Max Observability PMU Placement with Cross-Validation

Technical Report UM-CS-2011-14
Daniel Gyllstrom, Elisha Rosensweig, and Jim Kurose
Department of Computer Science. University of Massachusetts Amherst USA
{dpg, elisha, kurose}@cs.umass.edu

***Abstract*—Significant investments have been made to deploy phasor measurement units (PMUs) on electric power grids worldwide. PMUs allow the state of the power system – the voltage phasor of system buses and current phasors of all incident transmission lines – to be directly measured. In some cases, it is also possible to infer the voltage and current phasors at neighboring buses and lines. Because PMUs are expensive, it is typically not possible to deploy enough PMUs to observe all phasors in a grid network [3], [6].**

In this paper, we formulate three PMU placement problems that place PMUs at a subset of system buses to achieve different goals: MAXOBSERVE, SAFEPLACE, and MAXSAFEPLACE. MAXOBSERVE aims to observe the maximum number of buses with a given number of PMUs. SAFEPLACE and MAXSAFEPLACE consider PMU placements that meet the requirement that all PMUs are placed near each other so their measurements can be cross-validated. SAFEPLACE considers cases when the entire network can be observed, and MAXSAFEPLACE tries to maximize the number of observed buses under this new constraint.

We prove that all three problems are NP-Complete. We then consider the performance of a simple greedy algorithm that places PMUs incrementally, with the next PMU placed at a bus where it observes the maximum of number of buses. Through simulations, we compare the performance of this greedy algorithm with the optimal placement of PMUs over several IEEE bus systems as well as synthetic graphs. For all three placement problems, the greedy algorithm yields, on average, a PMU placement that is within 96% of optimal.

I. INTRODUCTION

Significant investments have been made to deploy phasor measurement unit (PMUs) on electric power grids worldwide. A new generation of PMUs provide synchronized voltage and current measurements at a sampling rate orders of magnitude higher than the status quo: 10 to 60 samples per second rather than one sample every 1 to 4 seconds. Consequently, PMUs have the potential to enable an entirely new set of applications for the power grid: protection and control during abnormal conditions, real-time distributed control, postmortem analysis of system faults using time synchronized data, advanced state estimators for system monitoring, and the reliable integration of renewable energy resources [1].

An electric power system consists of a set of buses – an electric substation, power generation center, or aggregation of loads – and transmission lines connecting those buses. The state of a power system is defined by the voltage phasor – the magnitude and phase angle – of all system buses and the current phasor of all transmission lines. PMUs placed on buses provide real-time measurements of these system variables. However, because PMUs are expensive, they cannot be deployed on all system buses [3], [6]. Fortunately, all system variables can be observed even when PMUs are placed at only a subset of system buses, by estimating the voltage phasor of buses without PMUs and the current phasors of unmeasured transmission lines using Ohm's and Kirchhoff's laws [3], [4].

In this work, we study two sets of PMU placement problems. The first problem – MAXOBSERVE – aims to observe the maximum number of system buses given a fixed number of PMUs. A bus is said to be *observed* if there is a PMU placed at it or if its voltage phasor can be estimated using Ohm's or Kirchhoff's Law. MAXOBSERVE is closely related to another well-studied PMU placement problem: finding the minimum number of PMUs that can result in the observation of all system buses [3], [4], [8], [10], [12]. However, the MAXOBSERVE problem formulation presented here is unique.

The second set of placement problems aim to identify PMU errors, which have been recorded in PMUs used in practice [11]. Informally, the goal of the first such placement problem (MAXSAFEPLACE) is to observe, given a set of PMUs, the maximum number of buses while deploying PMUs "near" each other to enable them to cross-validate each other's measurements. Next, we formulate a placement problem similar to MAXSAFEPLACE, called SAFEPLACE. In SAFEPLACE, the goal is to find the minimum number of PMUs such that all system buses are observed under the constraint that each PMU is placed near enough to at least one other PMU to allow its PMU measurements to be cross-validated.

We make the following contributions in this paper:

- We formally define graph-theoretic rules for cross-validating PMU placements. PMUs are cross-validated when PMU data from two buses can be used to each, independently, compute the voltage phasor of a non-PMU bus (Section III-C).
- We define three related PMU placement problems. The first problem seeks to observe the maximum number of nodes given a fixed number of PMUs. The second two problems consider placing the minimum number of PMUs such that all PMUs are cross-validated and the maximum number of nodes is observed (Section II).
- We prove that all three PMU placement problems are NP-Complete. This represents our most important contri-

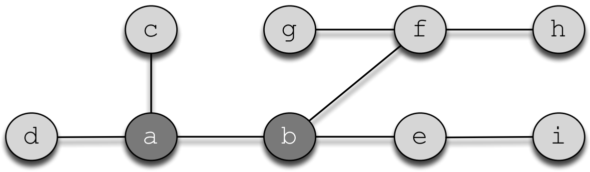


Fig. 1. Example power system graph. The dark shaded nodes – a and b – have PMUs.

bution (Section II).

- We present greedy approximation algorithms for each PMU placement problem (Section IV) and through simulations, we evaluate these algorithms over real IEEE bus systems and synthetic graphs (Section V).

II. Preliminaries

In this section we introduce notation and underlying assumptions (Section II-A), and define our observability (Section II-B) and cross-validation (Section III-C) rules.

A. System Assumptions, Notation and Terminology

In this work, we make the following assumptions about PMU placements and buses:

1) A PMU can only be placed on a bus.
2) A PMU on a bus measures the voltage phasor at the bus and the current phasor of all transmission lines connected to it.
3) We assume all buses are zero-injection. A bus is zero-injection if it has no load nor generator [14]. Kirchhoff's Current Law can only be applied to zero-injection buses.

We model a power grid as an undirected graph $G = (V, E)$. Each $v \in V$ represents a bus. A bus is either an electrical substation, a power generation center, or an aggregation of loads. Each $(u, v) \in E$ is a transmission line connecting buses u and v. Figure 1 is an example of a power system modeled as an undirected graph.

Using the same notation as Brueni and Heath [4], we define two Γ functions. For $v \in V$ let $\Gamma(v)$ be the set of v's neighbors in G, and $\Gamma[v] = \Gamma(v) \cup \{v\}$. A PMU placement $\Phi_G \subseteq V$ is a set of nodes at which PMUs are placed, and $\Phi_G^R \subseteq V$ is the set of observed nodes for graph G with placement Φ_G (see definition of observability below). $k^* = \min\{|\Phi_G| : \Phi_G^R = V\}$ is used to denote the minimum number of PMUs needed to observe the entire network.

