Outage Detection Using Load and Line Flow Measurements in Power Distribution Systems

Raffi Avo Sevlian, Yue Zhao, Ram Rajagopal, Andrea Goldsmith, and H. Vincent Poor

***Abstract*—An outage detection framework for power distribution networks is proposed. Given the tree structure of a distribution system, a detection method is developed combining the use of real-time power flow measurements on the edges of the tree with load forecasts at the nodes of the tree. The maximum likelihood (ML) detection decision for an arbitrary number of outages is shown to be efficiently computable due to decoupling across local areas determined by the sensor locations. To minimize the maximum missed detection probability, the optimal sensor placement is efficiently solved for tree networks. Finally, a set of case studies is conducted using feeder data from the Pacific Northwest National Laboratories. It is shown that 10% mean detection error probability can be achieved by a sensor density of 30% for a typical feeder with the proposed optimal sensor placement and outage detection methods.**

I. Introduction

Outage detection and management is a long-standing problem in power distribution networks. Outages are caused by protective devices closing off a part of the network to automatically isolate faults. Usually, a short circuit fault will trigger this protective operation. The term *outage detection* is the task of finding the status of the protective devices, while *fault detection* is finding the faults that caused the resulting outage situation. Outage detection is often performed prior to fault detection and can greatly improve the accuracy of fault diagnosis.

Many methods for outage and fault detection are based on artificial intelligence methods. Fuzzy set approaches have been proposed based on customer calls and human inspection [2], and based on real-time measurement with a single sensor at the substation [3]. In networks where supervisory control and data acquisition (SCADA) systems are available, a subset of the protective devices' status can be obtained via direct monitoring. When two-way communications from the operator and the smart meters are available, AMI polling has been proposed to enhance outage detection [4]. There have also been knowledge-based systems that combine different kinds of information (customer calls, SCADA, AMI polling) [5]. Outage detection has also been extensively studied for transmission systems [6], [7], [8], [9], [10].

For fault detection, using only a single digital transient recording device at the substation, fault location and diagnosis systems have been developed based on fault distance computation using impedance information in the distribution system [11]. Using only the outage detection results, i.e., the status of the protective relays, expert systems have been applied to locate the underlying faults [12]. Incorporating voltage measurements in the distribution system with the outage detection results, fault detection methods based on knowledge based systems have been proposed [13]. Fault detection that uses fault voltage-sag measurements and matching has been proposed in [14], [15]. Fault diagnosis based on fuzzy systems and neural networks have also been proposed that can resolve multiple fault detection decisions [16].

A major alternative to these mechanisms is the so called last gasp, where area's in outage will notify via a distress signal that they are out of power. Last gasp provides a duplicate method of outage detection which can be combined with the proposed methods here. Combining both of these methods can further reduce the time to outage localization and recovery in practical scenarios.

This work proposes an outage detection scheme combining *real-time line sensing* from a limited number of sensors and *infrequent load updates* from AMI. Comparisons of real time line flows to expected load forecasts reveal changes in the network. The proposed framework can be implemented in practice since there is a growing number of deployments of distribution system line measurements [17], which can measure line current with high precision, and a growth of smart meter deployments in distribution systems.

II. Problem Formulation and Main Contributions

We consider a power distribution network that has a tree structure. Power is supplied from the feeder at the root, and is drawn by all the downstream loads. An outage is a protective device isolating a faulted area. When this occurs, the loads downstream of the faulted area will be in outage. The optimal design and performance of automatic outage detection is investigated with the use of the following two types of measurements:

Noisy Nodal Loads typically in the form of forecasts which have forecast errors that must be taken into account.

Error Free Edge Flows which typically come from real-time SCADA measurements or line sensors of the power flows on a fraction of the lines.

Edge flow measurements are modeled as error free for the following reason: measurements from SCADA systems are communicated in real time, whose errors are negligible compared with the errors of load forecast based on non-real-time information.

In this work, an edge flow refers to the real power flow flowing on a line, and we make an assumption of an lossless power distribution system. We note that the developed methodology is also applicable to the case where an edge flow refers to the current flow on a line, and no lossless assumption is needed since Kirchhoff's law always holds. The main contributions of this work are the following:

This research was supported in part by the DTRA under Grant HDTRA1-08-1-0010, in part by the Tomkat Center and in part by a Powell Foundation Fellowship, and in part by the National Science Foundation award Grants CIF1116377, CMMI-1435778 and ECCS-1549881. The results from this work were presented in part at the IEEE Power and Energy Society General Meeting, Vancouver, Canada, 2013 [1].

R. Sevlian is with the Dept. of Electrical Engineering, Stanford University, Stanford, CA, 94305 USA, (e-mail: rsevlian@stanford.edu).

Y. Zhao is with the Dept. of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, 11794 USA, (e-mail: yue.zhao.2@stonybrook.edu).

R. Rajagopal is with the Dept. of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: ramr@stanford.edu).

A. Goldsmith is with the Dept. of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: andrea@stanford.edu).

H. V. Poor is with the Dept. of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

The first two authors contributed equally to this work.

Outage Detection: For the ML hypothesis test for detecting any number of possible outages, an optimal and computationally efficient decentralized detector is developed.

Sensor Placement: The decentralized nature of the optimal detector is used to provide an efficient optimal sensor placement algorithm that minimizes the maximum missed detection error over all outage hypotheses.

III. System Model and Notation

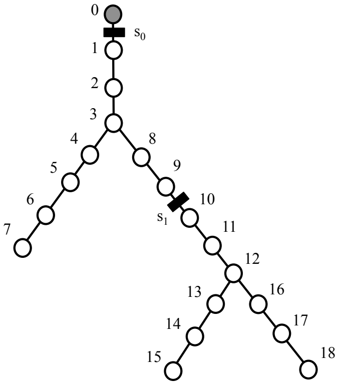


Fig. 1. Example tree $\mathcal{T}_1$ is used to illustrate various properties. Each node in the network is numbered. Node v_n is connected to it's parent via edge e_n consuming $x(v)$ power at each node. Two flow measurement sensors s_0, s_1 are deployed.

Topology of the Distribution System: The vertices in the distribution network are indexed by $V = \{v_0, v_1, \ldots, v_N\}$, with bus v_0 denoting the root of the tree. Edge e_n connects bus v_n and its parent node.

Outage Hypothesis Model: Outages are modeled as disconnected edges corresponding to protective devices disconnecting loads on a network. Consider single line outages in a tree with N edges: there will exist N single edge outage hypotheses and a non-outage situation. Let $\mathcal{H}^1 = \{e_1, \ldots, e_N, \emptyset\}$ be the set of all the single-line outage hypotheses for a tree $\mathcal{T}$.

In general, we consider an unknown number of possible outages. We define the set of up to k edge outages by $\mathcal{H}^k$.

Load Model: Each node v in the graph has a consumption load $x(v)$. The forecast of each load is $\hat{x}(v)$ with error $\epsilon(v) = x(v) - \hat{x}(v)$. We assume that errors are mutually independent random variables that follow $\epsilon(v) \sim N(0, \sigma(v)^2)$. Given load forecasts, the true load is an unknown, and modeled as a random variable distributed as $x(v) \sim N(\hat{x}(v), \sigma^2(v))$. In the vector case,

$$\hat{\mathbf{x}} \sim N(\mathbf{x}, \Sigma), \tag{1}$$

where Σ is a diagonal covariance matrix.

Measurement Model: For any edge e, s is the power flow on it towards all active downstream loads. The measured flow depends on the network topology, outage situation and the true loads. The sensor placement is denoted as $\mathcal{M}$ with $\mathcal{M} \subset E$. The vector of all measurements is $\mathbf{s} \in R^{|\mathcal{M}|}$.

The measured power consumption of the i^{th} sensor measurement under hypothesis $H \in \mathcal{H}^k$ is

$$s_i(H) = \sum_{v \in V_i(H)} x(v), \tag{2}$$

where the set $V_i(H)$ indicates the set of vertices to be summed over for this particular hypothesis, i.e., the loads that remain to be served via that line. In general, given a particular hypothesis H, the observed flows $\mathbf{s}$ can be expressed as

$$\mathbf{s} = \Gamma_H \mathbf{x}, \tag{3}$$

where $\Gamma_H \in \{0,1\}^{|\mathcal{M}| \times |V|}$ depends on the outage hypothesis. The load forecast error covariance Σ (i.e., the covariance of $\mathbf{x}$) can be estimated from the load forecasting process.

