. Values in bold correspond to optimal solutions and * to methods with observability rules differing from ours.

Method	Instances									
	IEEE 14		IEEE 30		IEEE 39		IEEE 57		IEEE 118	
	#PMU	*T*	*#PMU*	*T*	*#PMU*	*T*	*#PMU*	*T*	*#PMU*	*T*
Integer Programming (Xu and Abur, 2004)	**3**	2.04	–	–	–	–	12	4.15	**29**	43.96
Integer Programming (Dua et al., 2008)	**3**	–	-	–	–	–	14	–	**29**	0.031
Genetic Algorithm (Mohammadi-Ivatloo, 2009)	**3**	–	**7**	–	9	–	12	–	**29**	–
Genetic Algorithm (Marín et al., 2003)	**3**	–	**7**	–	–	–	12	–	**29**	–
Parallel Tabu Search (Mesgarnejad and Shahrtash, 2008)	**3**	2172.59	-	–	–	–	–	–	–	–
Recursive Tabu Search (Koutsoukis et al., 2013)	**3**	0.01	**7**	0.33	**8**	0.57	**11**	2.11	**28***	29.12
Immunity Genetic Algorithm (Aminifar et al., 2009)	**3**	2.00	**7**	4.00	–	–	**11**	11.00	**29**	72.00
Particle Swarm Optimization (Ahmadi et al., 2011)	**3**	–	**7**	–	–	–	13	–	**29**	–
Particle Swarm Optimization (Hajian et al., 2007)	**3**	–	**7**	–	–	–	**11**	–	**28***	–
Simulated Anneling (Nuqui and Phadke, 2005)	**3**	-	**7**	–	–	–	**11**	–	–	–
Chemical Reaction Optimization (Xu et al., 2013)	**3**	0.01	**7**	0.04	–	–	14	0.13	**29**	0.44

faster than CP (Poirion et al., 2016), which indicates that Alg 4.6.1 consumes more CPU time mainly because it has a duplication in the number of decision variables, and not because of the improvements that we added. Recall that this duplication is due to the fact that our bilevel model is a generalization of the one in Poirion et al. (2016). To conclude this analysis, columns it_{UB} show that the best upper bound is usually found in the last iterations, before the algorithm stops.

5.3. Results for the uncapacitated PPP

In this section we discuss the results obtained for the general uncapacitated PPP. This problem was preliminary addressed by Sodhi et al. (2010), where IEEE instances 14 and 39 were solved using rule 1 (withouth consideration of zero-injection nodes). Although for $|V|=14$ their optimum coincides with ours, for $|V|=39$ they get 13 PMUs while we get 8.

Table 3 presents relevant computational results from the literature, both for exact and non-exact approaches, when zero-injection nodes are considered. For several reasons, explicited in the following text, those results cannot be compared with the ones provided by our work. Xu and Abur (2004) and Dua et al. (2008) do not seem to properly propagate the observability rules, which justifies the fact that the results for instance 57 are not optimal. In the Genetic Algorithm's approaches proposed by Mohammadi-Ivatloo (2009) and Marín et al. (2003) computational times are not provided, preventing comparison of results. Furthermore, Mesgarnejad and Shahrtash (2008) present a parallel Tabu Search method which is only tested in instance 14, presenting a long running time. On the other hand, the Tabu Search approached by Koutsoukis et al. (2013) is extremely fast achieving optimal solutions. However, the observability rules that they use differ from ours, preventing again comparison. The difference in rules explains the optimal number of PMUs for instance 118 achieved by those authors. Again, computational times are not provided in Ahmadi et al. (2011), Hajian et al. (2007) and Nuqui and Phadke (2005). Regarding the Chemical Reaction method proposed by Xu et al. (2013), which runs in less than 1 second, it is not capable of getting the optimal solution for instance 57.

Due to the lack of information in the literature for a proper comparison of results in this section we only compare the performance of the approaches that we propose. Main advantage of our approach over heuristics is that it guarantees attainment of optimal solutions.

We start by presenting the results for the PPP with no limitation on capacity. Similarly to the previous computational experiments (Section 5.2), we will compare the observability propagation model (PPP_{It}) with an improved version of the model (PPP_{It}+VIfix). For the cutting plane approaches we will carry out computations for the following algorithm variants:

1. Basic scheme (BS 4.2.1) as in Section 5.2;
2. Basic scheme with valid inequalities and variables fixing (BS+VIfix);
3. Basic scheme with the improved inequalities introduced in Section 4.4.1 (BS+Ineq.);
4. Variant 3 with the stopping criterion presented in Section 4.5 (BS+Ineq.+stop);
5. Finely tuned Algorithm 4.6.1, considering all these improvements together.

Table 4 summarizes the results for the random instances. In order to evaluate the impact of zero-injection nodes, we consider three options: *(i)* $S=\emptyset$; *(ii)* $|S|=\lfloor\frac{|V|}{2}\rfloor$ and *(iii)* $S=V$. These sets of zero injection nodes were generated by Python's Random module (see Foundation, 2012) and are available upon request.

Similarly to the experiments for the problem with $L=1$ presented before, the observability propagation models are again ineffective for this set of instance when the number of zero-injection nodes is large. If $S=V$ some instances with $25\leq|V|\leq40$ were not solved. For $|V|\geq45$ none of the instances were solved. If $|S|=\lfloor\frac{|V|}{2}\rfloor$ there are some unsolved instances of size 45 and 50. Still the observability propagation model performs better than the basic scheme when there are no zero-injection nodes. When valid inequalities and variable fixing are included in the model (PPP_{It}+VIfix), we may observe a slight decrease on the CPU time (*e.g.*, compare the results for $|V|=40,\ |S|=20$), or an improvement in the number of instances solved (*e.g.*, $|V|=45,\ |S|=22$).

Once again, in general the cutting plane algorithms are more effective for solving the problem. Although there are some instances unsolved with BS for $|V|\geq25$, just by inclusion of valid inequalities and variable fixing (BS+VIfix) we are able to solve all the instances to optimality within the time limit. Moreover, when only valid inequalities are considered (BS+ineq.) the decrease of CPU time is even more drastic, especially for $|S|=\lfloor\frac{|V|}{2}\rfloor$ (see instances with $|V|\geq35$). Finally the inclusion of all the improvements in Algorithm 4.6.1 allowed us to solve all the instances in less than 0.1 s in average, the number of iterations being also small. For the instances with $|V|\geq25$, it is evident that the increase in the number of zero-injection nodes increases the computational times. On the other hand, the instances with $\lfloor\frac{|V|}{2}\rfloor$ zero-injection nodes need in average more iterations to be solved than the ones where all nodes are zero-injection.