For convenience, we refer to any node with a PMU as a *PMU node*. Additionally, we shall say that a set $W \subseteq V$ is observed if all nodes in the set are observed, and if $W = V$ we refer to the graph as *fully observed*.

B. Observability Rules

We use the simplified observability rules elegantly stated by Brueni and Heath [4]. For completeness, we restate the rules here:

1) **Observability Rule 1 (O1)**. *If node v is a PMU node, then $\Gamma[v]$ is observed. Formally, if $v \in \Phi$, then $\Gamma[v] \subseteq \Phi_G^R$.*
2) **Observability Rule 2 (O2)**. *If a node v is observed and $\Gamma(v)\backslash\{u\}$ is observed for some $u \in \Gamma(v)$, then $\Gamma[v]$ is observed. Formally, if $v \in \Phi_G^R$ and $|\Gamma(v) \cap (V - \Phi_G^R)| \leq 1$, then $\Gamma[v] \subseteq \Phi_G^R$.*

Consider the example in Figure 1, where the shaded nodes are PMU nodes. Using O1, the PMU at a results in the observation of a, c, and d. Likewise, the PMU at b make b, f, and e observed. Finally, O2 can be applied at e because e is observed and all of e's neighbors except i are observed. As a result, i becomes observed. Note that O2 cannot be applied at f because f has two unobserved neighbors.

C. Cross-Validation Rules

From Vanfretti et al. [11], PMU measurements can be cross-validated when: (1) a voltage phasor of a non-PMU bus can be computed by PMU data from two different buses or (2) the current phasor of a transmission line can be computed from PMU data from two different buses. [1] Although PMU data is actually being cross-validated, for convenience, we say a PMU is cross-validated. Below is a precise statement of the cross-validation rules taken from Vanfretti et al. [11]. A PMU is *cross-validated* if one of the rules below is satisfied:

1) **Cross-Validation Rule 1 (XV1)**. *Adjacent PMU nodes cross-validate each other. Formally, $u, v \in \Phi$, $u \in \Gamma(v)$, and $v \in \Gamma(u)$.*
2) **Cross-Validation Rule 2 (XV2)**. *PMU nodes with a common neighbor with no PMU cross-validate each other. Formally, $u, v \in \Phi$, $v \notin \Gamma(u)$, $u \notin \Gamma(v)$ and $\exists w$ such that $v \neq w$, $u \neq w$, $w \in \Gamma(v)$, $w \in \Gamma(u)$, and $w \notin \Phi$.*

In short, the cross-validation rules require that *the PMU is within two hops of another PMU*. For example, in Figure 1, the PMUs at a and b cross-validate each other by XV1.

III. Problem Formulations and NP-Completeness Proofs

In this section we define three PMU placement problems: MaxObserve (Section III-B), SafePlace (Section III-C), and MaxSafePlace (Section III-D). For each problem, we first define the optimization version of the problem and then its corresponding decision problem. Finally, we prove that each decision problem is NP-Complete. Note that in all three placement problems we are only concerned with computing the voltage phasors of each bus. Using the values of the voltage phasors, Ohm's Law can be easily applied to compute the current phasors of each transmission line.

A. Proof Strategy

In the coming sections we use slight variations of the approach presented by Brueni and Heath in [4] to prove the NP-completeness of the problems we consider. We found their scheme to be extensible for proving many properties of PMU placements. For purposes of clarity, we begin by explaining

[1] Vanfretti et al. [11] use the term "redundancy" instead of cross-validation.

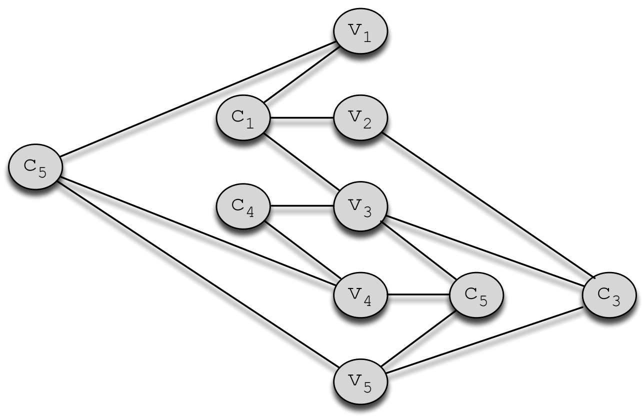


Fig. 2. $G(\varphi) = (V(\varphi), E(\varphi))$ formed from φ in Equation (1)

this approach in general terms, and then consider the approach in detail for each problem.

Our NP-Completeness proofs all reduce from planar 3-SAT (P3SAT). A P3SAT formula, ϕ, is a boolean formula in conjunctive normal form (CNF) such that each clause contains at most 3 literals and the undirected graph $G(\phi)$ is planar [9]. $G(\phi) = (V(\phi), E(\phi))$ is a bipartite graph constructed from a 3-SAT formula ϕ with variables $\{v_1, v_2, \ldots, v_r\}$ and the set of clauses $\{c_1, c_2, \ldots, c_s\}$ as follows: $V(\phi) = \{v_i \mid 1 \leq i \leq r\} \cup \{c_j \mid 1 \leq j \leq s\}$ and $E(\phi) = \{(v_i, c_j) \mid v_i \in c_j$ or $\overline{v_i} \in c_j\}$. For example, P3SAT formula

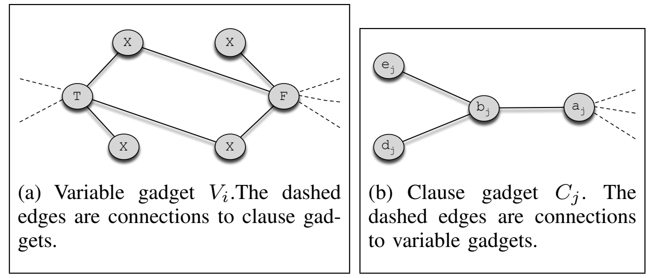
$$\begin{aligned} \varphi \;=\; & (\overline{v_1} \vee v_2 \vee v_3) \wedge (\overline{v_1} \vee \overline{v_4} \vee v_5) \wedge (\overline{v_2} \vee \overline{v_3} \vee \overline{v_5}) \\ & \wedge (v_3 \vee \overline{v_4}) \wedge (\overline{v_3} \vee v_4 \vee \overline{v_5}) \end{aligned} \quad (1)$$

has graph $G(\varphi) = (V(\varphi), E(\varphi))$ shown in Figure 2. Note that φ is a specific P3SAT formula used for this example and is thus different from the generic P3SAT formula, ϕ, used in our reductions.