IV. Optimal Outage Detection

Given the vector of load forecasts, $\hat{\mathbf{x}}$, forecast error covariance Σ and real time load flow measurements $\mathbf{s}$ along a set of edges, the general outage detector must determine the correct number and location of each edge in outage. Note that these are single snapshot values of load forecast and line flow. The framework presented here can be extended to the multi-period case, but that is outside of the scope of this work.

A simple outage detector relying on ML detection is considered. Given the forecast model in eq. (1) and the measurement model in eq. (3), the ML detector is the following:

$$\hat{H} \in \underset{H \in \mathcal{H}^k}{\arg\max} \Pr\left(\mathbf{s} \mid \hat{\mathbf{x}},\ H\right) \tag{4}$$

The flow likelihood can be computed as:

$$\begin{aligned} \mathbf{s}|\{\hat{\mathbf{x}},\ H\} &= \Gamma_H \mathbf{x} \\ &= \Gamma_H(\hat{\mathbf{x}} + \epsilon) \\ &\sim N\left(\Gamma_H \hat{\mathbf{x}}, \Gamma_H \Sigma \Gamma_H^T\right) \quad \forall H \in \mathcal{H}^k. \end{aligned} \tag{5}$$

Eq. (5) evaluates a likelihood under each possible hypothesis. Therefore, a naive detector will enumerate every possible outage, evaluate it's likelihood, and choose the maximum. This is difficult for the following reasons:

1) Enumerating the entire ML detector requires $\sum_{k=1}^{K} \binom{|E|}{k}$ evaluations, given some maximum possible outage size K.
2) Many of the potential hypotheses map to the same observed flows, therefore the detector output is not unique. This occurs when one outaged edge is a descendant of another.
3) Missed detection errors in a multivariate hypothesis testing framework can only be evaluated via Monte Carlo testing. This unfortunately offers no insight in optimizing sensor placement.

V. Decoupled Maximum Likelihood Detection

In this section, we show that the issues regarding the general maximum likelihood detector can be overcome by decoupling the hypotheses and the observations, given the tree structure of the topology in the outage detection problem. This leads to a simple decentralized detector that is equivalent to eq. (4), where detection decisions are made by local decoupled scalar hypothesis tests, which when combined represent the optimal ML detection decision.

For the remainder of this section, we will demonstrate:

1) The original search space $\mathcal{H}^k$ can be replaced by a set $\mathcal{H}_u$ of uniquely detectable outages, due to the tree structure of the network.
2) Processing zero/positive flow information reduces the search space from $\mathcal{H}_u$ to $\mathcal{H}_u^+$. which decouples into a product set of local hypotheses: $\mathcal{H}_u^+ = \bigwedge_{A \in \mathcal{A}^+} \mathcal{H}_u^+(A)$, where A indicates a local area with its own outage hypotheses.
3) The joint likelihood function $\Pr(\mathbf{s} \mid \hat{\mathbf{x}},\ H)$ decouples across each area leading to a decentralized scalar value ML detector that can be solved efficiently.

A. Unique Outages

Finding outage hypotheses that give the ML of observations can lead to non-unique solutions. Without loss of generality (WLOG), uniquely detectable outages are considered, where no possible edge-outage event is downstream of any other. Define the set of all "unique" outage hypotheses as:

$$\begin{aligned}\mathcal{H}_u =& \{H \in \mathcal{H}^k \text{ for some } k, \text{ s.t no two edges} \\ & \text{are descendant of each other.}\}\end{aligned} \tag{6}$$

Consider tree $\mathcal{T}_2$ shown in Figure 2(c). Here $\mathcal{H}_u$ can be enumerated by simple observation

$$\mathcal{H}_u = \{\emptyset,\ e_1,\ e_2,\ e_3,\ e_4,\ e_5,\ (e_3 \times e_5),\ (e_4 \times e_5)\}. \tag{7}$$

There is a single non-outage hypothesis $\emptyset$, 5 single outage hypotheses, and 2 double outage hypotheses.

Next, we show that outage hypotheses in (7) for $\mathcal{T}_2$ and more generally for any arbitrary tree can be computed recursively. The set $\mathcal{H}_u$ can be enumerated using a "branch-network" based on the original tree. For example, Tree $\mathcal{T}_1$ in Figure 1 is depicted in Figure 2(a) with nodes and edges removed, which highlights the various branches of the graph. Each set of branches are aggregated as a node to be traversed in the hypothesis enumeration procedure, as in Figure 2(b).

Consider a set-valued function $E(b) = \{e \in E : e \text{ is along branch } b\}$ to enumerate the set of edges on a branch. Given the two examples, we have the following branch-edges:

- $\mathcal{T}_1$: $E(b_1) = \{e_1, e_2, e_3\}$, $E(b_2) = \{e_4, e_5, e_6, e_7\}$, $E(b_3) = \{e_8, e_9, e_{10}, e_{11}, e_{12}\}$, $E(b_4) = \{e_{13}, e_{14}, e_{15}\}$ and $E(b_5) = \{e_{16}, e_{17}, e_{18}\}$.
- $\mathcal{T}_2$: $E(b_1) = \{e_1, e_2\}$, $E(b_2) = \{e_3, e_4\}$, $E(b_3) = \{e_5\}$.

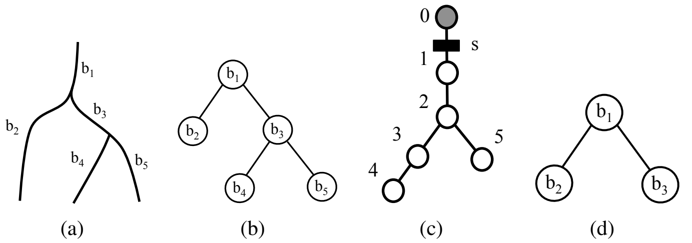


Fig. 2. 2(a) Branches of tree $\mathcal{T}_1$. 2(b) Branch network for tree $\mathcal{T}_1$. 2(c) Simple tree network $\mathcal{T}_2$. 2(d) Branch network for $\mathcal{T}_2$.

Using this definition, we provide the following recursive definition of the set $\mathcal{H}_u(b)$, which indicates the set of uniquely detectable hypotheses formed from branch b and all descendants. From this definition, it is clear that $\mathcal{H}_u \equiv \mathcal{H}_u(b_1)$, since b_1 is the root branch of the tree

$$\mathcal{H}_u(b) = E(b) \cup \left(\bigcup_{\mathbf{b}\in\mathbb{P}(\mathbf{child}(b))} \left(\bigwedge_{b\in\mathbf{b}} \mathcal{H}_u(b) \right) \right). \tag{8}$$

Here b is the current node, and $\mathbf{child}(b)$ is the set of children of b in the branch tree. The set $\mathbb{P}(\mathbf{child}(b))$ is the power set of all the child branches, where any element of the power set is $\mathbf{b}$. Note that if a branch has no descendants, we merely evaluate $E(b)$, since the remaining terms are null. Eq. (8) is quite unwieldy, but can be interpreted easily.

For tree $\mathcal{T}_2$ in Figure 2(c) the child branches are $\mathbf{child}(b_1) = \{b_2, b_3\}$ while the power set is:

$$\mathbb{P}(\{b_2, b_3\}) = \{\underbrace{\{\emptyset\}}_{\mathbf{b}_0}, \underbrace{\{b_2\}}_{\mathbf{b}_1}, \underbrace{\{b_3\}}_{\mathbf{b}_2}, \underbrace{\{b_2, b_3\}}_{\mathbf{b}_3}\}. \tag{9}$$

Evaluating (8), we arrive at the following:

$$\begin{aligned}\mathcal{H}_u(b_1) = E(b_1) &\cup \left(\bigwedge_{b\in\{\emptyset\}} \mathcal{H}_u(b) \right) \cup \left(\bigwedge_{b\in\{b_2\}} \mathcal{H}_u(b) \right) \\ &\cup \left(\bigwedge_{b\in\{b_3\}} \mathcal{H}_u(b) \right) \cup \left(\bigwedge_{b\in\{b_2,b_3\}} \mathcal{H}_u(b) \right)\end{aligned} \tag{10}$$

With the following definitions:

D1 Non-outage case: $\bigwedge_{b\in\{\emptyset\}} \mathcal{H}_u(b) = \emptyset$.