Table 5 presents results for the IEEE instances. The observability propagation model works relatively well for these instances, though it is not able to solve the two biggest ones, and it took more than 1000 s to solve the problem with 150 buses with the

Table 4
Comparison of different approaches for randomly generated instances from Poirion et al. (2016), PMUs with unlimited capacity.

				PPP_{lt}		PPP_{lt}+VIfix		BS 4.2.1			BS 4.2.1 +VIFix		BS 4.2.1 +Ineq. 4.4.1		BS 4.2.1 +Ineq. 4.4.1			Algorithm 4.6.1 +stop 4.5		
$\lvert V\rvert$	$\lvert S\rvert$	$\lvert S_0\rvert$	$\lvert V_0\rvert$	*T*	*gap*	*T*	*gap*	*T*	*it*	it_{un}	*T*	*it*	*T*	*it*	*T*	*it*	it_{UB}	*T*	*it*	it_{UB}
5	0	0	1.6	0		0		0.00	4.0		0.00	1.0	0.00	3.0	0.00	2.7	1.0	0.00	1.0	1.0
	2	0.4	1.2	0.01		0.01		0.00	2.4		0.00	1.9	0.00	2.3	0.00	2.1	1.0	0.00	1.1	1.0
	5	1.6	0	0.03		0.03		0.00	2.1		0.00	2.1	0.00	2.0	0.00	2.1	1.0	0.00	1.0	1.0
10	0	0	1	0		0		0.02	26.1		0.00	1.0	0.00	5.5	0.00	5.2	2.4	0.00	1.0	1.0
	5	1.4	0.5	0.09		0.08		0.00	7.0		0.00	4.3	0.00	3.5	0.00	3.2	1.6	0.00	2.1	1.0
	10	2.5	0	0.29		0.31		0.00	3.2		0.00	2.5	0.00	2.6	0.00	2.0	1.0	0.00	1.5	1.0
15	0	0	0.8	0		0.01		0.32	110.0		0.00	1.0	0.00	8.5	0.00	7.8	4.2	0.00	1.0	1.0
	7	2.1	0.5	0.53		0.31		0.01	15.9		0.01	11.8	0.00	5.5	0.00	5.3	2.4	0.00	4.9	2.1
	15	3.9	0	5.09		6.36		0.00	4.2		0.00	3.2	0.00	2.9	0.00	2.5	1.3	0.00	2.2	1.3
20	0	0	0.8	0.01		0.01		63.29	713.9		0.00	1.0	0.01	11.0	0.00	10.0	7.2	0.00	1.0	1.0
	10	2	0.1	0.99		0.9		0.05	45.6		0.01	15.8	0.00	6.6	0.00	6.5	4.3	0.00	5.3	3.1
	20	4.7	0	193.75		173.65		0.00	7.6		0.00	7.0	0.01	4.1	0.00	3.2	1.4	0.00	3.1	1.4
25	0	0	1	0.01		0.01		(8) 768.21	2747.5	3534.5	0.00	1.0	0.01	13.9	0.01	13.2	9.9	0.00	1.0	1.0
	12	3	0.2	1.45		1.4		0.70	139.8		0.07	36.6	0.01	9.5	0.01	9.8	6.5	0.01	7.1	3.6
	25	6.3	0	(8) 240.19	99.95	(8) 187.91	98.07	0.00	9.6		0.00	7.4	0.01	5.4	0.01	4.8	1.5	0.01	4.8	1.5
30	0	0	0.4	0.01		0.01		tl	–	2990.3	0.00	1.0	0.01	15.8	0.01	15.0	11.5	0.00	1.0	1.0
	15	3.8	0.3	9.03		6.83		5.04	309.5		0.47	67.9	0.01	10.3	0.01	10.2	6.9	0.01	7.3	4.9
	30	8.2	0	(3) 669.56	100	(3) 713.9	100	0.01	15.6		0.01	13.9	0.02	7.7	0.02	6.5	2.5	0.02	6.5	2.5
35	0	0	0.9	0.01		0.01		tl	–	2294.1	0.00	1.0	0.02	20.5	0.02	20.2	15.7	0.00	1.0	1.0
	17	5.3	0.4	29.04		22.83		109.91	994.8		2.93	178.3	0.02	12.1	0.01	11.0	8.4	0.01	8.6	5.5
	35	10.3	0	(4) 921.9	100	(4) 561.84	100	0.02	21.9		0.01	16.3	0.03	7.2	0.02	5.5	2.6	0.02	5.5	2.6
40	0	0	1	0.01		0.01		tl	–	2186.3	0.00	1.0	0.02	22.9	0.02	22.3	17.8	0.00	1.0	1.0
	20	5.4	0.6	285.77		171.61		(9) 395.41	1747.9	2782.0	12.85	215.7	0.02	13.9	0.02	12.6	9.2	0.01	8.4	5.5
	40	10.3	0	(1) 369.85	92.59	(1) 835.45	87.04	0.06	42.5		0.05	37.4	0.05	9.8	0.04	8.4	2.6	0.04	8.4	2.6
45	0	0	1.1	0.01		0.01		tl	–	2036.9	0.00	1.0	0.02	25.6	0.02	24.7	20.9	0.00	1.0	1.0
	22	7.1	0.6	(8) 289.2	28.55	(9) 301.92	28.57	(2) 833.13	2448.0	3034.0	9.71	262.6	0.03	16.1	0.03	15.3	11.4	0.02	10.4	6.8
	45	14	0	tl	100	(1) 880.54	92.59	0.08	45.0		0.09	41.6	0.06	9.6	0.06	9.8	3.8	0.06	9.8	3.8
50	0	0	0.4	0.01		0.02		tl	–	2095.2	0.00	1.0	0.03	29.2	0.03	28.4	24.4	0.00	1.0	1.0
	25	7.5	0.1	(5) 405.58	37.38	(5) 334.92	34.8	tl	–	2452.7	145.59	752.0	0.05	19.0	0.04	18.3	11.5	0.03	13.6	7.3
	50	13.4	0	tl	97.5	tl	92.5	0.19	67.2		0.25	64.8	0.09	11.1	0.07	8.8	3.0	0.07	8.8	3.0

Table 5
Comparison of different approaches for IEEE bus systems, PPP with no capacity limit for PMUs.