Following the approach in [4], for P3SAT formula, ϕ, we replace each variable node and each clause node in $G(\phi)$ with a specially constructed set of nodes, termed a *gadget*. Specifically, each variable gadget has a subgraph of nodes that represent a "True" assignment to that variable, and a subgraph of nodes that represent a "False" assignment to the variable. All variable gadgets have the same structure, and all clause gadgets have the same structure (that is different from the variable gadget structure). We then prove that a PMU placement for this graph results in a fully (maximally) observed graph if and only if that PMU placement can be interpreted as assigning unambiguous truth values to each variable, in a manner which satisfies the formula ϕ. For convenience, when we refer to reference [4], we are alluding to Theorem 9 in [4].

B. MAXOBSERVE *Problem Statement*

MAXOBSERVE is a variation of the PMUP problem described by Brueni and Heath [4] and the PDS problem defined by Haynes et al. [8]: rather than consider the minimum number of PMUs required for full system observability, MAXOBSERVE finds the maximum number of nodes that can be observed using a fixed number of PMUs.



(a) Variable gadget V_i.The dashed edges are connections to clause gadgets.

(b) Clause gadget C_j. The dashed edges are connections to variable gadgets.

Fig. 3. Gadgets used in Theorem III.1 proof.

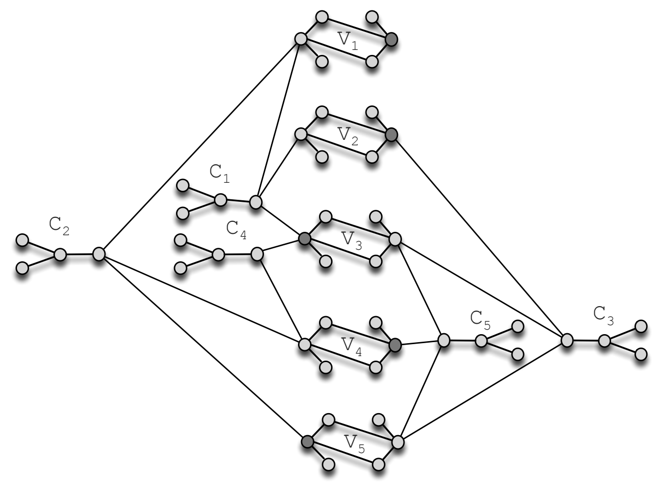


Fig. 4. Graph $G = (V, E) = H_1(\varphi)$ formed from φ formula in Theorem III.1 proof.

MAXOBSERVE Optimization Problem:

- Input: Graph $G = (V, E)$, k PMUs such that $1 \leq k < k^*$.
- Output: A placement of k PMUs, Φ, such that $|\Phi_G^R|$ is maximum.

MAXOBSERVE Decision Problem:

- Instance: Graph $G = (V, E)$, k PMUs such that $1 \leq k < k^*$.
- Question: For a given $m < |V|$, is there a Φ_G such that $|\Phi_G| \leq k$ and $m \leq |\Phi_G^R| < |V|$?

Before proving that MAXOBSERVE is NP-Complete, we provide some background on NP-Completeness. NP-Complete problems are the hardest problems in complexity class $\mathcal{NP}$. Proving a decision problem, Π, is NP-Complete is a three step procedure. First, we need to show Π is in $\mathcal{NP}$. Second, we select a known NP-Complete problem Π' and construct a polynomial-time transformation, f, that maps any instance of Π' to an instance of Π. Finally, we need to prove that f is a transformation: f maps any "yes" instance of Π' to a "yes" instance of Π and any "no" instance of Π' to a "no" instance of Π [7].

Theorem III.1. MAXOBSERVE *is NP-Complete.*

Proof idea: First, we construct problem-specific gadgets for variables and clauses. We then demonstrate that any solution that observes m nodes must place the PMUs only on nodes in the variable gadgets. Next we show that as a result of this, the problem of observing m nodes in this graph reduces to the NP-complete problem presented in [4], which concludes our proof.

Proof: We start by arguing that MAXOBSERVE $\in \mathcal{NP}$. First, nondeterministically select k nodes in which to place PMUs. Then we use the rules specified in Section II-B to determine the number of observed nodes.

We reduce from P3SAT, where ϕ is an arbitrary P3SAT formula, to show SAFEPLACE is NP-hard. Note that ϕ is different from the example φ formula used to introduce the P3SAT problem. Specifically, given a graph $G(\phi)$ we construct a new graph $H_1(\phi) = (V_1(\phi), E_1(\phi))$ by replacing each variable (clause) node in $G(\phi)$ (Figure 2) with the variable (clause) gadget shown in Figure 3(a) (3(b)). The edges connecting clause gadgets with variable gadgets express which variables are in each clause: for each clause gadget C_j, node a_j is attached to node T in variable gadget V_i if, in ϕ, v_i is in c_j, and to node F if $\overline{v_i}$ is in c_j. The resulting graph for the example given in Figure 2 is shown in Figure 4; the corresponding formula for this graph, φ, is satisfied by truth assignment A_φ: $\overline{v_1}, \overline{v_2}, v_3, \overline{v_4}$, and $\overline{v_5}$ are True. This corresponds to the dark shaded nodes in Figure 4. We also note that $H_1(\phi)$ is identical to the graph $H(\phi)$ constructed in [4], except that there C_j consisted only of nodes $\{a_j, b_j\}$, and thus $|H_1(\phi)| = |H(\phi)| + 2s$. We return to this similarity later in the proof. For convenience, we let $G = H_1(\phi)$.

With this construct in place, we move on to our proof. Here we consider the case of $k = r$ and $m = 6r+2s$, and show that ϕ is satisfiable if and only if $r = |\Phi_G|$ PMUs can be placed on G such that $m \leq |\Phi_G^R| < |V|$. We will later discuss how to extend this proof for any larger value of m.

($\Rightarrow$) Assume ϕ is satisfiable by truth assignment A_ϕ. Then, consider the placement Φ_G s.t. for each variable gadget V_i, $T_i \in \Phi_G \Leftrightarrow v_i = True$ in A_ϕ, and $F_i \in \Phi_G \Leftrightarrow v_i = False$. It has been shown in [4] that for $H(\phi)$ this placement observes all $H(\phi)$, and it can be easily verified that all nodes in $H_1(\phi)$ are observed as well except for d_j, e_j for each C_j. This amounts to $2s$ nodes, so exactly m nodes are observed by Φ_G, as required.

($\Leftarrow$) We begin by proving that any solution that observes m nodes must place the PMUs only on nodes in the variable gadgets. Assume that there are $1 < t \leq r$ variable gadgets without a PMU. Then, at most t PMUs are on nodes in clause gadgets, so *at least* $\max(s-t, 0)$ clause gadgets are without PMUs. We want to show here that for $m = 6r + 2s$, $t = 0$.

