We present a method of determining the topology of the network given measurements typically available to the operator. We formulate the problem as a spanning tree identification problem over a graph given nodal and edge power flow information. In the deterministic case of known nodal power consumption and edge power flow we show that the placement of sensors on the network alone determines whether the set of spanning trees can be correctly identified. In the stochastic case where loads are given by forecasts derived from delayed smart meter data, we present a locally optimal sensor placement algorithm and maximum likelihood detector.

I. NOMENCLATURE TABLE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G = (V, E)$</td>
<td>Undirected Graph $G$; Vertices $V$; Edges $E$.</td>
</tr>
<tr>
<td>$T$</td>
<td>Spanning tree on $G$.</td>
</tr>
<tr>
<td>$\mathbb{T}(G)$</td>
<td>Set of all spanning trees that are constructed on $G$.</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Set of edges in constructing extended island graph.</td>
</tr>
<tr>
<td>$\mathcal{M}$</td>
<td>Sensor placement ($\mathcal{M} \subset E$).</td>
</tr>
<tr>
<td>$\mathcal{M}(G)$</td>
<td>Set of all sensor placements leading to identifiably in $\mathbb{T}(G)$.</td>
</tr>
<tr>
<td>$x_n, \hat{x}_n$</td>
<td>True and forecasted load of node $v_n$.</td>
</tr>
<tr>
<td>$\sigma_{x_n}^2, \Sigma$</td>
<td>Forecast error variance and covariance matrix.</td>
</tr>
<tr>
<td>$\Gamma(T, \mathcal{M})$</td>
<td>Observation matrix for tree $T$, sensor placement $\mathcal{M}$.</td>
</tr>
<tr>
<td>$s$</td>
<td>Set of measured power flow.</td>
</tr>
<tr>
<td>$r_{ij}$</td>
<td>Acceptance region of $T_i$ over $T_j$.</td>
</tr>
<tr>
<td>$R_i$</td>
<td>Acceptance region of hypothesis $h_i$ over all alternatives.</td>
</tr>
<tr>
<td>$g_{1,2}(\mathcal{M})$</td>
<td>Shorthand notation for (1) maximum / (2) mean missed detection probability over all hypotheses.</td>
</tr>
<tr>
<td>$c$</td>
<td>Cycle in graph $G$.</td>
</tr>
<tr>
<td>$\mathcal{C}(G)$</td>
<td>Cycle space of $G$ or set of all possible cycles.</td>
</tr>
</tbody>
</table>

II. INTRODUCTION

The need for advanced controls in the distribution system is an emerging topic in power system and the controls community. Proposed computational models for problems such as dispatching of distributed energy resources [1], [2] or coordinated voltage control [3], [4], [5], [6] assume known system topology and network parameters. In reality customer level feeders are not known to within the accuracy as that of the transmission system. Therefore, situational awareness at the distribution system level is needed for future smart grid applications.

Situational awareness in power systems is a long studied problem. This problem is generally split into the network topology analysis and state estimation task and has almost always focused on high voltage transmission systems [7]. In this work we look at network topology identification and focus on feeder level identification. Transmission and distribution system topology identification differ in a number of ways. (1) In the transmission system the network before and after a topology change is connected graph. In a distribution system however, the topology at all times is a tree. This is done so that in outage management, if a fault occurs, the faulted section can be safely isolated and restored. In a feeder system, there are generally multiple power sources which can be connected and disconnected via tie switches and breakers to ensure that different islands will always be connected to some larger energized tree [8]. (2) In transmission systems, there is a higher density of sensing than in the distribution system. For example almost all substations will have SCADA outputs of voltage and current. In the distribution system, there is very little sensing besides of feeder level and recent smart meter information. (3) In transmission systems, line parameters are...
known with high accuracy and have an observable impact on power flow studies due to long transmission line lengths. At the residential feeder level, parameters are only estimated by lengths and the interaction of various devices on the network is generally poorly understood. For these reasons, conventional and many non-conventional models have been proposed to providing network topology awareness.

The classical power systems state estimation approach has been adopted by a number of authors. Such a method relies on the Generalized State Estimator (GES) [9], [7] which will solve for system voltages and breaker statuses. In a GES, breaker status variables are included into the state estimation equations as \( \{0, 1\} \) variables that are usually relaxed and solved by rounding strategy [10]. These methods have been applied to radial distribution system with changing topology as well. There is recent interest around distribution systems topology detection. In [11], a traditional GSE is employed in identifying the correct topology in a distribution system. They make the assumptions similar to [14], [15] where we aim only power flow in a network. Our model does not consider power injections and observed voltage magnitude/phase are assumed at each bus. In [13], [14], [15] structure properties are considered since they rely on a general model formulation. In [12], distribution system. In [11] all forms of measurement are considered since they rely on a general model formulation. In [12], high frequency voltage magnitude and phase measurements are assumed at each bus. In [13], [14], [15] structure properties of the network are exploited, and linear models relating true power injections and observed voltage magnitude/phase are used. Additionally none of the method are used to provide a sensor placement strategy separate from heuristic evaluation of various scenarios. In [16], the authors provide a method similar to [14], [15] where correlation analysis is used. However they rely on long time captures, so their method is more in line of network discovery not real time topology detection. In [17] the authors present a general state estimator based method that is used in topology detection. They use the method for a heuristic placement strategy.

We should note some important features of each work. In [11], [12] no approximations are made in measurements or system models and make no use of the tree structure of the distribution system. In [11] all forms of measurement are considered since they rely on a general model formulation. In [12], high frequency voltage magnitude and phase measurements are assumed at each bus. In [13], [14], [15] structure properties of the network are exploited, and linear models relating true power injections and observed voltage magnitude/phase are used. Additionally none of the method are used to provide a sensor placement strategy separate from heuristic evaluation of various scenarios. In [16], the authors provide a method similar to [14], [15] where correlation analysis is used. However they rely on long time captures, so their method is more in line of network discovery not real time topology detection. In [17] the authors present a general state estimator based method that is used in topology detection. They use the method for a heuristic placement strategy.