D2 Hypotheses double counting: $e_i \cup e_i = e_i$, including the empty set $\emptyset \cup \emptyset = \emptyset$.

D3 Cross product reduction: $\emptyset \times e_i = e_i$, which implies $E(b) \times \emptyset = E(b)$.

In the case of $\mathcal{T}_2$, eq. (8) leads to

$$\mathcal{H}_u(b_1) = \{\emptyset \cup E(b_1) \cup \mathcal{H}_u(b_2) \cup \mathcal{H}_u(b_3) \cup (\mathcal{H}_u(b_2) \times \mathcal{H}_u(b_3))\}. \tag{11}$$

For the branches b_2 and b_3 we use eq. (8) and D1 resulting in $\mathcal{H}_u(b_2) = \{E(b_2) \cup \emptyset\}$ and $\mathcal{H}_u(b_3) = \{E(b_3) \cup \emptyset\}$. Using D2 and D3, we have the following:

$$\begin{aligned}\mathcal{H}_u(b_1) &= \{E(b_1) \cup \{\emptyset\} \cup \{E(b_2) \cup \emptyset\} \cup \{E(b_3) \cup \emptyset\} \\ &\quad \cup \{E(b_2) \cup \emptyset\} \times \{E(b_3) \cup \emptyset\}\} \\ &= \{\emptyset \cup E(b_1) \cup E(b_2) \cup E(b_3) \cup (E(b_2) \times E(b_3))\} \\ &= \{\emptyset,\ e_1,\ e_2,\ e_3,\ e_4,\ e_5,\ (e_3 \times e_5),\ (e_4 \times e_5)\}.\end{aligned}$$

This is identical to the simple enumeration in (7). With the use of a unique hypothesis set, the maximum likelihood detector is reduced to

$$\hat{H} = \underset{H\in\mathcal{H}_u}{\arg\max} \Pr(\mathbf{s} \mid \hat{\mathbf{x}},\ H). \tag{12}$$

Note the equality in the maximization since the restriction of the search space allows us to find a unique solution. Notice that this procedure will automatically enumerate all possible number of outages as well as their locations in the graph. Under the general model of any number of edge outages, this is the complete enumeration of the hypothesis set which still has a high computational burden.

The remaining sections show that a) this search space decouples to a product space of local hypotheses given binary flow indicators and b) the likelihood function decouples along these local "areas", thereby reducing to a decoupled scalar hypothesis test for each local "area".

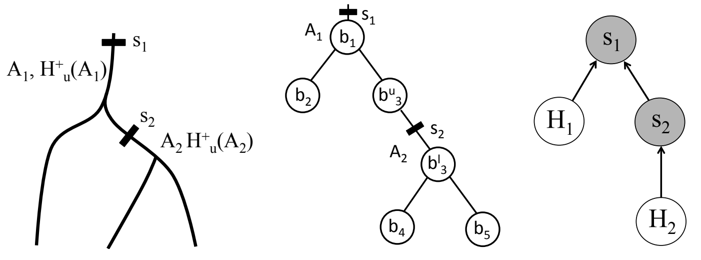
B. Decoupling the Hypotheses Set with Binary Flow Information

The detection problem will encounter a set of positive ($\mathbf{s}_+$) and zero ($\mathbf{s}_z$) flow observations depending on whether a sensor is downstream of an edge outage. The ML detector reduces as:

$$\begin{aligned}\hat{H} &= \underset{\forall H\in\mathcal{H}_u}{\arg\max} \Pr(\{\mathbf{s}_z\ \mathbf{s}_+\} \mid \hat{\mathbf{x}},\ H) && (13) \\ &= \underset{\forall H\in\mathcal{H}_u^+}{\arg\max} \Pr(\{\mathbf{s}_+\} \mid \hat{\mathbf{x}},\ H), && (14)\end{aligned}$$

where, by processing the flow information, the set of hypotheses in the detector search space reduces from $\mathcal{H}_u$ to $\mathcal{H}_u^+$. The reduced hypothesis set is defined as:

$$\begin{aligned}\mathcal{H}_u^+ =& \{H \in \mathcal{H}_u \text{ where the outages do not violate the observed} \\ & \text{positive and zero flow observations}\}.\end{aligned} \tag{15}$$



(a) (b) (c)

Fig. 3. 3(a) The branch network for network $\mathcal{T}_1$ with associated sensors s_1, s_2 and area hypotheses $\mathcal{H}_u^+(A_1)$ and $\mathcal{H}_u^+(A_2)$. 3(b) Each area will keep the local branches. Branch b_3 is split between two areas and processed as different vertices in branch-network (b_3^u and b_3^l. 3(c) The network can be modeled by a directed graphical model indicating observations (shaded) and variables.

More importantly, $\mathcal{H}_u^+$ does not require the full recursive enumeration, instead it decouples as a product space of local recursively enumerated sets. The set $\mathcal{H}_u^+(b)$ follows the similar definition of $\mathcal{H}_u(b)$, indicating all unique outages that do not violate the power and zero flow observations formed from branch b and its descendants.

This is first shown in an example, then the general form is stated. Consider $\mathcal{T}_1$, with the branch network with flow measurements as shown in Figure 3(b). In this example, all branches are unchanged except b_3. Branch b_3 is split into b_3^u and b_3^l separated by flow measurement s_1. In the case of splitting branch nodes, the upper branch will take the edge with the measurements, and hence $E(b_3^u) = \{e_8, e_9, e_{10}\}$ and $E(b_3^l) = \{e_{11}, e_{12}\}$.

The following illustrative example highlights a general decoupling principle of the hypothesis set $\mathcal{H}_u^+$. Consider the two cases separately ($s_1, s_2 > 0$ and $s_1 > 0, s_2 = 0$) which provides the intuition for the general case.

1) Case 1: ($s_1 > 0$, $s_2 > 0$) This implies that *there cannot be any outage with edges in* b_1 *or* b_3^u, else $s_2 = 0$. A brute force enumeration of $\mathcal{H}_u^+(b_1)$ is done by enumerating $\mathcal{H}(b_1)$ according to Figure 2(b), then removing any terms with $E(b_1)$ and $E(b_3^u)$ outages. The set of unique outages is:

$$\begin{aligned}\mathcal{H}_u(b_1) = \{&\emptyset \cup E(b_1) \cup E(b_2) \cup (E(b_3^u) \cup \mathcal{H}_u(b_3^l)) \\ &\cup \big(E(b_2) \times (E(b_3^u) \cup \mathcal{H}_u(b_3^l))\big)\}.\end{aligned}$$

Removing the possible outages due to the binary flow indicators, (i.e. any tuple with edges in $E(b_3^u)$, or $E(b_1)$) results in

$$\begin{aligned}\mathcal{H}_u^+(b_1) &= \{\emptyset \cup E(b_2) \cup \mathcal{H}_u^+(b_3^l) \cup \big(E(b_2) \times \mathcal{H}_u^+(b_3^l)\big)\} \\ &= \{\emptyset_1 \cup E(b_2)\} \times \{\emptyset_2 \cup \mathcal{H}_u^+(b_3^l)\} \\ &= \mathcal{H}_u^+(A_1, f_1) \times \mathcal{H}_u^+(A_2, f_2),\end{aligned}$$

where the notations with A_i, f_i are explained below. Local areas A_1 and A_2 are used to partition the original tree. Each area will contain a root sensor (ex. A_1 has root sensor s_1) and a set of child sensors which are sensors immediately downstream of the root sensor. For example, $\textbf{child}(s_1) = \{s_2\}$. Within a local area, a set of unique outage hypotheses are evaluated $\mathcal{H}_u^+(A_1, f_1)$ satisfying the child binary flow indicators $f_i = \mathbb{I}_{\{\textbf{child}(s_i)>0\}}$. This will decouple the set of all hypotheses into a product space of "area hypotheses" $\mathcal{H}_u^+(A_1, f_1)$ and $\mathcal{H}_u^+(A_2, f_2)$, where $\mathcal{H}_u^+(b_1) = \mathcal{H}_u^+(A_1, f_1) \times \mathcal{H}_u^+(A_2, f_2)$. Note that, given the binary flow indicators, the search space of hypotheses no longer needs a full recursive enumeration, but can be computed as a product of smaller hypothesis sets. Finally, note that $\emptyset_1$ and $\emptyset_2$ are non-outage conditions within the local areas. This is discussed in more detail in Appendix C.