To prove this, we rely on the following two simple observations:

- In any variable gadget V_i, nodes X (Figure 3(a)) cannot be observed unless there is a PMU somewhere in V_i. Note that there are 4 such nodes per V_i.
- In any clause gadget C_j, nodes e_j and d_j cannot be observed unless there is a PMU somewhere in C_j. Note that there are 2 such nodes per C_j.

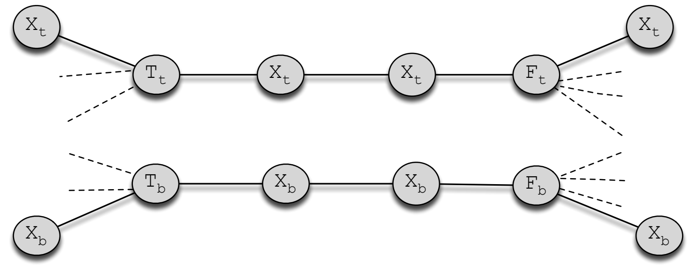


Fig. 5. Variable gadget used in Theorem III.2 proof. The dashed edges are connections to clause gadgets.

Thus, given some t, the number of unobserved nodes is *at least* $4t + \max(2(s-t), 0)$. However, since $|V| - m \leq 2s$, there are *at most* $2s$ unobserved nodes. So we get $2s \geq 4t + \max(2(s-t), 0)$. We consider two cases:

- $s \geq t$: then we get $2s \geq 2s + 2t \Rightarrow t = 0$.
- $s < t$: then we get $2s \geq 4t \Rightarrow s \geq 2t$, and since we assume here $0 \leq s < t$ this leads to a contradiction and so this case cannot occur.

Thus, we have concluded that the r PMUs must be on nodes in variable gadgets, all of which, it is important to note, were also part of the original $H(\phi)$ graph. We return to this point shortly.

We now observe that for each clause gadget C_j, such a placement of PMUs cannot observe nodes of type e_j, d_j, which amounts to a total of $2s$ unobserved nodes - the allowable bound. This means that all other nodes in G must be observed. Specifically, this is exactly all the nodes in the original $H(\phi)$ graph, and PMUs can only be placed on variable gadgets, all of which are included in $H(\phi)$ as well. Thus, the problem reduces to the problem in [4]. We use the proof in [4] to determine that all clauses in ϕ are satisfied by the truth assignment derived from Φ_G. ∎

C. SAFEPLACE *Problem Statement*

SAFEPLACE Optimization Problem:

- Input: Graph $G = (V, E)$.
- Output: A placement of PMUs, Φ_G, such that $\Phi_G^R = V$, each $v \in \Phi_G$ is cross-validated according to the rules specified in Section III-C, and Φ_G is minimal.

SAFEPLACE Decision Problem:

- Instance: Graph $G = (V, E)$, k PMUs such that $k \geq 1$.
- Question: Is there a Φ_G such that $|\Phi_G| \leq k$, $\Phi_G^R = V$, and each $v \in \Phi_G$ is cross-validated?

Theorem III.2. SAFEPLACE *is NP-Complete.*

Proof: First, we argue that SAFEPLACE $\in \mathcal{NP}$. Given a SAFEPLACE solution, we use the polynomial time algorithm described in our proof for Theorem III.1 to determine if all nodes are observed. Then, for each PMU node we run a breadth-first search, stopping at depth 2, to check that the cross-validation rules are satisfied.

To show SAFEPLACE is NP-hard, we reduce from P3SAT. Our reduction is similar to the one used in Theorem III.1. For this problem, we use different variable and clause gadgets. The clause gadgets consist of the edge (a_j, b_j) from Figure 3(b), which are the same as used in [4]. The new variable gadget is shown in Figure 5. As can be seen in this figure, the variable gadgets are comprised of two disconnected subgraphs: we refer to the upper subgraph as V_{it} and the lower subgraph as V_{ib}. Clause gadgets are connected to a variable gadgets in the following manner: for each clause c_j that contains variable v_i in ϕ, the corresponding clause gadget has the edges $(a_j, T_t), (a_j, T_b)$, and for each clause c_j that contains variable $\overline{v_i}$ in ϕ, the corresponding clause gadget has the edges $(a_j, F_t), (a_j, F_b)$. We denote the resulting graph as $H_2(\phi)$, and for what follows assume $G = H_2(\phi)$.

We now show that ϕ is satisfiable if and only if $k = 2r$ PMUs can be placed on G such that G is fully observed under the condition that all PMUs are cross-validated, and that $2r$ PMUs are the minimal bound for observing the graph with cross-validation.

($\Rightarrow$) Assume ϕ is satisfiable by truth assignment A_ϕ. For each $1 \leq i \leq r$, if $v_i = True$ in A_ϕ we place a PMU at T_b and at T_t of the variable gadget V_i. Otherwise, we place a PMU at F_b and at F_t of this gadget. In either case, the PMU nodes in V_i must be adjacent to a clause node, making T_t (F_t) two hops away from T_b (F_b). Therefore, all PMUs are cross-validated by XV2.

Now we argue that Φ_G observes all $v \in V$:

- Consider a clause node a_j. Since ϕ is satisfied, for some index i we have $v_i \in c_j \wedge v_i \in A_\phi$ or $\overline{v_i} \in c_j \wedge \overline{v_i} \in A_\phi$. For the first case, the PMUs in V_i are placed on $\{T_b, T_t\}$ and as a result a_j is observed by applying O1 at T_b or at T_t. A similar argument applies for the second case. So, all a_j nodes are observed.
- Next, consider the nodes on the variable gadgets. When $v_i \in A_\phi$, T_t's neighbors, in V_{it}, are observed via O1. (the second case, $\overline{v_i} \in A_\phi$, follows by symmetry). The remaining V_{it} nodes are observed via O2 - note that if F_t is connected to a clause gadget we know from the previous step this clause is observed. By symmetry of V_{ib} and V_{it}, the same argument can be made for V_{ib} to show all V_{ib} nodes are observed.
- Finally, all the neighbors of a_j in variable gadgets are observed, and a_j is observed, so we can now apply O2 at each node a_j to observe the remaining b_j nodes.

This completes this direction of the theorem.

($\Leftarrow$) Suppose Φ_G observes all nodes in G under the condition that each PMU is cross-validated, and that $|\Phi_G| = 2r$. We want to show that ϕ is satisfiable by the truth assignment derived from Φ_G. We prove this by showing that (a) each variable gadget must have exactly 2 PMUs and (b) there must be a PMU at each subgraph of the variable gadget. Once (b) is shown, (c) cross-validation restrictions force the PMUs to be either on both T-nodes or both F-nodes. We conclude by showing that (d) the PMU nodes correspond to true/false assignments to variables which satisfy ϕ.