Our work differs from these more traditional methods for a number of reasons. We focus on a simplified model measuring only power flow in a network. Our model does not consider voltage measurements or losses in the network. Second we make the assumptions similar to [14], [15] where we aim to detect the operational tree structure of the network. We propose a general \textit{Spanning Tree Identification Problem} and show the necessary and sufficient conditions for identifiability of all spanning trees under the measurement and power flow models.

The advantage of this simple model is that it provides a fundamental insight to the detection problem purely in terms of graph theory. This allows us to make a guarantee whether a particular sensor placement can uniquely detect every possible spanning tree i.e. network configuration. The main result we show in this paper is the following statement: \textit{A necessary and sufficient condition of detection all spanning trees via flow and nodal measurements is that the graph generated by the removal of edges containing flow sensors forms a spanning tree}. This is an intuitive result in terms of algebraic graph theory, which surprisingly is an unaddressed problem.

When load data is uncertain (in the case of forecasting) we provide an placement algorithm that is locally optimal where we can bound the maximum missed detection rate over all potential configurations. In addition to the simplicity and intuition, the method has some operational advantages since it does not require full parameter knowledge of the system. It relies only on measured and forecasted power flows. This makes it useful priori network topology processor as described in [7].

\textit{It is crucial to emphasize, that in traditional state estimation/topology identification frameworks, there is no fundamental insight into the sensor placement problem. Our work shows how a simplified power flow model can provide provable identifiability over all topologies.}

The paper is organized as follows. Section III formulates the problem of topology detection where we show how the detection of switch configurations of a feeder can be converted to that of spanning tree identification. We present the load and sensor models used in the identification problem along with the sensor placement optimization. Section IV presents the metrics used to evaluating sensor placement in the deterministic and stochastic setting. Sections V, VI solves the placement problem in both settings. Numerical demonstrations are given in Section VII.

\section{Problem Formulation}

We propose a graph representation of the valid configurations of a feeder for topology identification goals. The topology identification problem is then formulated as a \textit{spanning tree identification} problem. Given known loads at every vertex and known flow along select edges, we infer the current spanning tree on the network. In a deterministic setting, the vertex loads and edge flows are known with perfect certainty. This is unrealistic in practice, so in the stochastic setting, the vertex loads are inferred through forecasts from historical data and are thereby known with uncertainty. The performance criteria is then used to formulate appropriate sensor placement problems under each scenario. We define the measurement and load model as well as the detection criterion and sensor placement objectives in both cases.

Reduced Island Graph Consider, for example, the IEEE 123 node feeder in Figure 1.(a). There are multiple loads connected to each other and separated by switches. These switches demarcate various islands of the distribution system. We can
then think of the network as a set of connected islands. The typical feeder will have multiple power sources i.e. feeders. The set of switches are appropriately set so that each group of islands is connected to a load while ensuring no flow from one feeder to another. Therefore the distribution system must always maintain a tree structure.

This network can be reduced to a simplified graph as shown in Figure 1(b). In this reduced island graph, islanded loads are lumped together, while source of power are represented as feeder nodes. For example, sources 150, 251, 195 and 451 in Figure 1(a) are Feeder nodes $F_1, F_2, F_3, F_4$ in (b). Additionally, the set of loads connected to these feeders, and each other are converted to loads $v_1, \ldots, v_4$ in Figure 1(b).

In the reduced island graph, we must maintain a switch configuration such that no feeders are connected to each other. The connection of two feeders, for example by both $w_1$ and $w_8$ being closed will violate the tree structure of the network since two feeders will be energizing the same set of loads.

Extended Island Graph

2 Represent each of the feeders $F_1 \ldots F_4$ as vertices in the graph.
3 Represent each switch of the Reduced Island Graph as an undirected edge in the graph.
4 Add virtual root node with edges $(v_r \rightarrow F_j)$ for all feeders.

The result is the extended island graph in Figure 2(a). A complete mapping between the reduced feeder and the extended island graph is given in Table I. For example, switch $w_2$ connects nodes $v_1$ and $v_2$ in the Reduced Island Graph, or loads 151 and 300 in the test feeder. In the Extended Island Graph, it is now an edge between the vertices $v_1$ and $v_2$.

![Fig. 1. (b) Typical test feeder with sectionalizing switch operation. (b) Reduced island graph simplifying topology of feeder.](image1)

![Fig. 2. (a) Island graph showing potential topologies achievable. (b) Example spanning tree in network.](image2)

The Extended Island Graph $G = (V, E)$ is the network used in the remaining analysis. Denote the added edges $T = \{e \in E : e = (v_r, F_i) \forall F_i\}$. This construction leads to a simple method for enumerating each valid topology of the extended island graph.

Consider $T$ to be any spanning tree over $G$ and $\mathcal{T}(G)$ the set of all spanning trees constructed on $G$. We refer to the set of spanning trees containing subtree $T$ as $\mathcal{T}_T(G)$. Figure 2 represents an example spanning tree that can be constructed.

**Proposition 1**: The set $\mathcal{T}_T(G)$ represents all valid switch configurations.

**Proof**: Any $T \in \mathcal{T}_T(G)$ will maintain that every island is connected. Additionally no feeders are connected besides through the virtual node. □

Therefore enumerating every spanning tree of the island graph which includes subtree $T$ will generate the set of topologies corresponding to valid switch configurations. Note the naive method of testing all $2^{|W|}$ switch positions and testing for full connectivity of all loads as well as invalid feeder loops is much less practical than simply enumerating the set $\mathcal{T}_T(G)$.