2) Case 2: ($s_1 > 0$, $s_2 = 0$) This implies that *all possible outage scenarios must contain* b_1 *or* b_3^u, else $s_2 > 0$. First we enumerate $\mathcal{H}_u(b_1)$ then keep only those elements which lead to $s_2 = 0$ (i.e. every tuple with edges in $E(b_3^u)$, or $E(b_1)$) , we have:

$$\begin{aligned}\mathcal{H}_u^+(b_1) &= \{E(b_1) \cup E(b_3^u) \cup E(b_2) \times E(b_3^u)\} \\ &= \{\mathcal{H}_u^+(A_1, f_1)\}\end{aligned}$$

Note that in this case, $\mathcal{H}_u^+(A_2, f_2)$ is never evaluated since $s_2 = 0$.

3) General Hypothesis Decoupling: For any arbitrary tree and set of binary flow indicators the unique hypothesis set conditional on the binary line flow indicators, $\mathcal{H}_u^+$, will decouple according to:

$$\mathcal{H}_u^+ = \bigwedge_{A \in \mathcal{A}^+} \mathcal{H}_u^+(A, f). \tag{16}$$

The set $\mathcal{A}^+$ indicates the areas which have root measurements $s_i > 0$ and $\mathcal{H}_u^+(A, f)$ is the local conditional hypothesis.

Any given $H_i \in \mathcal{H}_u^+$ is represented by a product of area hypotheses $H_i = H_{1,i(1)} \times \ldots \times H_{M,i(M)}$, where for the k^{th} area $H_{k,i(k)} \in \mathcal{H}_u^+(A_k, f_k)$. Index $i(k)$ is the particular index into the k^{th} hypothesis set restricted to area A_k. Appendix D presents a general algorithm to enumerate $\mathcal{H}_u^+(A, f)$, for arbitrary binary flow information.

C. Decoupling the Joint Likelihood:

We show that the joint likelihood of all observations decouple across the areas. In general:

$$\Pr(\mathbf{s}_+ \mid \hat{\mathbf{x}},\ H) = \prod_{i: A_i \in \mathcal{A}^+} \Pr(s_i \mid \textbf{child}(s_i),\ \hat{\mathbf{x}},\ H_i) \tag{17}$$

The tree $\mathcal{T}_1$ is presented here as an example, while the general case can be easily shown as an extension of the example.

Consider again the sub-graph in Figure 3(a), where hypotheses are reduced using binary flow information (both $s_0, s_1 > 0$).

As discussed in Section V-B3, a unique hypothesis $H_i \in \mathcal{H}_u^+$ can be represented as $H_i = H_{1,i(1)} \times H_{2,i(2)}$. Where $H_{1,i(1)} \in \mathcal{H}(A_1)$ and $H_{2,i(2)} \in \mathcal{H}(A_2)$.

From eq. (2), the observed flow s_1 can be computed as follows:

$$s_1 | \{\hat{\mathbf{x}},\ H_i\} = \sum_{v \in V(H_{1,i(1)} \times H_{2,i(2)})} x(v) \tag{18}$$

$$= \sum_{v \in V(H_{1,i(1)} \times H_{2,\emptyset}) \setminus V(H_{2,\emptyset})} x(v) + \sum_{v \in V(H_{2,i(2)})} x(v) \tag{19}$$

$$= \sum_{v \in V(H_{1,i(1)})} x(v) + s_2 \tag{20}$$

where:

- Eq. (18) is the summation of true loads. By decoupling of the hypotheses across areas, we represent H_i as the product of the two local hypotheses $H_{1,i(1)}$ and $H_{2,i(2)}$.
- In (19), this can be separated as the sum of parts. (1) Term $V(H_{1,i(1)} \times H_{2,\emptyset}) \setminus V(H_{2,\emptyset})$ are the vertices in the summation in A_1 independent of what is happening of downstream areas. $H_{2,\emptyset}$ indicates the non-outage hypothesis in area 2. (2) Term $V(H_{2,i(2)})$ is the set of vertices in the summation for area 2. Even though $H_{2,i(2)}$ is unknown, $s_2 = \sum_{v \in V(H_{2,i(2)})} x(v)$.

- Eq. (20), replaces the second summation of eq. (19), with the observed flow, since they are the same. Therefore, if we condition on the remaining flow observation, **child**(s_1), the likelihood function for observation s_1 only depends on the local hypotheses and on no other area hypotheses.

Finally, since $x(v)$ is not known, a likelihood function for *the net flow in the area* can be constructed given the load forecasts. The first term in eq. (20) is modeled with the following:

$$s_1 - s_2 | \{s_2,\ \hat{\mathbf{x}},\ H\} \sim N(\mu_1(\hat{\mathbf{x}}, H), \sigma_1(\hat{\mathbf{x}}, H)). \tag{21}$$

Evaluating $\mu_1(\hat{\mathbf{x}}, H)$, and $\sigma_1(\Sigma, H)$ can be computed easily, since the load forecasts, uncertainties and local hypotheses can all be enumerated. Section VI-B, described this methodology in more detail. Similarly, s_2 is decoupled from any hypothesis in $\mathcal{H}_u^+(A_1)$, since the measurement depends only on downstream hypotheses (assuming s_2 has some positive flow to begin with).

In this example the decoupled likelihood function is

$$\Pr(s_1\ s_2 |\ \hat{\mathbf{x}},\ H_1,\ H_2) = \Pr(s_2 |\ \hat{\mathbf{x}},\ H_2) \Pr(s_1 | s_2,\ \hat{\mathbf{x}},\ H_1)$$

where $H_1 \in \mathcal{H}_u^+(A_1)$ and $H_2 \in \mathcal{H}_u^+(A_2)$.

This decoupling can be represented as a simple graphical model as shown in Figure 3(c), where each local hypothesis is an unknown variable that must be determined via likelihood maximization. Conditioning on observations s_1, s_2, the likelihood function will be a product of two terms that depend on H_1 and H_2 separately. This decoupling allows for maximization of the product to be done independently for each variable. This graphical model formulation also extends to the general case where there may be noise in the flow measurements. In such a case, the decoupling would not work, but a message passing algorithm can be used. The graphical model formulation can also be applied in different sensor types.

In the general case, eq. (18)-(20) can be extended to a general area network with multiple downstream sensors.