We begin by showing that each variable gadget must have 2 PMUs. Let V_i be a variable gadget with less than two PMUs. By placing PMUs on clause gadgets attached to V_i, at most we can observe T_t, T_b, F_t and F_b directly from the clause gadgets. Next, at least one of the V_i subgraphs has no PMU: without loss of generality, let this be V_{it}. We cannot apply O1 at T_t or F_t, since they have no PMU. We cannot apply O2 at these nodes since they each have two unobserved X_t nodes. Thus, all X_t nodes are unobserved in V_{it}, contrary to our assumption that the entire graph is observed. Thus we have shown that there must be at least 2 PMUs at each variable gadget. Also it is clear from this proof that, in fact, there must be at least one PMU in each subgraph of each variable gadget. Finally, since there are $2r$ PMUs and r variables, we conclude that each variable gadget has exactly two PMUs – one PMU for each variable gadget subgraph – and there are no PMUs on clause nodes.

Due to the cross-validation constraint, it is clear that a PMU on V_{it} can only be cross-validated by a PMU on V_{ib} (since all other variable-gadgets are more than 2 hops away), and specifically this would require both to be either on $\{T_t, T_b\}$ or $\{F_t, F_b\}$.

Without loss of generality, assume for an arbitrary variable gadget, V_i, we placed the PMUs at $\{T_t, T_b\}$. By applying O1 and O2, this placement can observe all nodes in the variable gadget if $\{F_t, F_b\}$ in this gadget are not adjacent to a clause node. If they are adjacent to some a_h node, each of $\{F_t, F_b\}$ can observe its adjacent leaf-X-node only via O2, and only if a_h is already observed. Since we are given a PMU placement that observes the entire graph, this implies that a_h is indeed observed and thus adjacent to some variable node with a PMU, such that O1 could be applied to view a_h. Assume without loss of generality, a_h is adjacent to PMU nodes T_b, T_t from variable gadget V_l, then the clause $c_h \in \phi$ is satisfied if v_l is true. A similar argument can be made if V_l is adjacent to PMU nodes F_t, F_b. We conclude that all clauses in ϕ are satisfied by the truth assignment derived from Φ_G. ∎

D. MAXSAFEPLACE *Problem Statement*

MAXSAFEPLACE Optimization Problem:

- Input: Graph $G = (V, E)$ and k PMUs such that $1 \leq k < k^*$.
- Output: A placement of k PMUs, Φ_G, such that $|\Phi_G^R|$ is maximum under the condition that each $v \in \Phi_G$ is cross-validated according to the rules specified in Section III-C.

MAXSAFEPLACE Decision Problem:

- Instance: Graph $G = (V, E)$, k PMUs such that $1 \leq k < k^*$, and some $m < |V|$.
- Question: Is there a Φ_G such that $|\Phi_G| \leq k$, $m \leq |\Phi_G^R| < |V|$ under the condition that each $v \in \Phi_G$ is cross-validated?

Theorem III.3. MAXSAFEPLACE *is NP-Complete.*

Proof Idea: We show MAXSAFEPLACE is NP-hard by reducing from P3SAT. Our proof is a combination of the

NP-hardness proofs for MAXOBSERVE and SAFEPLACE. Due to space constraints, From a P3SAT formula, ϕ, we create a graph $G = (V, E)$ with the clause gadgets from MAXOBSERVE (Figure 3(b)) and the variable gadgets from SAFEPLACE (Figure 5). The edges in G are identical the ones the graph created in our reduction for SAFEPLACE.

We show that any solution that observes $m = |V| - 2s$ nodes must place the PMUs exclusively on nodes in the variable gadgets. As a result, we show 2 nodes in each clause gadget – e_j and d_j for clause C_j – are not observed, yielding a total $2s$ unobserved nodes. This implies all other nodes must be observed, and thus reduces our problem to the scenario considered in Theorem III.2, which is already proven.

Proof: MAXSAFEPLACE is easily in $\mathcal{NP}$. We verify a MAXSAFEPLACE solution using the same polynomial time algorithm described in our proof for Theorem III.2.

We reduce from P3SAT to show MAXSAFEPLACE is NP-hard. Our reduction is a combination of the reductions used for MAXOBSERVE and SAFEPLACE. Given a P3SAT formula, ϕ, with variables $\{v_1, v_2, \ldots, v_r\}$ and the set of clauses $\{c_1, c_2, \ldots, c_s\}$, we form a new graph, $H_3(\phi) = (V(\phi), E(\phi))$ as follows. Each clause c_j corresponds to the clause gadget from MAXOBSERVE (Figure 3(b)) and the variable gadgets from SAFEPLACE (Figure 5). As in Theorem III.2, we refer to the upper subgraph of variable gadget, V_i, as V_{it} and the lower subgraph as V_{ib}. Also, we let $H_3(\phi) = G = (V, E)$.

Let $k = 2r$ and $m = 12r + 2s = |V| - 2s$. As in our NP-hardness proof for MAXOBSERVE, m includes all nodes in G except d_j, e_j of each clause gadget. We need to show that ϕ is satisfiable if and only if $2r$ cross-validated PMUs can be placed on G such that $m \leq |\Phi_G^R| < |V|$.

($\Rightarrow$) Assume ϕ is satisfiable by truth assignment A_ϕ. For each $1 \leq i \leq r$, if $v_i = True$ in A_ϕ we place a PMU at T_b and at T_t of the variable gadget V_i. Otherwise, we place a PMU at F_b and at F_t of this gadget. In either case, the PMU nodes in V_i must be adjacent to a clause node, making T_t (F_t) two hops away from T_b (F_b). Therefore, all PMUs are cross-validated by XV2.

This placement of $2r$ PMUs, Φ_G, is exactly the same one derived from ϕ's satisfying instance in Theorem III.2. Since Φ_G only has PMUs on variable gadgets, all a_j and b_j nodes are observed by the same argument used in Theorem III.2. Thus, at least $12r + 2s$ nodes are observed in G. Because no PMU in Φ_G is placed on a clause gadget, C_j, we know that all e_j and d_j are not observed. We conclude that exactly m nodes are observed using Φ_G.

($\Leftarrow$) We begin by proving that any solution that observes m nodes must place the PMUs only on nodes in the variable gadgets. Assume that there are $1 < t \leq r$ variable gadgets without a PMU. Then, at most t PMUs are on nodes in clause gadgets, so *at least* $\max(s - t, 0)$ clause gadgets are without PMUs. We want to show here that for $m = 12r + 2s$, $t = 0$.