**Load Model** Each node $v_n$ excluding the virtual root and feeder nodes in the graph has a consumption load $x_n$. We assume for now that the load is single time invariant value. The pseudo-measurement of each load is $\hat{x}_n$. We denote the pseudo-measurement error of load $x_n$ by $\epsilon_n = x_n - \hat{x}_n$. We assume that the loads are single phase real power quantities and the errors are mutually independent random variables: $\epsilon_n \sim N(0, \sigma_n^2)$. Therefore, $x_n \sim N(\hat{x}_n, \sigma_n^2)$. $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the vector of true loads and that of load pseudo-

---

**Table I**

<table>
<thead>
<tr>
<th>Island Graph</th>
<th>IEEE Test Feeder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>Node</td>
</tr>
<tr>
<td>$w_1$</td>
<td>$F_2, v_8 - v_1$</td>
</tr>
<tr>
<td>$w_2$</td>
<td>$v_1 - v_2$</td>
</tr>
<tr>
<td>$w_3$</td>
<td>$v_2 - v_3$</td>
</tr>
<tr>
<td>$w_4$</td>
<td>$v_4 - v_5$</td>
</tr>
<tr>
<td>$w_5$</td>
<td>$v_4 - F_3, v_7$</td>
</tr>
<tr>
<td>$w_6$</td>
<td>$v_7 - v_3$</td>
</tr>
<tr>
<td>$w_7$</td>
<td>$v_1 - v_3$</td>
</tr>
<tr>
<td>$w_8$</td>
<td>$F_1, v_6 - v_1$</td>
</tr>
<tr>
<td>$w_9$</td>
<td>$F_1, v_6 - F_4, v_9$</td>
</tr>
</tbody>
</table>

To simplify the analysis of the switch configuration problem we modify the reduced island graph. This is done by the following steps.

1 Represent the loads from the Reduced Island Graph as vertices in the graph.
measurements; thus $x \sim N(\hat{x}, \Sigma)$. Covariance matrix $\Sigma$ being diagonal.

**Measurement Model:** For any edge $e$, we denote by $s$ the power flow on it to all active downstream loads. The measured flow depends on unknown network topology and true loads.

The sensor placement $\mathcal{M} \subset E$ is a subset of edges of the network. We assume that the magnitude and direction of power flow is measured. Additionally, we assume that the power flow measurements are noise free, with errors only with load pseudo-measurements.

In general for a topology $\mathcal{T} \in \mathcal{T}(G)$ and measurements $s = \{s_1, \ldots, s_{|\mathcal{M}|}\}$ the $k^{th}$ flow is $s_k(x) = \sum_{v_j \in V_k(\mathcal{T})} x_j$. Where $V_k(\mathcal{T})$ is the subset of downstream nodes for a particular placement and spanning tree. Given a placement, we can represent the observations with $s = \Gamma(\mathcal{T}, \mathcal{M}) \mathbf{x}$ where $\Gamma(\mathcal{T}, \mathcal{M}) \in \{0, 1\}^{|\mathcal{M}| \times |\mathcal{V}|-|\mathcal{E}|}$.

We use shorthand $\mathcal{G} = \{e_1, e_9\}$.

Topography Detection and Sensor Placement: We can now precisely state the topology detection problem. Given an unknown switch set of switch positions $w = \{w_1, \ldots, w_n\}$ and the resulting spanning tree $\mathcal{T}(w)$ we measure a set of flows in real time as well as set of load forecasts $\hat{x}$ from individual smart meters. Given an unknown switch configuration and resulting spanning tree, we must infer the topology from the measured flows and load pseudo measurements.

- Deterministic case: We choose the topology matching the measured flows under the known loads.
- Stochastic case: We infer the most likely topology via general MAP detector given by:

$$\hat{T} = \arg \max_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\mathcal{T} | \hat{x}, s) \quad (1)$$

The goal of the optimal sensor placement is to enable the topology detection and provide the best tradeoff between the number of sensors and reduction in the detection error. Based on characterizing the performance of optimal detection and the minimal identifiability requirements we can develop an optimal sensor placement problem formulation for both the deterministic and stochastic cases.

**IV. Topology Identification: Identifiability and Optimal Detection**

**A. Deterministic Case: Tree Identifiability**

We only require a sensor placement such that different spanning trees lead to unique measured flow values. The following definition is of use:

**Definition 1:** The set $\mathcal{T}(G)$ is identifiable if $\forall \mathcal{T}, \mathcal{T}' \in \mathcal{T}(G)$ where $\mathcal{T} \neq \mathcal{T}'$ we have that $s \neq s'$.

In the deterministic case, we desire a placement $\mathcal{M}$ such that $\mathcal{T}(G)$ is identifiable. A naive method of determining whether $\mathcal{T}(G)$ is identifiable is quite intensive. We must enumerate every single spanning tree, generate $\Gamma(\mathcal{T}, \mathcal{M})$ to verify that for no two trees $\Gamma(\mathcal{T}, \mathcal{M}) = \Gamma(\mathcal{T}', \mathcal{M})$ (note $\Gamma(\mathcal{T}, \mathcal{M}) \neq \Gamma(\mathcal{T}', \mathcal{M}) \implies s \neq s'$). This naive procedure has $O(|\mathcal{M}||\mathcal{V}|^{3}|\mathcal{T}(G)|)$ complexity, which can be quite intensive since $|\mathcal{T}(G)|$ is generally very large.

**B. Stochastic Case: Minimax and Mean Error**

In the stochastic case, we assume that line flows, which represent SCADA connected equipment, reports flows in a real time manner are error free. In comparison, load measurements, which represent Smart Meter based load forecasts of current consumption, will have a forecast error.