					PPP_{lt}		PPP_{lt}+VIfix		BS 4.2.1			BS 4.2.1 +VIfix			BS 4.2.1 +Ineq.4.4.1			BS 4.2.1 +Ineq.4.4.1 +stop				Alg. 4.6.1		
$\lvert V\rvert$	$\lvert S\rvert$	$\lvert S_0\rvert$	$\lvert V_0\rvert$	*opt*	*gap*	*T*	*gap*	*T*	*LB*	*T*	*it*	*LB*	*T*	*it*	*gap*	*T*	*it*	*gap*	*T*	*it*	it_{UB}	*T*	*it*	it_{UB}
9	3	0	0	2		0.02		0.02	2	0.01	13	2	0.01	13		0.01	8		0	8	2	0.01	8	2
14	1	0	3	3		0.01		0.01	3	0.05	52	3	0	2		0	8		0	7	6	0	2	1
24	4	1	0	6		0.07		0.06	6	576.95	2123	6	0.03	20		0.02	20		0.02	17	12	0.01	8	6
30	6	0	5	7		0.12		0.12	7	1048.4	3211	7	0	7		0.01	11		0.01	11	9	0	2	1
39	12	2	1	8		2.63		3.36	4	tl	3177	8	2.24	254		0.03	22		0.02	21	13	0.01	9	5
57	15	7	3	11		2.34		1.67	3	tl	2323	11	8.02	369		0.06	31		0.06	29	28	0.03	16	15
73	13	4	0	17		12.43		5.59	3	tl	2233	15	tl	4036		0.29	64		0.25	61	44	0.13	33	24
118	10	3	11	29		1.14		1.3	2	tl	2410	29	0.67	127		0.32	66		0.31	68	67	0.03	8	7
150	33	1	3	31		1318.61		1018.77	2	tl	2206	22	tl	3467		0.63	85		0.6	86	81	0.24	40	39
300	68	24	9	67	23.29	tl	17.14	tl	2	tl	1785	59	tl	2847		3.79	135		3.81	130	126	0.27	26	24
2007	308	30	15	482	-				1	tl	1137	382	tl	804	19.05	tl	1268	8.22	tl	1401	1400	288.73	466	456

improved model. As in previous experiments, instance 2007 was not run due to the size of model. The basic scheme of the algorithmic approach (BS) struggled to solve the instance with 30 buses. However, by introducing different improvements we were able to solve more instances within the time limit, including the instance with 2007 buses that was solved just in 288 s. Notice that this last instance was solved only after all the ingredients were embedded (compare Alg 4.6.1 and BS+VIfix+stop).

Summarizing the results of the two experiments, first we may conclude that the problem with capacitated PMUs is harder to solve both by the observability propagation models and the cutting plane approaches. Although the observability propagation models are useful for solving relatively small instances with small number of zero-injection nodes, they perform extremely bad for randomly generated instances with all the nodes of the system being zero-injection. The final version of the algorithm is the most effective approach for solving the problem, and outperforms significantly the observability propagation models in all the experiments carried out. The random instances are particularly easy for this approach both for capacitated and uncapacitated PMUs: when $L = 1$ the algorithm spent, in the worst case, 30 s to solve an instance with 50 buses, while in the unlimited case all the instances were solved in less than 1 second.

5.4. Results for the PPP variants

To conclude we present the computational experiments for the *L*-channel and for the variable cost problem variants. Based on the conclusions above, we will use only the IEEE instances for these tests, as they are the most difficult ones and the finely tuned version of Algorithm 4.6.1, which proved to be the most effective approach.

Aminifar et al. (2010) and Fan and Watson (2015) address the *L*-channel PMU variant of the problem for IEEE instances 14, 30, 57, 73, 118 and 300. However the size for the set of zero-injection

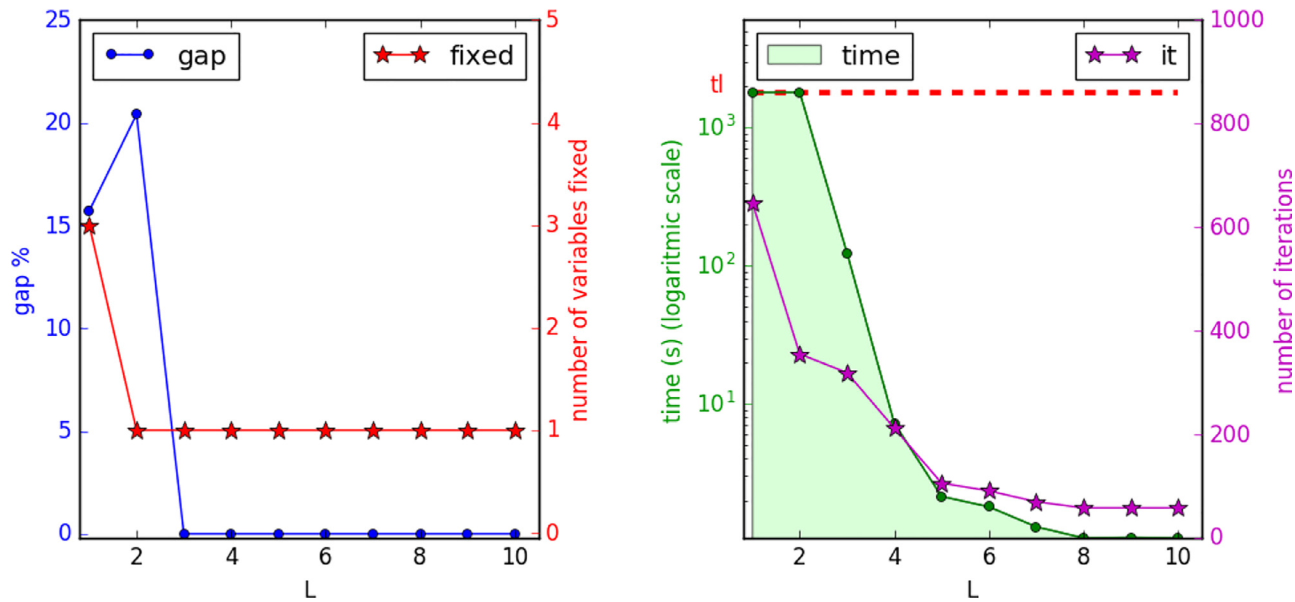

Fig. 8. Gap and number of variables fixed (left) and computational time and number of iterations (right) from the application of Algorithm 4.6.1 to solve IEEE instance with 150 buses for L-channel PPP.