To prove this, we rely on the following observations:

- As shown in Theorem III.2, a variable gadget's subgraph with no PMU has at least 4 unobserved nodes.
- In any clause gadget C_j, nodes e_j and d_j cannot be observed if there is no PMU somewhere in C_j. Note that there are 2 such nodes.

Thus, given some t, the number of unobserved nodes is *at least* $4t + \max(2(s - t), 0)$. However, since $|V| - m \leq 2s$, there are *at most* $2s$ unobserved nodes. So we get $2s \geq 4t + \max(2(s - t), 0)$. We consider two cases:

- $s \geq t$: then we get $2s \geq 2s + 2t \Rightarrow t = 0$.
- $s < t$: then we get $2s \geq 4t \Rightarrow s \geq 2t$, and since we assume here $0 \leq s < t$ this leads to a contradiction and so this case cannot occur.

Thus, we have concluded that the $2r$ PMUs must be on variable gadget. We now observe that for each clause gadget C_j, such a placement of PMUs cannot observe nodes of type e_j, d_j, which amounts to a total of $2s$ unobserved nodes - the allowable bound. This means that all other nodes in G must be observed. Specifically this is exactly all the nodes in $H_2(\phi)$ from the Theorem III.2 proof, and PMUs can only be placed on variable gadgets, all of which are included $H_2(\phi)$ from the Theorem III.2 proof. Thus, the problem reduces to the problem in Theorem III.2 and so we the Theorem III.2 proof to determine that all clauses in ϕ are satisfied by the truth assignment derived from Φ_G. ∎

E. Extending Gadgets to Cover a Range of m and $|V|$ values

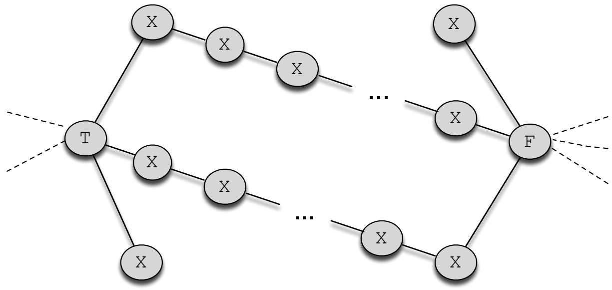
In the MAXSAFEPLACE and MAXOBSERVE proofs we demonstrated NP-completeness for $m = |V| - 2s$. We show that slight adjustments to the variable and clause gadgets can yield a much wider range of m and $|V|$ values. We present the outline for new gadget constructions and leave the detailed analysis to the reader.

To increase the size of m (e.g., the number of observed nodes), we simply add more X nodes between the T and F nodes in the variable gadgets used in our proofs for MAXSAFEPLACE and MAXOBSERVE. The new variable gadgets for MAXOBSERVE and MAXSAFEPLACE are shown in Figure 6(a) and Figure 6(b), respectively. The same PMU placement described in the NP-Completeness proofs for each problem observes these newly introduced nodes.

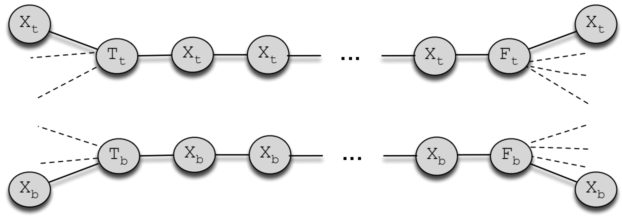
In order to increase the size of $|V|$ while keeping m the same, we replace each clause gadget, C_j for $1 \leq j \leq s$, with a new clause gadget, C'_j, shown in Figure 7(a). For MAXOBSERVE, the optimal placement of PMUs on C'_j is to place PMUs on every fourth $b_{j,h}$ node, as shown in Figure 7(b). As a result, the optimal placement of l PMUs on C'_j can at most observe $6l$ nodes. By adding $6l$ T nodes to each variable gadget, more nodes are always observed by placing a PMU on the variable gadget rather than at a clause gadget. We can use this to argue that PMUs are only placed on variable gadgets and then leverage the argument from Theorem III.1 to show MAXOBSERVE is NP-Complete for any $\frac{m}{|V|}$. A similar argument can be made for MAXSAFEPLACE.

IV. APPROXIMATION ALGORITHMS

Because MAXOBSERVE, SAFEPLACE, and MAXSAFEPLACE are NP-Complete problems, we propose greedy ap-

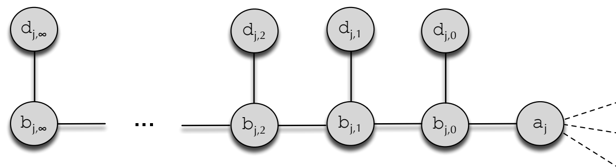


(a) Extended variable gadget used for MAXOBSERVE.

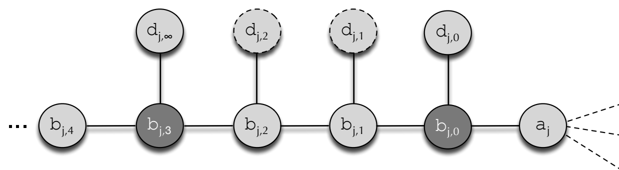


(b) Extended variable gadget used for MAXSAFEPLACE.

Fig. 6. Figures for variable gadget extensions described in Section III-E. The dashed edges indicate connections to clause gadget nodes.



(a) Extended clause gadget C_j.



(b) Observed nodes in extended clause gadget C_j shown in (a). PMU nodes are dark gray, nodes observe by O2 have a dashed border, and all other nodes are observed by O1.

Fig. 7. Figures for clause gadget extensions described in Section III-E. The dashed edges indicate connections to variable gadget nodes.

proximation algorithms for each problem, which iteratively add a PMU in each step to the node which observes the maximum number of new nodes.

MAXOBSERVE Greedy Algorithm. We start with $\Phi = \emptyset$. At each iteration, we add a PMU to the node that results in the observation of the maximum number of new nodes. The algorithm terminates when all PMUs are placed. This is the same greedy algorithm proposed by Aazami and Stilp [2].

SAFEPLACE and MAXSAFEPLACE Greedy Algorithm. The greedy algorithm is almost identical to MAXOBSERVE's, except that PMUs are added in pairs such that the selected pair observe the maximum number of nodes under the condition that the PMU pair satisfy one of the cross-validation rules.

V. SIMULATIONS

We evaluate our approximation algorithms using simulations. For each algorithm, we determine the number of nodes that are observed by placing k PMUs on IEEE bus systems 14, 30, 57, 118 and 300, [2] as well as synthetic graphs generated by using these IEEE graphs as templates.