The detector in eq. (1) will not only deal with undetectability, but also have a finite error rate. Given the maximum likelihood detector in eq. (1), we can compute the missed detection error for any hypothesis $\Pr(\hat{T} \neq T | \mathcal{T}; \mathcal{M})$. This allows us to evaluate a placement by minimizing one of two commonly used objective functions:

1) An alternative placement criterion is to minimize the mean missed detection error over all hypothesis. This is given by the following optimization:

$$\mathcal{M}^* = \arg \min_{\mathcal{M} \subset \mathcal{G}} \sum_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\mathcal{T}) \Pr(\hat{T} \neq T | \mathcal{T}; \mathcal{M}) \quad (2)$$

2) The maximum error over all possible spanning trees in the network is given by:

$$\mathcal{M}^* = \arg \min_{\mathcal{M} \subset \mathcal{G}} \max_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\hat{T} \neq T | \mathcal{T}; \mathcal{M}) \quad (3)$$

This placement is very conservative since it is a guarantee on the worst possible detection error. If all spanning trees are possible in the network, this may be the best option.

We use shorthand $g_1(\mathcal{M})$ and $g_2(\mathcal{M})$ for the mean and maximum missed detection error.

**V. Deterministic Placement**

**A. Valid Placement for Spanning Tree Identifiability**

The primary result we show in this work is the following Theorem relating sensor placements and identifiability of all spanning trees.

**Theorem 1:** $\mathcal{T}(G)$ is identifiable if and only if the graph $E(G) \setminus E(\mathcal{M})$ of the Extended Island Graph forms a spanning tree.

We refer to any placement $\mathcal{M}$ satisfying Theorem 1 as a valid placement.

The intuition of Theorem 1 is that to have observability of all spanning trees, we must have a sensor placement such that any cycle that can be constructed on $G$ will have some flow sensor on it. This relates to the set the set of spanning trees that can be constructed. In graph theory literature there is a notion of a vector space of cycles. As it turns out, this space has a dimension equal to $\mu(G)$. Additionally, the minimum number of sensors required to identifying all spanning trees is $\mu(G)$. This gives us a $O(E)$ verifiable condition to ensure that all spanning trees are identifiable as opposed to $O(|\mathcal{M}||\mathcal{V}|^{3}|\mathcal{T}(G)|)$ with the naive method since we only need to remove particular edges of the graph and test if it is a spanning tree.
B. Enumerating Valid Placements

Theorem 1 provides a verifiable conditions under which all spanning trees are identifiable, but does not provide a way in which to construct the set of placement. Given a valid spanning tree $T$, then set of edges defined by $f(T) = E(G) \setminus E(T)$ can be applied to each tree. 

Corollary 1: The function $f : T \rightarrow M$ is a bijection between the set $\mathbb{T}(G)$ and the set of all valid placements $M(G)$.

This is an obvious consequence to Theorem 1.

Remark 1: Corollary 1 implies that $|M(G)| = |\mathbb{T}(G)|$.

Corollary 1 is quite important from a placement perspective since it actually yields a method to generate a valid placement in the deterministic case. Also, it allows us to enumerate all valid placements for a graph. This is important when dealing with a stochastic case where sensor placement relies on mostly evaluating each placement in $M(G)$.

In the case of valid placements on the extended island Graph this set is restricted, since having sensors on edges in $\tau$, since they have no physical meaning. The restricted set is given by $M_r(G) = \{ f(T) | T \in \mathbb{T}_r(G) \}$.

VI. STOCHASTIC PLACEMENT

A. Monotone Decreasing Objective

We can show that the placement objective functions $g_1(M)$ and $g_2(M)$ are monotonically decreasing functions. This guarantees that a greedy placement algorithm, such as Algorithm 5 is guaranteed to find a locally minimum solution. Given the stochastic model, we have the following pairwise test between spanning tree $T_i$, $T_j$:

$$s|h = \begin{cases} N(\mu_i, \Sigma_i), & h = T_i \\ N(\mu_j, \Sigma_j), & h = T_j \end{cases}$$

(4)

Where $\mu_i = \Gamma_i x$ and $\Sigma_i = \Gamma_i \Sigma \Gamma_i^T$. Recall $\Gamma_i$ and $\Gamma_j$ represent the observation matrices under different spanning trees. The maximum likelihood detector has the following form:

$$\hat{T}(\mathbf{s}) = \begin{cases} T_i, & s \in r_{ij} \\ T_j, & s \notin r_{ij} \end{cases}$$

(5)

Where the region of each $T_i$, $T_j$ is specified by the maximum likelihood criterion

$$r_{ij} = \{ s \in R^{|M|} : \Pr(s|h_i) \geq \Pr(s|h_j) \} \cap R_i \setminus T_i$$

(6)

We can then define acceptance region for deciding $h = T_i$ or $h \neq T_i$, as $R_i = \bigcap_{j \neq i} r_{ij}$.

Intuitively if we add more sensors into the detection problem, we will loose no additional additional information and can do no better. This allows us to state the following:

Remark 2: Under different sensor placements $M \subset M'$ we have $g_k(M) \leq g_k(M')$ for $k = 1, 2$.

B. Greedy Placement

We propose Algorithm 5 as a greedy method for minimizing either eq. (3) or eq. (2). The greedy placement is initialized with a valid placement which minimizes the objective. At every subsequent iteration, we evaluate the objective over all the remaining edges in the graph $E(G) \setminus E(M^g)$. Since any $|M| \leq \mu(G)$ will result in a deficient starting point, if the objective is the maximum error $g_1(M)$ we must initialize the algorithm with line 1. If we aim to minimize the mean error $g_2(M)$, then an alternative initialization is $M \leftarrow \{ \emptyset \}$.