D. Decoupled Maximum Likelihood Function:

Combining the results so far leads to the decoupled likelihood function and decoupled ML detector as follows:

$$\begin{aligned}
\hat{H} &\in \operatorname*{arg\,max}_{\forall H \in \mathcal{H}^k} \Pr(\mathbf{s} | \hat{\mathbf{x}},\ H) && (22)\\
&= \operatorname*{arg\,max}_{\forall H \in \mathcal{H}_u} \Pr(\{\mathbf{s}_+\ \mathbf{s}_z\} | \hat{\mathbf{x}},\ H) && (23)\\
&= \operatorname*{arg\,max}_{\forall H \in \mathcal{H}_u^+} \Pr(\mathbf{s}_+ | \hat{\mathbf{x}},\ H) && (24)\\
&= \operatorname*{arg\,max}_{\forall A_i \in \mathcal{A}^+\ \forall H_i \in \mathcal{H}^+(A_i)} \Pr(\mathbf{s}_+ | \hat{\mathbf{x}},\ H_1 \dots H_M) && (25)\\
&= \operatorname*{arg\,max}_{\forall A_i \in \mathcal{A}^+\ \forall H_i \in \mathcal{H}^+(A_i)} \prod_{i : A_i \in \mathcal{A}^+} \Pr(s_i | \textbf{child}(s_i), \hat{\mathbf{x}},\ H_i) && (26)\\
&= \bigwedge_{i : A_i \in \mathcal{A}^+} \operatorname*{arg\,max}_{\forall H_i \in \mathcal{H}^+(A_i)} \Pr(s_i | \textbf{child}(s_i), \hat{\mathbf{x}},\ H_i) && (27)
\end{aligned}$$

Here eq. (22) is the original hypothesis test over all possible outage hypotheses. This is reduced to a search space over $\mathcal{H}_u$ in eq. (23). This further reduces to an even smaller search space $\mathcal{H}_u^+$ due to processing of binary flow information in eq. (24). Eq. (25) decouples $\mathcal{H}_u^+$ to a product space of local search hypotheses. Eq. (26) decouples the likelihood functions by conditioning on the set of observations. This likelihood function is a product of terms which only depend on the local hypotheses. Therefore maximizing this product is equivalent to maximizing each term separately, as in eq. (27).

Algorithm 1: Maximum Likelihood Hypothesis Detector.

```
Result: Maximum Likelihood Hypothesis Detector
Input:  [1] Load Forecast/Nominal Statistics: $\hat{\mathbf{x}}$, $\Sigma$
        [2] Real Time Load: $\mathbf{s} = \{\mathbf{s}_+, \mathbf{s}_z\}$
$\mathcal{A}^+ \leftarrow$ prune-areas$(\mathcal{A}, \mathbf{s})$
for $A_i \in \mathcal{A}^+$ do
    // Generate Local Hypothesis set.
    $\mathcal{H}_u^+(A_i) \leftarrow$ local-hypotheses$(\mathcal{T}, s_i, \textbf{desc}(s_i))$
    // Local ML Detector
    $\hat{H}_i \leftarrow \operatorname*{arg\,max}_{\forall H_i \in \mathcal{H}^+(A_i)} \Pr(s_i | \textbf{child}(s_i), \hat{\mathbf{x}},\ H_i)$
end
// Combine Local Hypotheses
$\hat{H} \leftarrow \bigwedge_{A_i \in \mathcal{A}^+} \hat{H}_i$
```

The decoupling of the likelihood function in (27), leads to a simple decentralized detector in Algorithm 1. The input is (1) the set of load forecasts $\hat{\mathbf{x}}$ with their nominal statistics Σ, and (2) the real time measurements $\mathbf{s}$. The function **prune-areas** discards areas with zero flow measurements at their roots. The function **local-hypotheses** performs the generation of local hypothesis set $\mathcal{H}_u^+(A_i)$ as described in Appendix D.

Finally, the local ML detector is simple to evaluate as a multi-hypothesis test involving scalar Gaussian random variables of known means and variances as follows: For a local area, since $\mathcal{H}^+(A)$ is enumerated, we determine:

$$\hat{H} = \operatorname*{arg\,max}_{H \in \mathcal{H}^+(A)} \left(\frac{s_i - \sum_{s \in \textbf{child}(s_i)} s - \mu_i(\hat{\mathbf{x}}, H)}{\sigma_i(\Sigma, H)} \right)^2$$

Therefore each decision only depends on an effective measurement

$$\Delta s_i \triangleq s_i - \sum_{s \in \textbf{child}(s_i)} s \tag{28}$$

for each local area. Computation of the means and variance are discussed in Section VI-B.

VI. Sensor Placement Problem

This section investigates where to the place the line flow sensors for best outage detection performance. The metric used in evaluating the performance of the sensor placement is the maximum missed detection probability over all hypotheses. The missed detection probability for any hypothesis H, given any sensor placement $\mathcal{M}$, is $P_E(H, \mathcal{M}) = \Pr(\hat{H} \neq H; \text{placement } \mathcal{M})$, where $\hat{H}$ is the optimal solution of the outage detection problem as solved in Section V.

The optimal sensor placement, given a total number of sensors M, that minimizes the maximum missed detection probability is given by

$$\mathcal{M}^\star = \operatorname*{arg\,min}_{|\mathcal{M}| = M} \max_{H \in H_u} P_E(H, \mathcal{M}). \tag{OPT-1}$$

Note the objective is the missed detection probability maximized over all unique outage hypotheses H_u. This is difficult to solve outside of a combinatorial enumeration of sensor locations and hypotheses. A suitable proxy to optimize instead is:

$$\mathcal{M}^{\star} = \underset{|\mathcal{M}|=M}{\arg\min}\ \max_{A\in\mathcal{A}} P_E^{\max}(A), \qquad \text{(OPT-2)}$$

where $P_E^{max}(A) = \max_{f,H\in H_u^+(A,f)} P_E(H,\mathcal{M})$, which searches over the hypothesis in a local area A. This second optimization very closely approximates an upper bound to (OPT-1) (see Appendix F for details). We solve optimization OPT-2 via a bisection on P^{target} as follows:

$$\begin{aligned}\mathcal{M}^{\star} = &\underset{|\mathcal{M}|\le M}{\text{find}}\ \mathcal{M} && \text{(OPT-3)}\\ &\text{s.t. } P_E^{\max}(A) \le P^{target},\ \forall A\end{aligned}$$

So the minimum P^{target} is determined yielding a solution of size $|\mathcal{M}| = M$. We next show that this can be solved very efficiently, with a greedy placement algorithm.

A. Sensor Placement Algorithm

The problem (OPT-3) can be solved by a greedy algorithm the intuition of which is the following. Starting from the bottom of a tree, a temporary local area with root sensor in e_t is maintained. The root sensor is iteratively moved closer to the root edge e_1 of the entire tree, while maintaining that the maximum error of all areas below the root sensor is less than P^{target}. This is done by maintaining that the local area has error less than P^{target}. If this is true we move closer to the root, if not, we place a sensor and start a new sensor to move further up. Since the objective function, i.e., the maximum missed detection probability decouples across areas, we can maintain that the feasibility problem is always satisfied. Finally, if at the end the total number of sensors are less than M, $\mathcal{M}^{\star}$ is returned.

```
Algorithm 2: Solution to optimization (OPT-3) for tree network.
   Result: Placement for a Tree Network
   Input:  [1] Tree network $\mathcal{T}$
           [2] Nominal loads statistics $\hat{\mathbf{x}}$, $\Sigma$
           [3] Target error $P^{\text{target}}$
 1 // Generate node process order
 2 $E_{\text{process}} \leftarrow$ generate-edge-order($\mathcal{T}$)
 3 // initialize sensor placement as empty
 4 $\mathcal{M}^g \leftarrow \emptyset$
 5 for $e_t \in E_{process}$ do
 6     $A \leftarrow$ construct-area($e_t, \mathcal{M}^g$)
 7     // Evaluate the current local error metric
 8     if $P_E^{\max}(A) \le P^{target}$ then
 9         // continue to next node
10     else
11         if $|\mathbf{child(v_t)}| == \mathbf{1}$ then
12             $\mathcal{M}^g \leftarrow$ line-action($A, \mathcal{M}^g$)
13         else if $|\mathbf{child(v_t)}| > \mathbf{1}$ then
14             $\mathcal{M}^g \leftarrow$ tree-action($A, \mathcal{M}^g$)
15     end
16 end
17 return $\mathcal{M}^g$
```

This Algorithm is specified by Algorithm 2. The inputs to the method are the tree network $\mathcal{T}$ and the set of nominal load forecasts $\hat{\mathbf{x}}$ and forecast error covariance matrix Σ. To have the effect of starting at the leaf of the network and move our way up to the root, the algorithm will process a sequence of edges E_{process}. For example in Figure 1, $E_{\text{process}} = \{E(b_4)\ E(b_5)\ E(b_3)\ E(b_2)\ E(b_1)\}$. The list E_{process} is generated in line 2 with function **generate-edge-order**. Specifically, this function takes the tree $\mathcal{T}$ and traverses via breadth first search keeping track of the depth of each vertex/edge. Reversing this list of depths yields a list of nodes to process, the parent edge being $e \in E_{\text{process}}$.