A synthetic graph is generated from a given IEEE graph in the following manner: for each bus system, we start with the original graph and then randomly choose pairs of edges and "swap" their endpoints. Specifically, given two disjoint edges $(u, v), (w, y)$, an *edge swap* converts these to $(u, y), (v, w)$. We swap edges until the original graph and generated graph share no edges. Note that this edge-swapping procedure ensures that the degree distribution of each generated graph is exactly the same as the degree distribution of the original bus system.

When possible we compare the results of our greedy algorithms to an optimal placement of PMUs. The optimal placement was computed in brute-force manner, by enumerating all valid placements of k PMUs and then selecting the PMU placement that observes the maximum number of nodes. For SAFEPLACE and MAXSAFEPLACE, we ignore PMU placements that do not meet cross-validation requirements. Because this brute-force algorithm has exponential running time in the size of the input, we were unable to determine optimal PMU placements for larger k values in synthetic graphs generated from the IEEE 118 bus system and the actual IEEE 118 bus system. For this reason, the "optimal" curve stops abruptly in Figure 8(c), Figure 9(c), and Figure 10.

We first present the results for the synthetic graphs and then briefly discuss the results for the actual IEEE bus systems. For the synthetic graphs, each data point is generated as follows. For a given number of PMUs, k, we generate a graph, place k PMUs on the graph, and then determine the number of observed nodes. We continue this procedure until $[0.9(\overline{x}), 1.1(\overline{x})]$ – where $\overline{x}$ is the mean number of observed nodes using k PMUs – falls within the 90% confidence interval.

The results for MAXOBSERVE are shown in Figure 8. Figure 9 shows the results for SAFEPLACE and MAXSAFEPLACE. The 90% confidence interval is shown in each plot. The simulation results for graphs based on IEEE bus 14 are omitted because each greedy algorithm always correctly finds the optimal solution. Also, we do not show the IEEE bus 300 results because the graph size prevented us from running the brute-force optimization algorithm.

Our greedy algorithms perform well. For the MAXOBSERVE problem, on average, the greedy algorithm is within 96.5% of optimal. Greedy is never below 86% of the optimal solution, and in most cases gives the optimal result. Likewise, for MAXSAFEPLACE the greedy solution always gives a solution at least 84% of optimal. On average the MAXSAFEPLACE greedy algorithm is within 97% of optimal and in about half the cases, the greedy solution is optimal. These results suggest that the simple greedy approach works

[2]http://www.ee.washington.edu/research/pstca/

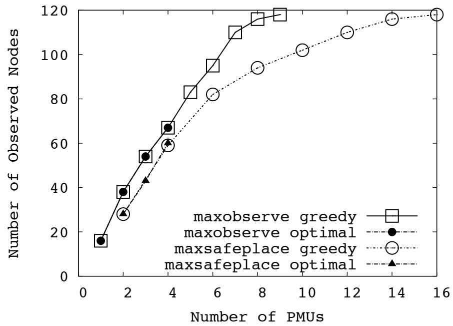


Fig. 10. Simulation results for MAXOBSERVE and MAXSAFEPLACE using IEEE bus 118.

well in practice. Typically, greedy algorithms fail because they commit to a choice too early and do not reconsider earlier decisions. Our results suggest that this is not the case for MAXOBSERVE, SAFEPLACE, and MAXSAFEPLACE.

Surprisingly, we found that `optimal` for MAXSAFEPLACE observed close to the same number of nodes as `optimal` for MAXOBSERVE. Across the same k values, on average `optimal` for MAXSAFEPLACE observed only 12% fewer nodes than `optimal` for MAXOBSERVE. This suggests that the restriction that all nodes are cross-validated does not have a significant effect on the resulting number of observed nodes.

We also ran simulations over the actual IEEE bus systems. We found the trends between our greedy and `optimal` algorithms were consistent with the results from our simulations over synthetic graphs. For example, Figure 10 shows the number of observed nodes for greedy and `optimal` for both the MAXOBSERVE and MAXSAFEPLACE problems over IEEE bus 118. In both cases, the greedy algorithm observes nearly as many nodes as the `optimal` solution. In many cases, greedy yields the optimal placement. These results are consistent with our findings for IEEE bus system 14, 30, and 57.

VI. RELATED WORK

The PMUP problem – find the minimum number and placement of PMUs to allow a bus system to be fully observable – is well-studied [3], [4], [8], [10], [12]. Although similar, the MAXOBSERVE problem differs from the PMUP problem: MAXOBSERVE considers the more general case in which a constant number of PMUs are given and the task is to place the PMUs such that the maximum number of nodes are observed. Haynes et al. [8] and Brueni and Heath [4] both prove PMUP is NP-Complete. We leverage these proofs in our NP-Completeness proofs.

The power systems literature generally ignores the fact that PMUP is NP-Complete because, in practice, power system graphs are small enough to allow for an exact solution to be found. Xu and Abur [12], [13] use integer programming to find the optimal PMU placement when a subset of buses are zero injection. O2 can only be applied to zero injection buses. As a result, the PMUP problem is simplified when only some buses are zero injection. Baldwin et al. [3] and Mili et al. [10] use simulated annealing to determine PMU placement.

Aazami and Stilp [2] investigate approximation algorithms for the PMUP problem. They derive a hardness approximation threshold of $2^{\log^{1-\epsilon} n}$ for PMUP. Also they prove that in the worst case, the same greedy algorithm presented in Section IV does no better $\Theta(n)$ of the optimal solution.

Chen and Abur [5] and Vanfretti et al. [11] both study the problem of bad PMU data. Chen and Abur [5] formulate their problem differently than SAFEPLACE and MAXSAFEPLACE. They consider graphs that are already fully observable and then add PMUs to the system to make all existing PMU measurements non-critical (a critical measurement is one in which the removal of a PMU makes the system no longer fully observable). Vanfretti et al. [11] define the cross-validation rules used in this paper. They also derive a lower bound on the number of PMUs needed to ensure all PMUs are cross-validated and the system is fully observable.