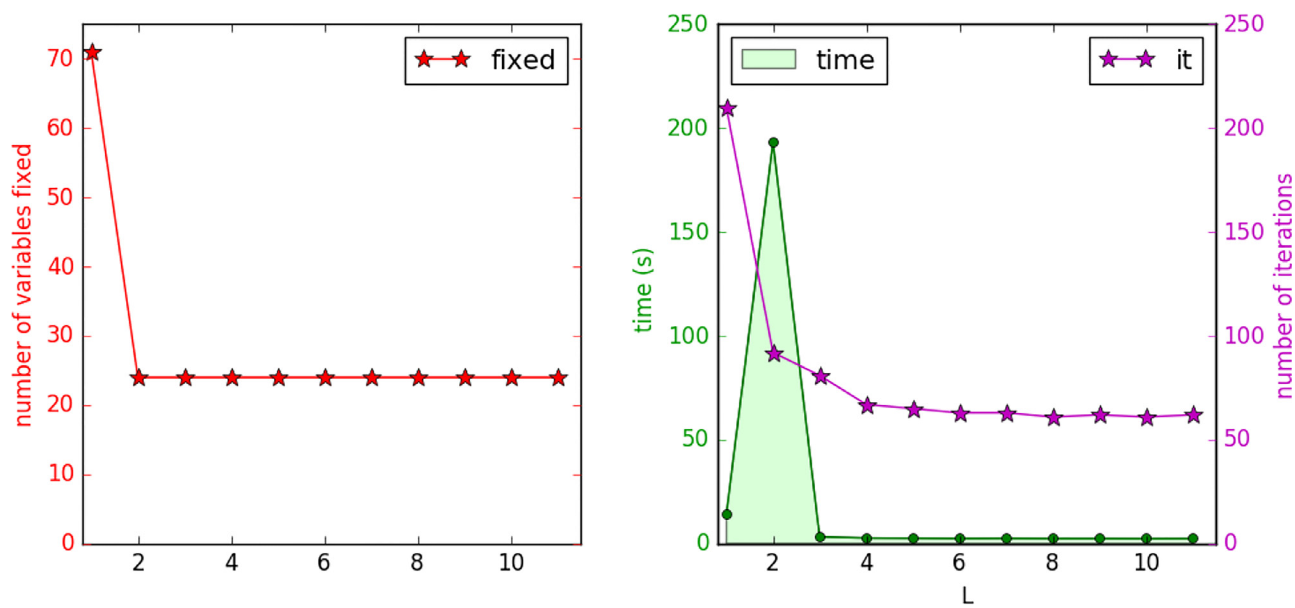

Fig. 9. Gap and number of variables fixed (left) and computational time and number of iterations (right) from the application of Algorithm 4.6.1 to solve IEEE instance with 300 buses for L-channel PPP.

nodes for IEEE instances 73 and 300 differs from ours. In the remaining instances it coincides. Due to that and the fact that in these papers the observability rules are not properly propagated, we do not perform any comparison to the other approaches.

L-channel PMU variant. Since in previous sections we have already analyzed the two extreme cases of this variant ($L = 1$ in Section 5.2, and uncapacitated in Section 5.3), the purpose of this section is to evaluate how L impacts the resolution of the problem. To recall, for $L = 1$ all the instances except 150 and 2007 were solved very fast (see Table 5), in at most 7 s (instance with 300 buses). For the uncapacitated problem all instances were solved. Moreover, except for instance 2007, the CPU time was less than 1 second. Due to this fact we will perform experiments with different values of L for the three biggest instances, with 150, 300 and 2007 buses. For each instance we consider $L = 1, \dots, n$, where $n = \max_{i \in V} |N(i)|$ corresponds to the uncapacitated case.

In Fig. 8 (left) we plot the *gap* and the number of variables fixed using the results from Section 3, and Fig. 8 (right) the CPU time and the number of iterations for the IEEE instance with 150 buses, depending on the capacity of the PMUs $L = 1, 2, \dots, 10$. The number of variables fixed does not increase as the PMUs capacity increases. It decreases from $L = 1$ to $L = 2$, because fixing the nodes in U_0 is not valid for $L > 1$ (see Theorem 4); it is maintained for $L \geq 2$, because the set S_0 is equal (already with $L = 1$, all zero-injection paths are supported). It can also be seen that the behavior of the gap is not monotonic. The algorithm finished work with the biggest value of *gap* when $L = 2$. One of the reasons for that fact can be the decrease in the number of iterations performed within the time limit, compared to $L = 1$. Once we consider PMUs with capacity 3 the problem becomes solvable within

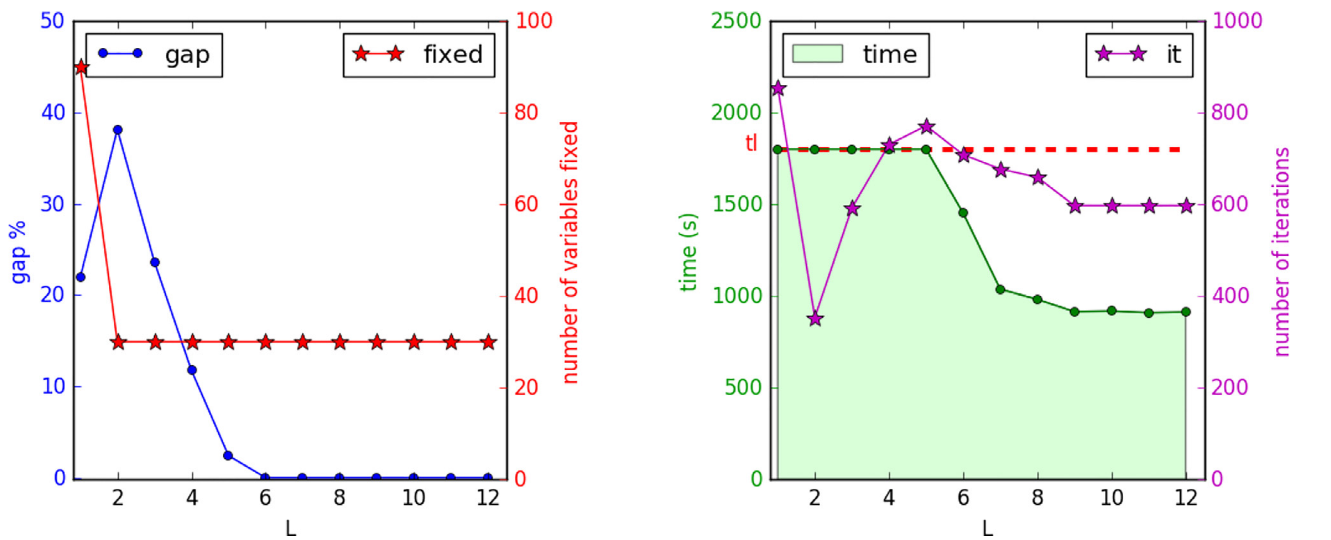

Fig. 10. umber of variables fixed (left) and computational time and number of iterations (right) from the application of Algorithm 4.6.1 to solve IEEE instance with 2007 buses for L-channel PPP.

the time limit of 1800 s (it is solved in 123.79 s). Moreover, for $L > 3$ the CPU time and the number of iterations decreased with the increase of L.