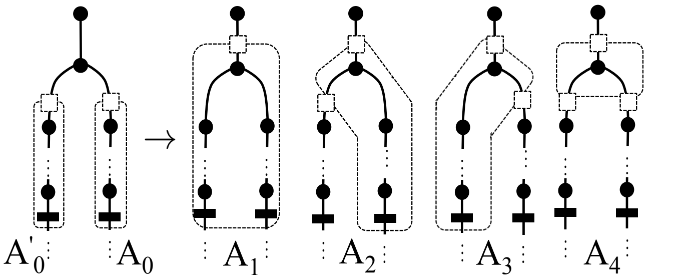


Fig. 4. (**greedy-tree-action**) Correct node traversal in the tree network assumes that $P_E^{\max}(A_0) < P^{\text{target}}$ and $P_E^{\max}(A'_0) < P^{\text{target}}$. If $P_E^{\max}(A_1) < P^{\text{target}}$ we do nothing else, generate and evaluate the error one of the $2^{|\mathbf{child(v_t)}|}$ area networks that can be constructed. In the case of greedy action, the area with the smallest number of sensors along with smallest $P_E^{\max}(A)$, where $P_E^{\max}(A) < P^{\text{target}}$ is chosen.

The greedy solution $\mathcal{M}^g$ must first be initialized as empty. In line 5 we iterate over the current root node e_t and current sensor placement $\mathcal{M}$. In line 6 we construct the current area network A. We then evaluate $P_E^{\max}(A)$ in line 8; If $P_E^{\max}(A) > P^{\text{target}}$ a placement action **line-action** or **tree-action** is performed depending on the number of child nodes of v_t (downstream of e_t). Each sub function is described in more detail below.

1) **construct-area***:* For each iteration, the temporary node e_t is visited and the area network A is constructed with e_t and the previous solution $\mathcal{M}^g$. The current edge e_t is the temporary root sensors of the area s_t. The terminal sensors of the area are any sensors in $\mathcal{M}^g$ that are children of s_t. Note that this may be empty at the start of the algorithm.

2) **line-action***:* Given that the current subproblem satisfies $P_E^{\max}(A) < P^{\text{target}}$, if the next subproblem does not satisfy the condition, the only option is to place a sensor at **child** (v_t). So $\mathcal{M}^g \leftarrow \mathcal{M}^g \cup e_t$.

3) **tree-action***:* Given that the algorithm up to now has placed sensors on the two disjoint trees with roots with v_1 and v_2. The current root node must move as far up to v_0 before placing a sensor to satisfy the error constraint. This leads to two types of actions: a greedy strategy that is easy to implement and the optimal strategy. We propose to implement the greedy strategy in practice which is almost always equal to the optimal strategy. The details are discussed in Appendix H.

Specifically, we illustrate the greedy strategy as follows. In Figure 4, first assume $P_E^{\max}(A_0)$ and $P_E^{\max}(A'_0) < P^{\text{target}}$ and $P_E^{\max}(A_1) > P^{\text{target}}$. In moving e_t closer to the root, the algorithm must choose either A_2, A_3, or A_4: a) if either A_2 or A_3 ensures that P^{target} is satisfied, we choose the one with the lower error, and b) if A_2 and A_3 both cannot satisfy P^{target}, we choose A_4 which surely satisfies P^{target}. In the general case, given a node with K children, all 2^K sensor options are evaluated, and the configuration of the fewest sensors that satisfies the error constraint and has the lowest error probability is selected.

B. Evaluating $P_E^{\max}(A_i)$

The objective $P_E^{\max}(A_i)$ can be evaluated straightforwardly. For each local hypothesis $H \in \mathcal{H}^+(A_i, f_i)$, the conditional distribution $\Delta s_i|\{\ \hat{\mathbf{x}},\ H\} \sim N(\mu(\hat{x}, H), \Sigma(\hat{x}, H))$ is computed. In particular, $\mu(\hat{x}, H)$ and $\Sigma(\hat{x}, H))$ are the sum of the forecasts and error variances of all the *remaining loads inside area* A_i given outage H, respectively. Finally, given the *scalar* Gaussian distributions of $\Delta s_i|\{\hat{\mathbf{x}}\ H\}$ for different H, the probability of missed detection for a scalar ML detector can be computed easily for this area A_i.

C. Optimality and Complexity of Algorithm 2

This discussion concerns only **optimal-tree-action** since it guarantees optimality, although the greedy strategy's output placement is almost always identical. We have the following theorem.

Theorem 1. *The bottom up placement solution $\mathcal{M}^g$ relying on* **optimal-tree-action** *traversal solves OPT-3.*

Appendix E discusses the proof of optimality, as well as the complexity of the optimal detector, evaluating the objective function $P_E^{max}(A)$, and the sensor placement algorithm. The results are summarized as follows for a worst case tree of depth D and K children at each vertex shown in Figure 12(b).

> **ML Detection**: Evaluating the detector for an area and computing the missed detection P_E^{max} depends on the size of $\mathcal{H}_u$. For an area of size $|E|$ the hypothesis set is of size $O(2^{|E|})$. Evaluating only outages of size k has $O(|E|^k)$ complexity, and evaluating a restricted MAP detector with only single outages has $O(|E|)$ complexity.
>
> **Sensor Placement**: If the cost of evaluating $P_E^{max}(A) = T$, the greedy placement is of $O(|E|\ T)$ complexity, while the optimal placement is $O(|E|^K\ T)$ for the general tree.

The optimal detector and placement complexity in the worst case is quite high. In practice, however, this cost is averted using a detector with a fixed hypothesis size and having areas of small sizes. Additionally, having multiple downstream measurements will cut the area hypotheses by a factor of $2^{|\textbf{child}(s_i)|}$ (See Appendix E for details).

VII. Numerical Study

A. PNNL Cases and Modeling

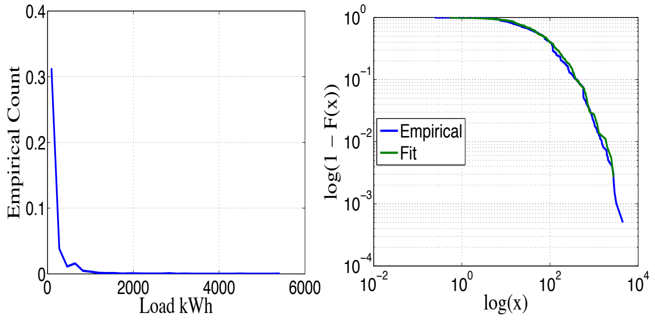
Outage detection is performed using a subset of the Pacific Northwest National Laboratory test feeders [18]. Table I gives an overview of the feeders chosen for the simulation study. The primary applications of the feeders are heavy to light urban networks, as well as suburban and rural networks. The climate zones refer to (1) temperate (2) hot/arid (3) cold (4) hot/cold (5) hot/humid according to [18].

TABLE I
PNNL Test feeders used in case study

Network	Voltage	Climate Zone	Type	Nominal Size	Reduced Size
R1-12.47-1	12.5 kV	1	suburban	613	61
R2-12.47-3	12.47 kV	2	urban	52	42
R5-12.47-1	13.8 kV	5	urban	265	149
R5-12.47-4	12.47 kV	5	commercial	643	111
R5-25.00-1	22.9 kV	5	suburban	946	76

1) Outage Model and Equivalent Reduced Networks: Outages are modeled by *fuses* and *switches* disconnecting the downstream loads from the substation. These devices represent the edges of the tree, on which the subsequent analysis is performed. The set of loads which are disconnected by a fuse or switch disconnecting are lumped to aggregate loads.

Table I indicates the nominal size of the networks, in terms of customer size per tree, along with the reduced size. The reduced sizes indicate the number of vertices of the tree with which the analysis is based on.