Algorithm 1: Greedy Sensor Placement

Input: [1] Graph $G$  
[2] Nominal Load Statistics $L$, $\Sigma$  

Output: Greedy Placement $M^g$

1. $M^g \leftarrow \arg\min_{M \in M(G)} g_1(M)$
2. while $|M^g| \leq M$ do
3. $e^* \leftarrow \arg\min_{e \in E(G) \setminus E(M^g)} g_1(M^g \cup \{e\})$
4. $M^g \leftarrow M^g \cup \{e^*\}$
5. end

VII. NUMERICAL EXPERIMENT

We conduct a number of numerical experiments evaluating the general performance of the placement methods presented. (1) We evaluate the deterministic placement for some randomly generated planar graphs with high spanning tree counts. (2) Then we evaluate the stochastic placement for both max and mean placement error on a graph with high edge counts and low circuit rank and compare the result to a randomized placement. (3) We present a numerical counterexample illustrating the optimization problem is not sub/sup modular. (4) We evaluate the stochastic placement, on the IEEE 123 Node Test Feeder.

A. Deterministic Placement

We test the placement problem on a set of planar graphs, shown in Figure 2. In both graphs, designate a single vertex as the source, which is indicated in the dashed horizontal lines. Graph $G_1$ has $v_{\text{root}} = v_1$, $G_2$ has $v_{\text{root}} = v_4$ and $G_3$ has $v_{\text{root}} = v_1$.

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>DETERMINISTIC TOPOLOGY DETECTION.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu(G)$</td>
<td>$</td>
</tr>
<tr>
<td>$G_1$</td>
<td>5</td>
</tr>
<tr>
<td>$G_2$</td>
<td>5</td>
</tr>
<tr>
<td>$G_3$</td>
<td>2</td>
</tr>
</tbody>
</table>

To test the placement problem, we enumerate the set of spanning trees for each of the graphs. The method relies on the backtracking method developed in [18]. The simulation was implemented in MATLAB and deemed correct by checking that each spanning tree was unique and the number of test trees corresponded to those calculated from the matrix-tree theorem [19], where $|\mathbb{T}(G)| = \det(L_v)$. Where $L_v$ is the $v$
minor of the laplacian matrix with the result being invariant to $v$.

For the three graphs in Figure 5, the graph statistics and experiment results are shown in Table II. We evaluate experimental error rate

$$\epsilon = \frac{1}{N(N-1)} \sum_{i \neq j} \mathbb{I}\{s(T_i) \neq s(T_j)\}. \quad (7)$$

From Theorem 1, the missed detection error must be zero. The computed $\epsilon$ in Table II verifies this.

We evaluate the output according to an arbitrary input because we would like to compare it to the case where only magnitude and not direction is measured. In this case, we evaluate $\epsilon'$ which now compares $|s(T_i)| \neq |s(T_j)|$ instead. The computed values for $\epsilon'$ are shown in Table II. We evaluate each valid placement in $\mathcal{M}(G)$ to illustrate the importance of flow direction. The value reported in Table II is the mean missed detection $\pm$ the standard deviation. This verifies that different placements result in different unsigned missed detection rates. We see that if the direction of flow is not known, around 10\% of the spanning trees are indistinguishable on average.

**B. Stochastic Sensor Placement**

We evaluate the performance of Algorithm 5 for both the max-error and mean-error objectives. The algorithm is tested on Graph $G_3$ with each node having an identical $\mu_t = 1$, $\sigma = 0.1$. For comparison, we evaluate a maximum of 100 randomly allocated placements for each size: $\min\left(100, \left\lceil \frac{|E|}{\mu(G)} \right\rceil \right)$. The performance is indicated in Figure 4 for both metrics. The graphs indicated a clear improvement as opposed to randomized placement. For the mean error metric, $|\mathcal{M}^\theta| \geq 6$ has an error rate less than 0.005 which is a sensor density of 17\%. A randomized method has much poorer performance on average. The results are much worse in the max error case, which is expected. For very large sensor densities, the maximum error is still quite high. For the max error metric, $|\mathcal{M}^\theta| \geq 6$ has an error rate less than 0.005 which is a sensor density of 17\%.

**C. Supermodularity Counterexample**

Sub modularity is a property commonly exploited in many combinatorial optimization problems (see [20]) for more details. It is useful since it guarantees that a greedy algorithm is within a factor of $\left(1 - \frac{1}{e}\right)$ of the optimal value.

**Definition 2:** For every $\mathcal{M} \subset \mathcal{M}' \subset E(G)$ we have $\forall e \in E(G) \setminus B \ g(M \cup \{e\}) - g(M) \leq g(M' \cup \{e\}) - g(M')$.

The relative decrease in the objective function must be larger for the smaller set under all subsets $\mathcal{M}$, $\mathcal{M}'$ and additional element $e$. Section VII-C, presents a numerical counterexample.

The sample graph $G_3$ in Figure 3(c) used to show a computable counterexample to sub modularity. Consider the following sets $\mathcal{M} = \{(1,2),(20,21)\}$ and $\mathcal{M}' = \{(1,2),(20,21),(10,11)\}$. For the remaining allowable edge $e \in E(G) \setminus \mathcal{M}'$, we can compute the marginal decrease for each additional measurement where

$$\Delta_k(e) = g_k(M \cup \{e\}) - g_k(M) \quad (8)$$

$$\Delta'_k(e) = g_k(M' \cup \{e\}) - g_k(M'). \quad (9)$$

For super modularity to hold, we must have $\Delta_k(e) \leq \Delta'_k(e)$ for all $e$. However, as the example shows, in a certain set of $u$ we have that $\Delta'_k(e) < \Delta_k(e)$ for $k = 1, 2$. 

![Fig. 3. Sample graphs used to benchmark deterministic and stochastic cases.](image)

![Fig. 4. Performance of the greedy placement in Algorithm 5 is shown for $G_3$. With the Max-error objective ($g_1(M)$ in Figure 4(a)) and mean error ($g_2(M)$ in Figure 4(b)). A random placement is shown for each $|M|$ as comparison, with an evaluation size of 100 placements. The mean performance over the 100 samples is indicated.](image)
D. 123 Test Feeder

As presented in Section II, we introduced a method of transforming the IEEE 123 test feeder into a reduced feeder (Figure 1(b)) and an island graph (Figure 2(a)).