Fig. 9 plots similar values for instance 300 with $L = 1, \dots, 11$. Notice, that as this instance was solved for all the values of L, the $gap = 0$ and is not plotted in the left graph. Again, the number of variables fixed does not increase as the PMUs capacities increase, due to the above mentioned reason. Despite the fact that all the cases were solved, we still may conclude that the case with $L = 2$ is the hardest, solved in 193.30 s, but, surprisingly, there was a significant decrease in the number of iterations. For $L > 2$ there was a drastic decrease of CPU time to 2 and 3 s.

Finally, Fig. 10 shows the results for IEEE instance with 2007 buses, solved with capacities $L = 1, 2, \dots, 12$. The *gap* and *fixed* plots has absolutely similar behavior as for instance 150, having the biggest gap with $L = 2$. Again, for $L = 2$, there is a significant decrease in the number of iterations. Once $L > 2$, the gap starts decreasing and, when $L = 6$, the problem is solved withing the time limit (precisely, in 1025.80 s). Notice that 12 is the maximum degree of a node in this instance. Thus, $L = 12$ does not limit the capacity of PMUs. However, to solve the 12-channel PPP the CPU time needed was 912.43 s, while the uncapacitated case was solved in 288.73 s (see Table 5). This result is explainable: the model run in this experiment was the one presented in Section 2.3, having the binary variables y_{ij}, $(i, j) \in E$, which are redundant in this case. Contrary to the IEEE instances with 150 and 300 buses, the number of iterations did not monotonically decreased for $L > 2$.

Variable Cost. In order to make experiments with this variant of the problem we will assume that the cost of a PMU for each type depends on its capacity. We set the number of types for each instance equal to the $n = \max_{i \in V} |N(i)|$, the most expensive type of PMUs being the one that corresponds to the uncapacitated case for a given instance. We will consider two cost functions:

- *Linear cost*: for types $k = 1, \dots, n$, let capacity $L_k = k$ and cost $c_k = \frac{1}{2}k$. We use a linear cost function, due to its simplicity and the coefficient $= \frac{1}{2}$ to evaluate the effect of not having integer coefficients in the objective function.
- *Logarithmic cost*: for each $k = 1, \dots, n$, let $L_k = k$ and cost $c_k = \log_{10}(k + 1)$. Notice that, in this case, if there are types of PMUs s, k and q with capacities $L_s = L_k + L_q$, then $c_s < c_k + c_q$, which

Table 6
Algorithm 4.6.1 for the PMU variable cost problem. The types of PMUs are $k = 1, \dots, n$ with $L_k = k$, $c_k = \frac{1}{2}k$.

$\lvert V\rvert$	n	$\lvert S\rvert$	$\lvert S_0\rvert$	*opt*	*gap*	*T*	*it*	it_{UB}
9	3	3	0	0.6		0.01	9	2
14	5	1	0	1.4		0.02	5	1
24	5	4	1	2		0.13	22	10
30	7	6	0	2.6		0.27	30	10
39	5	12	2	2.8		0.35	34	25
57	6	15	7	4.2		0.65	44	44
73	5	13	4	6		5.87	69	68
118	9	10	3	11.2		2.82	61	61
150	10	33	1	11.8	15.71	tl	350	274
300	11	68	24	25.4		167.02	309	265
2007	12	308	30	133.2	35.09	tl	229	70

Table 7
Algorithm 4.6.1 for the PMU variable cost problem. The types of PMUs are $k = 1, \dots, n$ with $L_k = k$, $c_k = \log_{10}(k + 1)$.

$\lvert V\rvert$	n	$\lvert S\rvert$	$\lvert S_0\rvert$	*opt*	*gap*	*T*	*it*	it_{UB}
9	3	3	0	0.9		0.04	11	2
14	5	1	0	1.88		0.09	5	5
24	5	4	1	2.98		5.14	26	10
30	7	6	0	3.35		3.38	35	35
39	5	12	2	3.96		14.82	50	46
57	6	15	7	6.01		75.8	62	46
73	5	13	4	8.49	4.82	tl	83	70
118	9	10	3	15.1		1253.79	70	70
150	10	33	1	13.9	24.29	tl	385	373
300	11	68	24	31.06	8.7	tl	132	122
2007	12	308	30	176.67	41.15	tl	123	106

may represent the realistic behavior for the prices of the devices.

Table 6 displays the results with linear costs.

Similarly to the previous experiments for the L-channel PMU problem variant with small L (particularly, $L = 1, 2$), the IEEE instances 150 and 2007 again were not solved withing the time limit. Moreover, a significantly smaller number of iterations was performed within the time limit (*e.g.*, for $L = 1$, for instance 150, 647 iterations were executed (see Table 2), while in this case only 350 iterations were performed). The reason for that is the fact that the number of variables in the model for this variant of the problem significantly increases (see Section 2.3, variables x_i are substituted by t_{ik}). The CPU time for solving other instances is also bigger than for $L = 1$ (*e.g.*, compare the results for instance 300).

Finally, Table 7 reports the results for logarithmic costs.

For this cost function the instances are harder to solve and the number of unsolved instances increases compared to the linear costs (instances with 73 and 300 buses). Moreover, the CPU time to solve the other instances also significantly increases (*e.g.*, 2.82 vs 1253.79 for the instance with 118 buses).

6. Conclusions and future directions

In this paper we provide two general mathematical programming formulations for the PMU's Placement Problem: a single level integer programming model, that we designate by observability propagation model, and a bilevel integer programming model. In both cases we explore the benefit of using a broad set of observability rules and consider the possibility of PMUs having different capacities and purchasing costs.