(a)

(b)

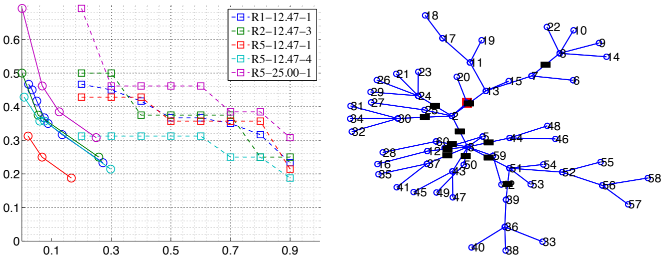
Fig. 5. An individual load is the grouping of all loads that can be disconnected from the feeder root using either switches or fuses. The empirical distribution is long tailed well fit with a generalized Pareto model. (a) Empirical histogram of all loads collected from 23 feeders. (b) Log survival function for data as well as randomly generated data using fit shows close agreement.

We analyze all 23 feeders and perform aggregate analysis based on the combined load information. An empirical histogram of the load values is given in Figure 5(a). In general, visual procedures such as this are the more useful way of diagnosing the particular nature of the data, before any fitting is done [19]. To validate the heavy tail distribution we consider a log survival function of both the load data and randomly sampled distribution following the fit parameters. The comparison of the two are plotted in Figure 5(b).

In particular, we model the loads at each node in the network with a generalized Pareto distribution: $f(x; \kappa, \sigma, \theta) = \left(\frac{1}{\sigma}\right)\left(1 + \kappa\frac{x-\theta}{\sigma}\right)^{-(\kappa-1)/\kappa}$. Using the aggregated loads from all the considered feeders, the following maximum likelihood estimates and 95% confidence intervals are generated for the model: $\theta = 0.25$, $\kappa = 0.58(0.51\ 0.65)$ and $\sigma = 74.28(68.53\ 80.50)$. This model is used in generating random test cases in Section VII-C, and used to understand the sensor placement performance.

2) Forecast Error Model: In the case of a residential feeder, the number of customers per vertex can vary, leading to different forecasting errors. In [20] the authors present a rule of thumb model for day ahead load forecasting at various aggregation levels based on smart meter data. The day ahead forecast coefficient of variation, $\kappa = \sigma/\mu$ is shown to be dependent on the mean load of the aggregate. Many studies make simplified assumptions on the relative forecast error. However, at the level of small aggregates, the forecast κ can vary greatly on the size of the aggregate and must be taken into account.

In [20] the following is proposed: $\text{CV}(W) = \mathbf{E}[\text{CV}(x_A, \hat{x}_A)\,|W_A = W] = \sqrt{\frac{\alpha_0}{W^p} + \alpha_1}(\%)$, where W_A is the mean load of aggregate A. A reasonable fit is $\kappa(W) = \sqrt{\frac{3562}{W} + 41.9}$.This formula is used to show that each set of islanded loads will have a different value of $\kappa(W)$.



(a) (b)

Fig. 6. (a) Outage detection performance for selected PNNL feeders. Square marker denotes specified error targets for optimization. Circular marker denotes empirical mean missed detection errors. (b) Feeder R1-12.47-1 with placement corresponding to a mean error of 24%.

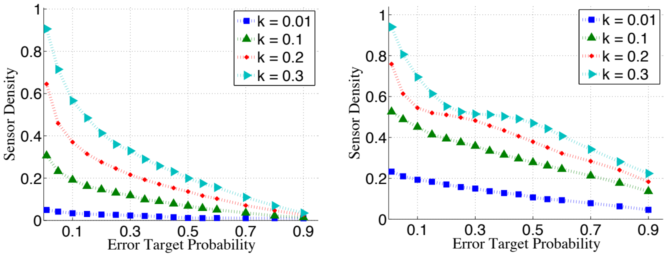
B. Sensor Placement

Algorithm 2 is evaluated for the reduced networks based on the PNNL systems. The sensor placement was computed by using a set of a max-error-targets and computing the optimal placement associated with them. For each placement, the numerical max and mean missed detection error are computed as well. Figure 6(a) shows the performance with the developed sensor placement algorithm. Even though each network represents a different application, they show somewhat similar performance in terms of placement density. Averaged over all the cases, attaining 10% mean missed detection error is achieved by having 30% sensor density. Seen another way, we can reduce the real-time monitoring of each fuse by 70% by tolerating a small amount of errors in the outage detection decision.

Figure 6(b) shows the reduced network $R1-12.47-1$. We observe a commonly appeared feature of the sensor placement: many parallel branches have sensors placed, since otherwise issues of indistinguishability would arise. For example, sensors are placed in almost all of the edges which are descendants of vertex v_3. The reason is that the mean loads looking downstream of these descendants are almost identical. Thus not placing sensors on these downstream parallel branches would lead to observations of similar load changes when outages occur on different branches among them. Specifically, the downstream observed loads at these edges below v_3 are: $\{9.9,\ 101.6,\ 43.9,\ 59.6,\ 104.8,\ 120.3,\ 43.9\}$.

In contrast, in junctures further downstream, child subtrees can have much more distinguishable mean loads, and determining their outage condition becomes easier at those junctures even without placing sensors on their immediate descendant edges.

C. Sensitivity Analysis on General Line and Tree Networks



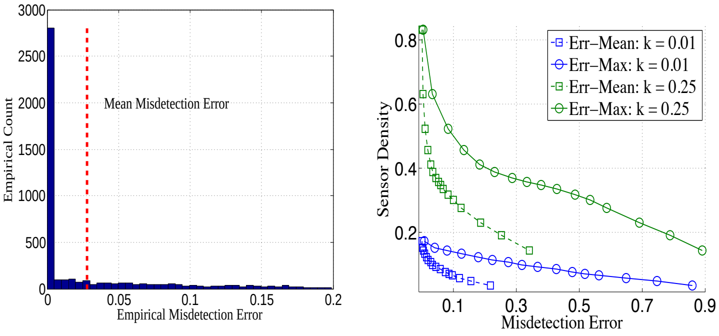
(a) (b)

Fig. 7. The effect of optimization parameter, error target P^{target}, and relative forecast error κ on both line (7(a)) and tree (7(b)) networks.

Figure 7 shows the sensitivity of the line and tree networks under different simulation parameters. Both networks are of size of 100 nodes, the tree was generated using the method in [21]. In an ideal line network with extremely high forecast accuracy ($\kappa = 1\%$), 1 or 2 sensors are required for extremely low missed detection errors. This extreme situation does not occur in practice, but serves as a baseline for realistic networks. Figure 7(a) shows that the required sensor density decreases quite quickly vs. P^{target}. The relation between sensor density and P^{target} is smoothly decaying. In comparison, randomly generated tree networks require on average $2-3$ times as many sensors to achieve the same error target.

D. Missed Detection Error

Optimization (OPT-2) is designed to minimize the maximum missed detection error among all possible hypotheses. The objective can however be too conservative of a requirement. Therefore it is useful to understand the nature of the actual hypothesis missed detection errors that arise from a given sensor placement. Figure 8(a) shows the distribution of missed detection probabilities for a tree network. Setting $P^{\text{target}} = 0.2$, and $\kappa = 0.3$ we record the value of each hypothesis error. The empirical maximum error is close to the target 0.2. This makes sense because in successively solving the feasibility problem, the area networks expand until the maximum error reaches P^{target}. However, the vast majority of the missed detection probabilities are much less than the target. For this example in particular the errors for 34% of the hypothesis are less than $1e^{-3}$ therefore essentially zero.



(a) (b)

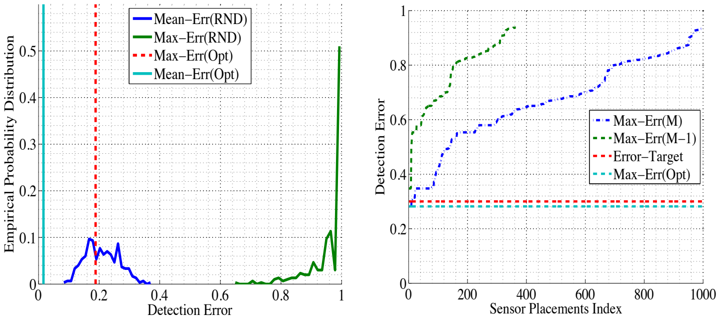
Fig. 8. Hypothesis missed detection analysis: 8(a) error histogram for $P^{\text{target}} = 0.2$; 8(b) Max-error reduction to mean-error.