The results we present allow us some insight into the topology detection problem. From Theorem 1, the minimum number of flow measurements is $\mu(G) = 4$. Since the actual set of allowable trees is $T_r(G)$, this restricts the set of trees to search over, in evaluating $g_1(M)$, $g_2(M)$.

We can now evaluate Algorithm 5 with the augmented sets, $M_r(G)$ and $T_r(G)$.

For simplicity assume a single phase flow of only real power. Given the three phase complex power consumption of each load, we reduce this to a single value power value. This is the case, if the measurement on a line sum and report the sum of the three phases. For a rough estimate of the forecast error, we follow the forecasting model in [21], where day ahead forecasts are computed for varying aggregation levels. The forecast errors are used to construct the following scaling law for coefficient of variation:

$$\overline{CV}(W) = \sqrt{\frac{3562}{W} + 41.9}. \quad (10)$$

Since the loads of each individual island is quite large and beyond the critical load reported in [21], the CV of each island is close to the irreducible error of 6.3%. Figure 6 shows the performance of an initial stochastic placement for the IEEE123 test feeder. We evaluate the set of restricted placements $M_r(G)$ and spanning trees $T_r(G)$, where $|M_r(G)| = 44$. Notice that for almost half of the placement the maximum error is negligibly small. Since the pairwise error probabilities are computed via monte carlo, they are small positive values that experimentally come out as 0.

Although voltage measurements provide a global monitoring of the network (i.e the jacobian is full rank and under very low measurement error every hypothesis discernible). In practice the set of all topologies generally map to range of $\pm 5\%$. Therefore, when factoring in uncertainties, the missed detection rates can be high. On the other hand, measuring flows lead to very large changes in the observation vector over the range of hypothesis. This gives us some rule of thumb in sensor selection for topology detection methods:

1. Flows measurements (power, current) may be confounded to observing local changes, but they have larger dynamic range.
2. Voltages can observe changes globally in the network but have a much more narrow dynamic range.

VIII. Conclusion

We formulate the problem of detecting switch configurations on residential feeders as a spanning tree detection problem on an ‘extended island graph’. The detection problem relies on power flow measurements on edges as well as load information at nodes. The deterministic case leads to notion of tree identifiability and sensor placement conditions to ensure identifiability. For the stochastic case, we propose a greedy algorithm which leads to a locally optimal solution.

REFERENCES

APPENDIX

A. Useful Graph Theory Definitions and Results

Referr to [19] for a more thorough presentation.

Cycle: A cycle $c = \{e_1, \ldots, e_N\}$ is connected subgraph where each vertex is of degree 2.

Cycle Space: The set $B_E$ is the power set over the edge set $B_E = \{0,1\}^{|E|}$. Any cycle $c$ is a vector in the space $B_E$. Vector addition is defined as $c' = c_1 \oplus c_2$ where new cycles are via symmetric difference operation on edges: $E_1 \oplus E_2 = (E_1 \cup E_2) \setminus (E_1 \cap E_2)$. The cycle space $C(G)$ of the graph is the vector space of all possible cycles in a particular graph.

Circuit Rank: The circuit rank of a graph is given by $\mu(G) = |E| - |V| + n(G)$ where $n(G)$ is the number of connected components of the graph. For example, the extended island graph in Figure 2(a) has $n(G) = 1$, and $\mu(G) = |13| - |10| + 1 = 4$.

Cycle Basis: The analog of a vector basis for cycle spaces is the cycle basis. A basis $B_C \subset B_E$ is the smallest number of cycles whereby all other cycles can be constructed via symmetric difference operations. The dimension of $C(G)$ and $B_C$ is $\mu(G)$, the circuit rank of the graph. Therefore, $\mu(G)$ is the smallest number of cycles required to produce all other cycles on a graph. We use $\mu(G)$ and $\mu$ interchangeably, whenever convenient.

Fundamental Cycle Basis: A Fundamental Cycle Basis $FC$ is a cycle basis constructed using the following procedure: Given a spanning tree $T$, enumerate the set of edges in $G$ but not $T$. Then for each $e \in E(G) \setminus E(T)$, construct $T + e$ and find the single cycle $c$ associated with $e$. So we can generate $\mu(G)$ cycles in $FC$, which is the dimensionality of the basis (and all other cycle bases). An equivalent definition for a Fundamental Cycle Basis is that each cycle will have one unique edge which is in no other cycle.

For example, the graph in Figure 2(b), the set $E_R$ is shown in dashed lines. From tree $T$ we construct the following Fundamental Cycle Basis $FC(T)$:

- $c_1 = \{(1,8), (8,R), (R,6), (6,1)\}$
- $c_2 = \{(1,3), (3,4), (4,7), (7,R), (7,R)\}$
- $c_3 = \{(4,9), (9,R), (R,7), (7,4)\}$
- $c_4 = \{(5,4), (4,7), (7,R), (R,6), (6,1), (1,2), (2,5)\}$

Generation of Spanning Tree: Let the set $E_R$ be the set of removed edges resulting in a spanning tree and let $FC$ be any Fundamental Cycle Basis on $G$.

Proposition 2: With respect to an arbitrary $FC$, edges in $E_R$ which form a spanning tree satisfy the following:

- **P1** Every cycle must have a removed edge and $|E_R| = \mu(G)$.
- **P2** Any cycle can have multiple removed edges on it.
- **P3** We cannot have removed edges $e_i, e_j$ along a path $p = c_i \cap c_j$ for any $c_i, c_j \in FC$.
- **P4** Every $c_i \in FC$ will contain a unique $e_i \in E_R$ which is no other cycle.

Proof: Properties (P1), (P2) and (P3) are needed to maintain a spanning tree and are fairly obvious. Property (P4) is true because mapping tree edges $E_R$ to cycle is identical to the Cycle-Sensor map (see below) where we replacing $E_R$ with $M$ and apply Corollary .