To strengthen the formulations we developed new valid inequalities to the set of feasible PMU placements, and derived properties valid for an optimal PMU placement. For the bilevel formulation, we developed a cutting plane algorithm where all the theoretical results mentioned above were integrated. Additionally, we added an algorithm to improve the cutting planes used and a stopping criterion based in the computation of upper bounds during that algorithm execution. Computational results on benchmark instances from the literature show that our contributions effectively increase the efficiency of state-of-the-art algorithms. For all instances solved with a finely tuned cutting-plane algorithm, we could solve open instances in the literature and significantly reduce CPU times for the others.

As future work we intend to address variants of the PPP of practical relevance: (i) minimize the number of PMUs such that a fixed percentage of nodes is observed, (ii) maximize the number of nodes observed for a given budget, and (iii) consider scenarios where there is a mix of measuring devices, with different observability capabilities and costs, where we should select the devices to use so that the network is again observed at minimum cost. Further work will focus in providing robust PMU placements, *i.e.*, PMU placements that guarantee full observability even if a fixed number of PMUs fail.

Acknowledgments

The authors thank Claudia D'Ambrosio and Pierre Poirion for inspiring discussions on the topic and for sharing the instances of the problem.

This work has been mainly performed when the first author was in INESC TEC and wish to thank the support of the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, National Funds through the FCT (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013, and Institute for data valorisation (IVADO). The second author was supported by FCT project SFRH/BPD/101134/2014.

Appendix A. Flowcharts

In this appendix we present the flowcharts associated to the algorithms proposed in our paper. Fig. A.11 describes Algorithm 4.2.1. and Fig. A.12 describes Algorithm 4.3.1, that computes the nodes that are observed for a given PMU location. The flowchart that describes the computation of a maximal PMU placement and of an upper bound, introduced in Section 4.5, Algorithm 4.4.1, is mimicked by flowchart A.13 (for the sake of simplicity we present the version of the algorithm for the uncapacitated PPP only). Finally, flowchart A.14, describes the improved cutting plane method in Algorithm 4.6.1

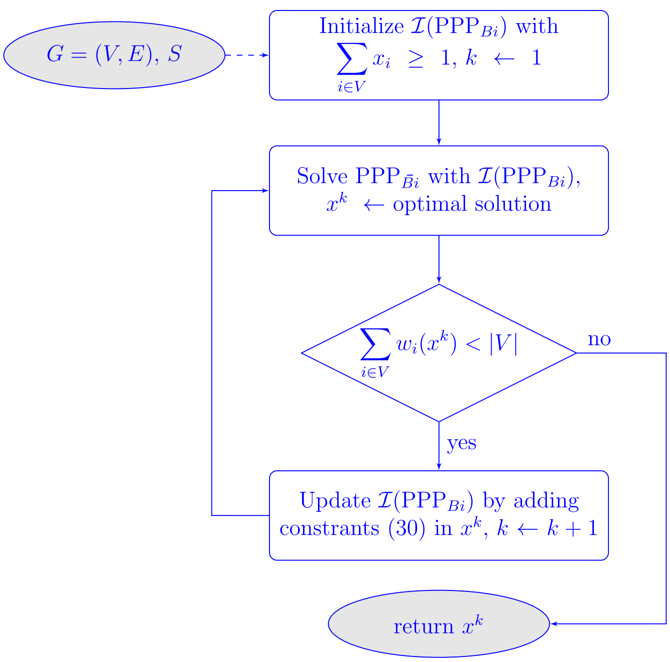

Fig. A11. Basic scheme. Algorithm 4.2.1.

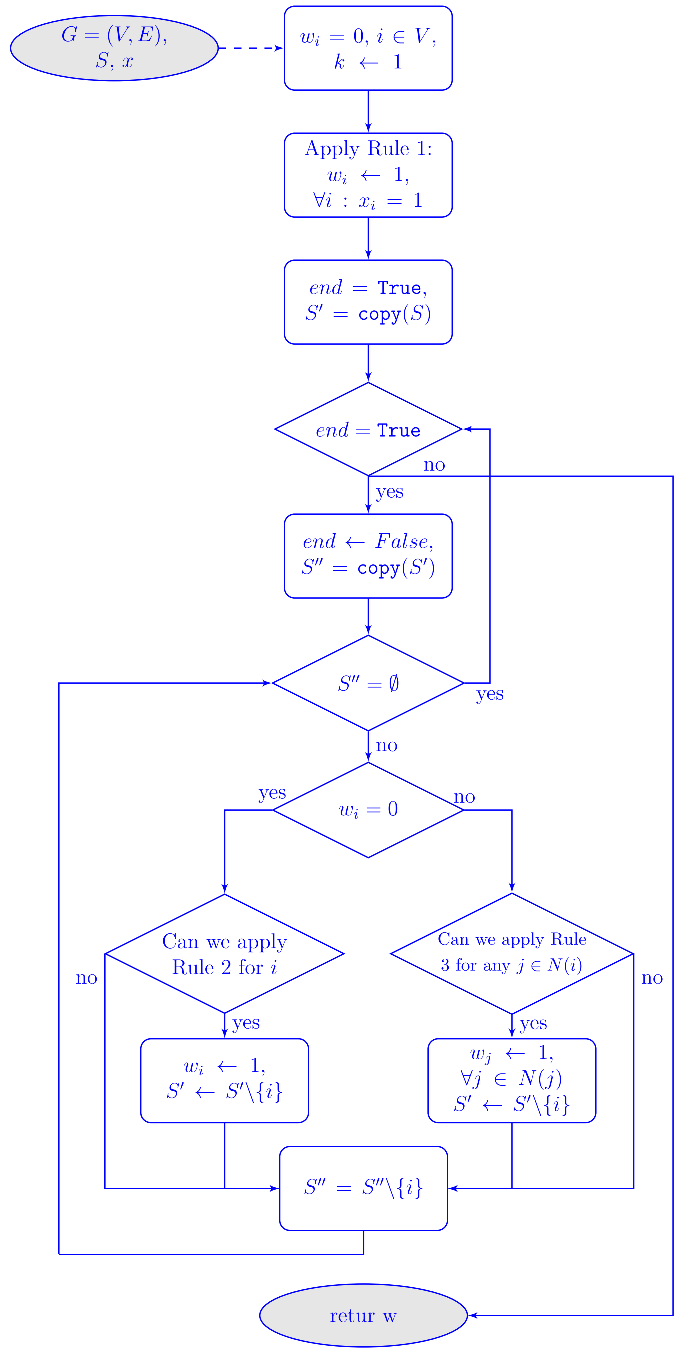

Fig. A12. Computation of the nodes observed. Algorithm 4.3.1.