VII. CONCLUSIONS AND FUTURE WORK

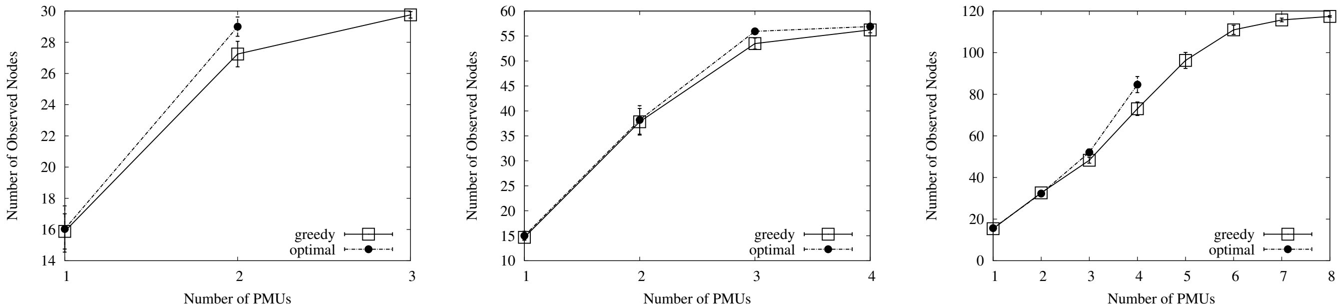
In conclusion, we have formulated three PMU placement problems: MAXOBSERVE, SAFEPLACE, and MAXSAFEPLACE. MAXOBSERVE aims to observe the maximum number of nodes given a constant number of PMUs. The observability rules from Brueni and Heath [4] were used to determine when the voltage phasor of non-PMU nodes can estimated. Like MAXOBSERVE, SAFEPLACE and MAXSAFEPLACE have the goal of observing the maximum number of nodes but also require that PMUs be placed near each other so their measurements can be validated. We called this cross-validation, a term which we formalized. Cross-validation is possible when two or more PMUs measure the current on the same transmission line or the voltage of the same (adjacent) bus.

We proved that MAXOBSERVE, SAFEPLACE, and MAXSAFEPLACE are all NP-Complete. For this reason, we presented a simple greedy algorithm which iteratively adds a PMU to the node which observes the maximum of number of nodes. In a simulation study, we compared our greedy algorithm with a brute-force optimal algorithm over several IEEE bus systems and synthetic graphs. For all three placement problems, our greedy algorithm, on average, yielded a PMU placement within 96% of optimal.

As future work, we plan to derive theoretical bounds on worst case performance of each greedy algorithm. Also, it would be interesting to relax our assumption that all buses are zero injection and evaluate our greedy algorithms over graphs with a mixture of zero and non-zero injection buses.

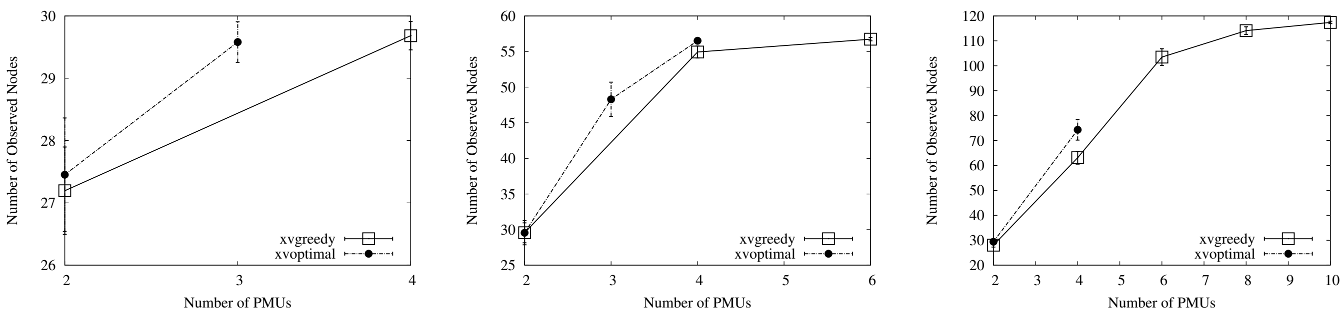
REFERENCES

[1] Synchro-Phasor Data Validation. Technical report, North-American Synchro-Phasor Initative, October 2010.
[2] A. Aazami and M.D. Stilp. Approximation Algorithms and Hardness for Domination with Propagation. *CoRR*, abs/0710.2139, 2007.
[3] T.L. Baldwin, L. Mili, Jr. Boisen, M.B., and R. Adapa. Power System Observability with Minimal Phasor Measurement Placement. *Power Systems, IEEE Transactions on*, 8(2):707 –715, May 1993.
[4] D. J. Brueni and L. S. Heath. The PMU Placement Problem. *SIAM Journal on Discrete Mathematics*, 19(3):744–761, 2005.



(a) Graphs based on IEEE Bus 30 (b) Graphs based on IEEE Bus 57 (c) Graphs based on IEEE Bus 118

Fig. 8. Simulation Results for MAXOBSERVE. The 90% confidence interval is shown.



(a) Graphs based on IEEE Bus 30 (b) Graphs based on IEEE Bus 57 (c) Graphs based on IEEE Bus 118

Fig. 9. Simulation Results for SAFEPLACE and MAXSAFEPLACE. `xvgreedy` refers to the greedy algorithm with cross-validation and `xvoptimal` is for the brute-force optimal algorithm with cross-validation. The 90% confidence interval is shown.

[5] J. Chen and A. Abur. Placement of PMUs to Enable Bad Data Detection in State Estimation. *Power Systems, IEEE Transactions on*, 21(4):1608 –1615, 2006.

[6] J. De La Ree, V. Centeno, J.S. Thorp, and A.G. Phadke. Synchronized Phasor Measurement Applications in Power Systems. *Smart Grid, IEEE Transactions on*, 1(1):20 –27, 2010.

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

[8] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning. Domination in Graphs Applied to Electric Power Networks. *SIAM J. Discret. Math.*, 15:519–529, April 2002.

[9] D. Lichtenstein. Planar Formulae and Their Uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[10] L. Mili, T. Baldwin, and R. Adapa. Phasor Measurement Placement for Voltage Stability Analysis of Power Systems. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pages 3033 –3038 vol.6, December 1990.

[11] L. Vanfretti, J. H. Chow, S. Sarawgi, and B. (B.). Fardanesh. A Phasor-Data-Based State Estimator Incorporating Phase Bias Correction. *Power Systems, IEEE Transactions on*, PP(99):1 –9, 2010.

[12] B. Xu and A. Abur. Observability Analysis and Measurement Placement for Systems with PMUs. In *Proceedings of 2004 IEEE PES Conference and Exposition, vol.2*, pages 943–946, 2004.

[13] B. Xu and A. Abur. Optimal Placement of Phasor Measurement Units for State Estimation. Technical Report PSERC Publication 05-58, October 2005.

[14] J. Zhang, G. Welch, and G. Bishop. Observability and Estimation Uncertainty Analysis for PMU Placement Alternatives. In *North American Power Symposium (NAPS), 2010*, pages 1 –8, 2010.