So in terms of any $FC$ any two $E_R, E'_R$ will have some edges that are unique to a cycle in $FC$.

**Definition 3:** An edge exchange with respect to $FC$ is $\Delta E = \{\Delta e_1, \ldots, \Delta e_m\}$ is the set of $\Delta e_k = \{e_k \to e'_k\}$, $e_k \in E_R$ and $e'_k \in E_R$.

Edges $e_k$ and $e'_k$ are the unique edges in $e_k$ which must exist due to Proposition 2 (4). A trivial edge exchange will have $e_k = e'_k$ for all $\Delta e_k$. Conversely for a non-trivial edge exchange $\exists \Delta e_k : e_k \neq e'_k$.

We want to restrict our attention to non-trivial $\Delta E$ where we generate $T'$ from $T$. Additionally, we look at whether $FC(T) = FC(T')$ or $FC(T) \neq FC(T')$. We use following notation: $\Delta E$ will produce $T \to T$ and $FC \to FC'$ (or $FC$). First we need the following:

**Definition 4:** The basis intersection is:

$$I(FC) = \{e : e \in c_i \cap c_j, \forall c_i, c_j \in FC\}.$$  

This is the set of edges that are in the intersection of two or more cycles in $FC$. We now state the following result:

**Lemma 1:** ([22] Corollary 2) For edge exchange $\Delta E$, we have $FC(T) = FC(T')$ iff $e_i, e'_j \in c_i$ but not in $I(FC)$.

As long as we restrict edge exchanges along a $c_i \in FC$ but not in $I(FC)$, we can generate new trees which always map to the same $FC$. Therefore we have the following transitions (shown in Figure 7(a)):

**Case 1:** $\Delta E$ where we have $FC \to FC$. We require at least one non-trivial exchange where $e \neq e'$, but $e' \notin I(FC)$.

**Case 2:** $\Delta E'$ where we have $FC \to FC'$. We only require 1 non-trivial exchange where $e' \in I(FC)$. The remaining edges can be trivial exchanges or follow Case 1.

**Deficient:** Spanning forest if any two $e_i', e_j' \in c_i \cap c_j$ since this violates Proposition 2(P3).

B. Proof of Theorem 1

We now prove Theorem 1 in the following steps. We consider an arbitrary edge exchange $\Delta E$. We show that regardless if satisfies Case 1 or Case 2, $\Delta s \neq 0$.

**Step 1:** Introduce a cycle-sensor mapping object $K(c)$ which tracks sensors on a cycle and derive set independence properties of $K(c)$ Namely, that $K(c)$ and $K(c')$ will always differ.
Step 2: We show that sensor measurements in $K(c)$ decouple from one cycle to another under specific edge exchanges. We show that edge exchanges fully decouple along sensors in the set $K(c)$.

Step 3: We use the independence of $K(c)$ and decoupling of sensors inside and outside of $K(c)$ to show inductively that no edge exchange of any size exist where $\Delta s = 0$.

Step 1 Cycle-Sensor Map

Definition 5: With respect to some $\mathcal{FC}$, a cycle-sensor mapping $K : c \rightarrow M \forall c \in \mathcal{FC}$ where $s_k \in K(c)$ if $s_k$ is on an edge in $c$.

For example, for the fundamental cycles associated with the tree in Figure 2(b), we have the following: $K(c_1) = \{s_1\}$, $K(c_2) = \{s_1, s_2\}$, $K(c_3) = \{s_2\}$ and $K(c_4) = \{s_1, s_2\}$.

Consider placement $M$, and constructed tree $T = \overline{G \setminus M}$. We must have have $K(c_k) = \{s_k\}$ where $c_k$ is the $kth$ cycle formed from adding edge containing $s_k$ to tree $T$. We denote $\lambda_k$ to be the $kth$ cycle in the Fundamental Cycle Basis $\mathcal{FC}_M$; $K(\lambda_k) = \{s_k\}$

We see that $K(c)$ actually encodes the construction of any arbitrary fundamental cycle using the elements $\lambda_1, \ldots, \lambda_n$.

Lemma 2: For any $\mathcal{FC}$ and $c \in \mathcal{FC}$, $c = \bigoplus_{k:s_k \in K(c)} \lambda_k$.

Proof: Any cycle $c$ can be represented as a combination of cycles $\lambda \in \mathcal{FC}_M$ since $\mathcal{FC}_M$ is a cycle basis. If any other $N$ outside of the set $\{s_k \in K(c)\}$ is used to construct $c$ then $K(c)$ will include edge containing a $s'$. Conversely if any additional $N'$ is needed to construct $c$, it’s $s$ must be in $K(c)$.

The following lemma is needed in Step 3.

Lemma 3: For all $\mathcal{FC}$ and subsets $A \neq B$ of cycles in $\mathcal{FC}$ we must have that:

$$\bigcup_{k \in A} K(c_k) \neq \bigcup_{k \in B} K(c_k).$$

Proof: Suppose that there exists some $\mathcal{FC}$ and partitions $A, B$ where the terms $K_A(c)$ and $K_B(c)$ are equal. Since $K_A(c)$ and $K_B(c)$ encode some cycle we have that $c_A = c_B$. However, since $c_A$ and $c_B$ are definition fundamental cycle’s in $\mathcal{FC}$ there will exist an edge in $c_A$ that is not in $c_B$ thus $c_A \neq c_B$.

Corollary 2: A special case is that $\forall c, A \subset \mathcal{FC}, \exists s \in K(c)$ s.t. $s \not\in \bigcup_{k \in A} K(c_k)$, for any $A$ not including $c$.

Corollary 3: The subset $C \subset \mathcal{FC}$, where $|C| = N$ will have at least $N$ unique sensors.

Corollary 4: The indicator vector associated with each $K(c)$ is a set of linearly independent vectors.