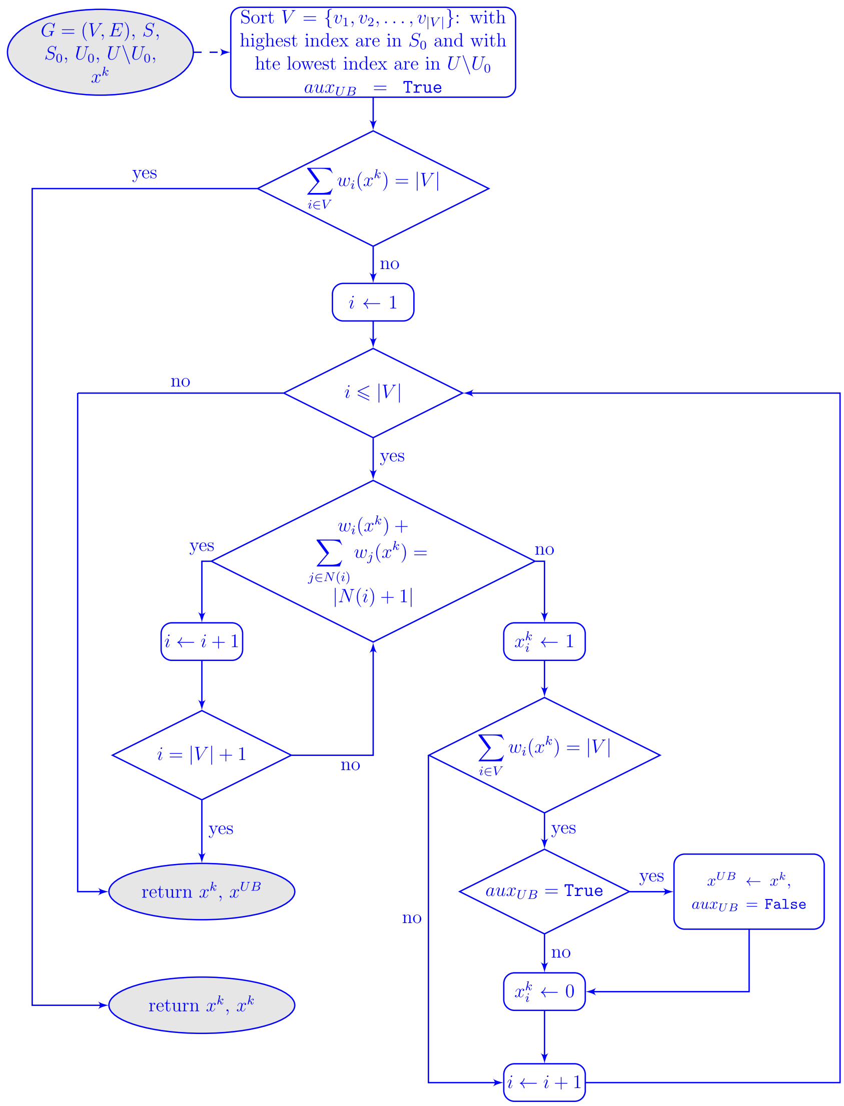

Fig. A13. Computation of a maximal PMU placement and of an upper bound. Algorithm 4.4.1.

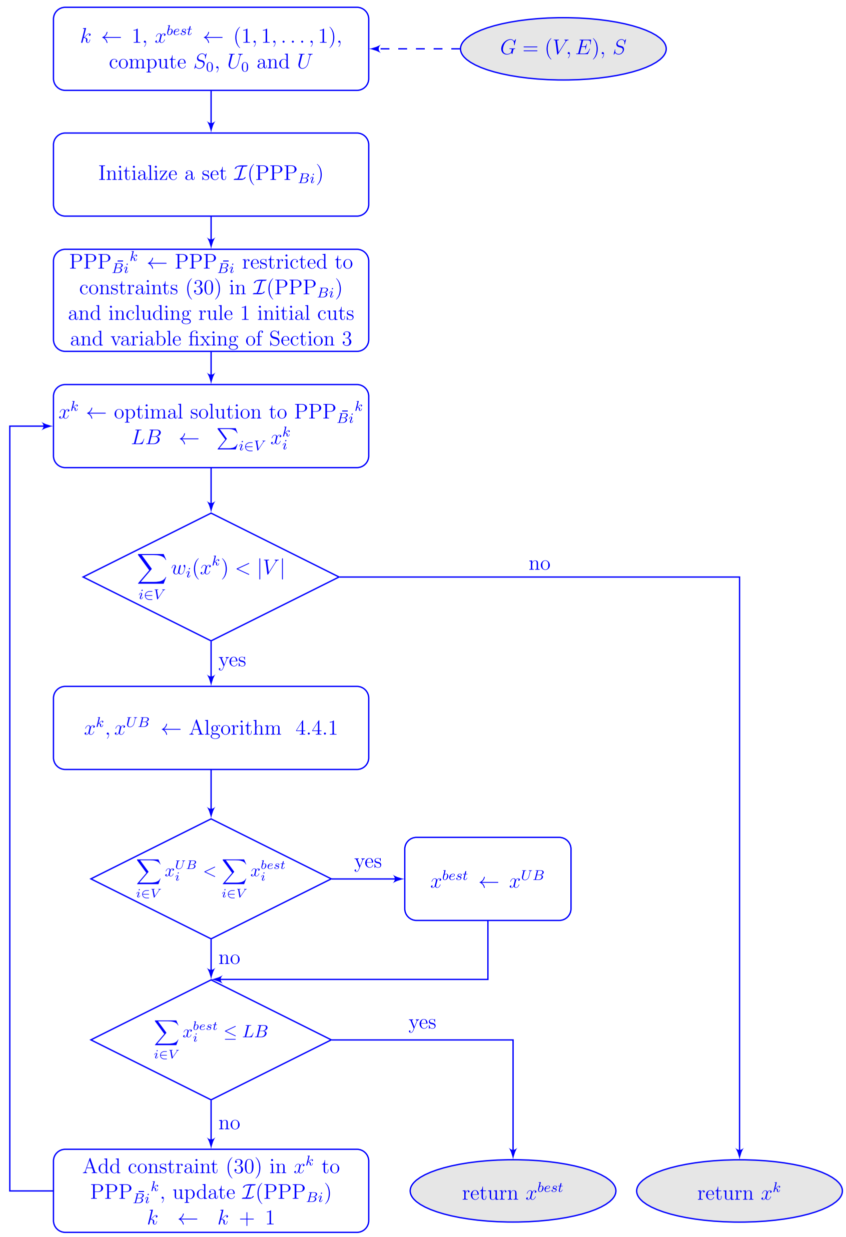

Fig. A14. Improved cutting plane. Algorithm 4.6.1.

References

Ahmadi, A., Alinejad-Beromi, Y., Moradi, M., 2011. Optimal PMU placement for power system observability using binary particle swarm optimization and considering measurement redundancy. Expert Syst. Appl. 38 (6), 7263–7269. doi:10.1016/j.eswa.2010.12.025.

Aminifar, F., Khodaei, A., Fotuhi-Firuzabad, M., Shahidehpour, M., 2010. Contingency-constrained PMU placement in power networks. IEEE Trans. Power Syst. 25 (1), 516–523. doi:10.1109/TPWRS.2009.2036470.

Aminifar, F., Lucas, C., Khodaei, A., Fotuhi-Firuzabad, M., 2009. Optimal placement of phasor measurement units using immunity genetic algorithm. IEEE Trans. Power Deliv. 24 (3).

Brueni, D.J., Heath, L.S., 2005. The PMU placement problem. SIAM J. Discret. Math. 19 (3), 744–761. doi:10.1137/S0895480103432556.

Caprara, A., Carvalho, M., Lodi, A., Woeginger, G.J., 2014. A study on the computational complexity of the bi-level knapsack problem. SIAM J. Optim. 24 (2), 823–838. doi:10.1137/130906593.

Colson, B., Marcotte, P., Savard, G., 2005. Bilevel programming: a survey. 4OR 3 (2), 87–107. doi:10.1007/s10288-005-0071-0.

Dempe, S., 2002. Foundations of Bilevel Programming, 61. Springer U.S.

Dua, D., Dambhare, S., Gajbhiye, R.K., Soman, S.A., 2008. Optimal multistage scheduling of PMU placement: an ILP approach. IEEE Trans. Power Deliv. 23 (4), 1812–1820. doi:10.1109/TPWRD.2008.919046.

Fan, N., Watson, J.-P., 2015. On integer programming models for the multi-channel PMU placement problem and their solution. Energy Syst. 6 (1), 1–19. doi:10.1007/s12667-014-0132-6.

Foundation, P. S., 2012. Python v2.7.3 documentation. http://docs.python.org/library/random.html.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.

Gou, B., 2008. Generalized integer linear programming formulation for optimal PMU placement. IEEE Trans. Power Syst. 23 (3), 1099–1104. doi:10.1109/TPWRS.2008.926475.

Gyllstrom, D., Rosensweig, E., Kurose, J., 2012. On the impact of PMU placement on observability and cross-validation. In: Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet. ACM, New York, NY, USA, pp. 20:1–20:10. doi:10.1145/2208828.2208848.

Hajian, M., Ranjbar, A.M., Amraee, T., Shirani, A.R., 2007. Optimal placement of phasor measurement units: particle swarm optimization approach. In: 2007 International Conference on Intelligent Systems Applications to Power Systems, pp. 1–6. doi:10.1109/ISAP.2007.4441610.

Haynes, T.W., Hedetniemi, S.M., Hedetniemi, S.T., Henning, M.A., 2002. Domination in graphs applied to electric power networks. SIAM J. Discret. Math. 15 (4), 519–529. doi:10.1137/S0895480100375831.

Illinois Center for a Smarter Electric Grid. http://icseg.iti.illinois.edu/power-cases/. Accessed: 2017-30-01.

Koutsoukis, N.C., Manousakis, N.M., Georgilakis, P.S., Korres, G.N., 2013. Numerical observability method for optimal phasor measurement units placement using recursive tabu search method. IET Gener, Transm. Distrib. 7 (4), 347–356. doi:10.1049/iet-gtd.2012.0377.

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

Manousakis, N.M., Korres, G.N., Georgilakis, P.S., 2012. Taxonomy of PMU placement methodologies. IEEE Trans. Power Syst. 27 (2), 1070–1077. doi:10.1109/TPWRS.2011.2179816.

Marín, F.J., García-Lagos, F., Joya, G., Sandoval, F., 2003. Optimal Phasor Measurement Unit Placement Using Genetic Algorithms. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 486–493. doi:10.1007/3-540-44868-3-62.

Mesgarnejad, H., Shahrtash, S.M., 2008. Multi-objective measurement placement with new parallel tabu search method. In: 2008 IEEE Canada Electric Power Conference, pp. 1–6. doi:10.1109/EPC.2008.4763294.

Mohammadi-Ivatloo, B., 2009. Optimal placement of PMUs for power system observability using topology based formulated algorithms. J. Appl. Sci. 9 (13), 2463–2468. doi:10.3923/jas.2009.2463.2468.

Nazari-Heris, M., Mohammadi-Ivatloo, B., 2015. Application of heuristic algorithms to optimal PMU placement in electric power systems: an updated review. Renew. Sustain. Energy Rev. 50, 214–228. https://doi.org/10.1016/j.rser.2015.04.152.

Nuqui, R.F., Phadke, A.G., 2005. Phasor measurement unit placement techniques for complete and incomplete observability. IEEE Trans. Power Deliv. 20 (4), 2381–2388. doi:10.1109/TPWRD.2005.855457.

Phadke, A.G., 1993. Synchronized phasor measurements in power systems. IEEE Comput. Appl. Power 6 (2), 10–15. doi:10.1109/67.207465.

Poirion, P.-L., Toubaline, S., D'Ambrosio, C., Liberti, L., 2016. The power edge set problem. Networks 68 (2), 104–120. doi:10.1002/net.21684.

Ralphs, T.K., 2017. MibS: mixed integer bilevel solver. https://github.com/tkralphs/MibS.

Sodhi, R., Srivastava, S., Singh, S., 2010. Optimal PMU placement method for complete topological and numerical observability of power system. Electr. Power Syst. Res. 80 (9), 1154–1159. doi:10.1016/j.epsr.2010.03.005.

Xu, B., Abur, A., 2004. Observability analysis and measurement placement for systems with PMUs. In: Proceedings of 2004 IEEE PES Conference and Exposition, vol.2, pp. 943–946.

Xu, J., Wen, M.H.F., Li, V.O.K., Leung, K.C., 2013. Optimal PMU placement for wide-area monitoring using chemical reaction optimization. In: 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), pp. 1–6. doi:10.1109/ISGT.2013.6497845.