Notice for example, that the cycle-sensor-map $K$ constructed for the network in Figure 2(b) does not satisfy Lemma 3 since the partition $A = \{c_1\}$, $B = \{c_2\}$ is a clear counter example. This is because the sensor placement does not lead to a spanning tree, so $\mathcal{FC}_M$ is not fully constructed.

Step 2 Decoupling of Sensors along Fundamental Cycle Basis

We show that when an edge exchange occurs on a cycle, we need to only consider changes of flow values in $K(c)$. In the development $K(c)$ and edge exchanges are focused on cycles in $\mathcal{FC}$.

Proposition 3: A single edge exchange on $c$, with vertices $\{v_0, \ldots, v_m\}$ results in a permutation of an uninterrupted path of the vertex. Therefore, if $e \neq e'$, $p(c, e) \neq p(c, e')$.

Lemma 4: Consider a single edge exchange on $c$, $e \rightarrow e'$ where (Lemma 1). The following holds:

$P_1 \forall s \not\in K(c), \Delta s = 0.$

$P_2 \forall s \in K(c), \Delta s \neq 0.$

Proof: (1) From Proposition 3, any sensor that measured a single vertex in $p$ will measure all the vertices in $p$ before and after the edge exchange. (2) From Proposition 3 since we rearrange all the nodes yet keep a fixed edge to measure flows, all sensors in $K(c)$ will change values.

Consider the example in Figure 7(b), where the cycle in consideration $c_1$ will have a single edge exchange: for example $\Delta e = (8, 7) \rightarrow (1, 2)$ where all $x_i = 1$. In this case, $s = 2$ and $s' = -2$. Since we assume that the magnitude and direction of the flow is measured, therefore there is no ambiguity. Any other edge will again lead to a new measured value. Since nodes $\{v_0, \ldots, v_{10}\}$ are always connected, all sensors outside of $K(c)$, for example a measurement on edge $e = (2, 10)$ will not detect an edge exchange.

Remark 3: In condition (2) of Lemma 4, we have $\Delta s \neq 0$ since we measure both magnitude and direction of flow.

Note that Lemma 4 contains both $e' \in I(\mathcal{FC})$ and $e' \not\in I(\mathcal{FC})$ cases. So if $e' \in c_i \cap c_j$, in terms of changes, a sensor in $K(c_j)$ must have some changed value. However, we do not treat this as an edge exchange on $c_j$ since not all values in $K(c_j)$ will change. For example, if $\Delta e_i = \{(8, 7) \rightarrow (3, 4)\}$, cycle $c_j$ the sensor on $(3, 4)$ will now report 0 but the sensor on $(2, 10)$ will remain unchanged.

Step 3 Inductive Proof of Case 1, 2

Note our effort is to show sufficiency, for necessity we only need to consider the cycle $c$ in Figure 7(b) If no sensor exists on $c_1$, $E(G) \setminus E(M)$ is not a spanning tree and every edge exchange leads on $c_i$ where $e_i \not\in I(\mathcal{FC})$, will lead $\Delta s = 0$. We consider single and multiple edge exchange procedures and show that no edge exchange exists which leads to $\Delta s = 0$. Care is taken in the argument because when an edge exchange occurs on a single cycle, sensors in $K(c)$ will change.

Case 1: We show that given any number of non-trivial edge exchanges such where $e' \not\in I(\mathcal{FC})$, there is at least one $s_i$ s.t. $\Delta s_i \neq 0$. 

**Base Case** Assume $c \subset FC$, contain a non-trivial edge exchange. From Lemma 4 (P2), $\forall s \in K(c), \Delta s \neq 0$.

**Inductive Hypothesis** Assume multiple cycles $C \subset FC$, contain a non-trivial edge exchange, where $|C| = N$. Assume there exists at least one $s_i \in K_C(c)$, where $\Delta s_i \neq 0$. Subset $C$ will have $N$ unique sensors.

**Inductive Step** Suppose we find a cycle $c_{n+1} \in FC \setminus C$ where an edge exchange leads to $\Delta s_i = 0$. From Corollary 3 subset $C \cup c_{n+1}$ will at least $N + 1$ unique sensors and the new cycle must introduce some sensor $s_j \notin K_C(c)$. From Lemma 4 (2) $\Delta s_j \neq 0$.

**Case 2:** We show that any number of edge exchanges where $e' \in I(FC)$, there is at least one $s_i$ s.t. $\Delta s_i \neq 0$ Any number of exchanges where $\Delta e \in I(FC)$ are allowed. We do not care about these cases since Lemma 4 decouples these cycles and Case 1 ensures any changes lead to $\Delta s \neq 0$.

**Base Case** An edge exchange on $c_1 e'_1 \in I(FC)$. From Lemma 4 (P2), $\forall s \in K(c), \Delta s \neq 0$. For all other cycles $c_i$, where $e'_i \in I(FC)$, we use the result from *Case 1*.

**Inductive Hypothesis** Assume $N$ edge exchanges $\Delta e_1, \ldots, \Delta e_N$ such that $e'_i \in I(FC)$. Additionally, cycles $C = \{c_1, \ldots, c_m\}$ are effected because $e'_i$ is on paths $p_i$ formed by the intersection of some subset of $C$. Assume there exists at least one $s_i \in K_C$, where $\Delta s_i \neq 0$.

**Inductive Step** Suppose we find a cycle $c'$ where an edge exchange leads to $\Delta s_i = 0$. We must show that the cycle $c'$ must not exist in $C$ hence the new cycle must introduce some sensor $s_j \notin K_C(c)$. If $c' \notin C$, then $s_i$ must be on the path between the intersection of $c'$ and some $c \in C$. However, from Proposition 2 (P4), the resulting graph will no longer be a spanning tree. So we are forced to introduce a new cycle. So again from Lemma 4(P2) $\Delta s_j \neq 0$. 