In comparing the mean and maximum errors for the range of achievable values of κ and P^{target}, the mean error is on average 25% of P^{target} (cf. Figure 8(b)). The maximum error in the network and P^{target} are always very close to each other.

E. Comparison with Random and Exhaustive Placement

We now investigate how performance of a randomized placement compares to that of placement by Algorithm 2. In Figure 9(a), the tree network used in Section VII-C is used, and an optimal placement $\mathcal{M}^{\star}$ is generated. The empirical missed detection error (max and mean) is evaluated for placement $\mathcal{M}^{\star}$. In addition, 1000 random placement positions are evaluated from the entire search space, with both the mean and maximum errors evaluated. It is clear that the random placement performs extremely poorly compared to the optimized placement $\mathcal{M}^{\star}$.

To understand why this is the case, recall that the feasibility problem will find some $|\mathcal{M}| \leq M$ with error below the target. The entire search space is of size $\alpha = |\{\mathcal{M}\}| = \binom{N}{M}$. There can exist



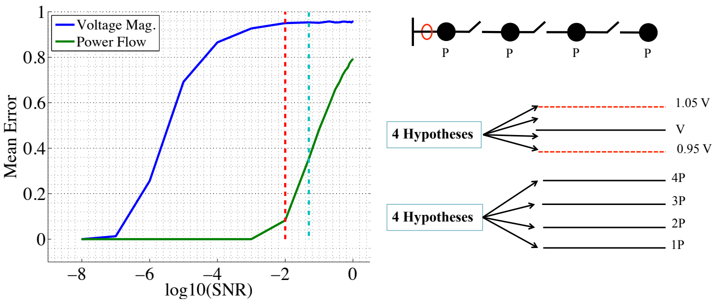
(a) (b)

Fig. 9. (a) Evaluation of randomized placement in terms of max and mean error vs. the optimal placement $\mathcal{M}^\star$. (b) Exhaustive enumeration of all placements vs. optimal placement $\mathcal{M}^\star$.

placement achieving an equal or smaller error value, and the number of such placements, k_α can be either singular or very small in large trees. Therefore, randomly sampling t placements and choosing the minimum will yield such a desired solution with the following probability: $\Pr(\mathcal{M}^\star \text{ chosen}) \approx 1 - \left(\frac{\alpha - k_\alpha}{\alpha}\right)^t$. Assuming $k_\alpha \ll \alpha$, if we want $\Pr(\mathcal{M}^\star \text{ chosen}) = P_c$, this leads to the following approximate number of samples: $t \approx -\frac{\alpha}{k_\alpha}\log(1-P_c)$. If there is a single unique solution, or if $k_\alpha = \Omega(1)$, then the number of samples we must take is $O(|\{\mathcal{M}\}|)$. In the simulated example in Figure 9(a), $|\{\mathcal{M}\}| = 5 \times 10^{12}$. From the arguments in Section H, the multiplicity of solutions is likely on the order of nodes with multiple children, which is $\Omega(N)$. This results in $t = O(\frac{1}{N}\binom{N}{M})$, thus explaining why random placement results in a poor missed detection even with a large number of sample placements.

Figure 9(b) illustrates an exhaustive sensor placement evaluation for a smaller tree. The optimization is run with an error target of 0.3, and the optimal placement produces a maximum error value of 0.282. With a total search space of 1001 possible placements, 9 of them were below the error target. However, it is clear the k_α is very small compared with the search space. This explains the performance in Figure 9(a). Note also that all placements of size $M-1$ have an error greater than the target, consistent with the statements in the proof of Theorem 1.

F. Comparison of Flow Based and Voltage Based Methods



(a) (b)

Fig. 10. (a) Toy line example voltage vs. power flow missed detection error under varying additive observation noise. (b) Conceptual model of voltage vs. power flow measurements dynamic range under various hypotheses. Power flow measurements will map into a range from $1P$ units of flow up to $4P$ units of flow in a linearly increasing manner. Voltage measurements will map into a range $\pm 5\% V$ units of voltage in a non-linear manner.

Many works in outage detection in distribution systems have proposed using voltage measurements to detect changes in distribution systems [14], [15]. In particular, small voltage deviations, which can be observed globally, are proposed to detect changes in the network.

In this section, we offer evidence that in the case of distribution systems, as opposed to transmission systems, measurements of flows are superior measurements for detecting changes in a network. The intuition for this can be seen in a simple line network in Figure 10(b), where 3 switches lead to 4 unique outage hypotheses. One hypothesis for each switch opened, and one for all switches closed. If each load is P units of per unit load, then the dynamic range of the entire set of hypotheses will range from $1P$ up to $4P$ in the case of power flow observation. If voltage measurements are used, then the 4 hypotheses will map into the 0.95 V to 1.05 V voltage range, which is the typical allowable voltage band that distribution systems are allowed to operate in. For this reason, it makes sense to use power flow measurements when detecting changes in system topology for distribution networks.

A simple simulation is performed to illustrate this via MatPower [22], which extends the diagram in Figure 10(b), to a larger detection space.

1) A line network is generated with 50 vertices.
2) Line r_i, x_i are drawn randomly from $r_i \sim U[0.001, 0.01]$ and $x_i \sim U[0.001, 0.01]$.
3) Loads are drawn uniformly from $p_i \sim U[0.25, 2.5]$ p.u.
4) A single sensor is placed at edge $e = (24, 25)$.

First, the voltage profile of each bus is verified to range from $1.064 \to 0.96$ V p.u. A single sensor is placed and outages are enumerated for all edge downstream of sensor. This guarantees that sensor will see some dynamic range of voltages. In fact, $1.035 \to 0.97$ V p.u. are seen by the voltage measurement. As a comparison, measuring flow in the same location will see a dynamic range of $1.3 \to 41.8$ P p.u.. For each hypothesis, a simple detector is constructed where additive noise of varying σ/μ is used to generate samples. The simulation consists of the following steps.

1) Observation Generation: $z_{obs,n} = z_{true}(h) + N(0, \sigma(h))$
2) ML Detection: $\hat{h}_n = \arg\min_{h \in H} \|z_{obs,n} - z(h)\|^2$
3) Mean Error Computation

Figure 10(a) shows the missed detection error from this experiment under varying additive noise. The range of additive errors were from $\sigma/\mu = \{1 \times 10^{-8} \dots 1\}$, with colored bars indicating 1% and 5% relative error values in the measurements. The plot indicates that even for 1% relative error of voltage sensing, hypothesis detection with a single observation can lead to high errors. In fact, most work relying on voltage measurements must use long time captures to improve their detector performance for this reason. Power flows, on the other hand, directly reflect changes in the number of downstream loads.

VIII. Conclusion

This work has proposed an outage detection framework combining power flow measurements on the lines of the distribution system along with consumption forecasts at the nodes of the network. In this framework, the ML detector for all (potentially multi-line) outage hypotheses is reduced to a set of scalar ML detectors for local outage hypotheses which are efficient to solve for. This decoupling leads to an efficient and optimal sensor placement algorithm which minimizes the maximum missed detection probability over the entire network. The system has been tested on typical distribution feeders provided by PNNL, as well as general line and tree networks. It has been shown that 10% mean missed detection error can be achieved by having only 30% sensor density.

References

[1] Y. Zhao, R. Sevlian, R. Rajagopal, A. Goldsmith, and H. V. Poor, "Outage detection in power distribution networks with optimally-deployed power flow sensors," in *Proc. of IEEE Power and Energy Society General Meeting*, 2013.

[2] Z. Sumic and R. Vidyanand, "Fuzzy set theory based outage determination," *Proc. of 1996 International Conference on Intelligent Systems Applications to Power Systems*, pp. 204–208, 1996.

[3] J. Aguero and A. Vargas, "Inference of operative configuration of distribution networks using fuzzy logic techniques - part i: real-time model," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1551– 1561, 2005.

[4] S. Mak and N. Farah, "Synchronizing scada and smart meters operation for advanced smart distribution grid applications," *Proc. IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–7, 2012.

[5] Y. Liu and N. Schulz, "Knowledge-based system for distribution system outage locating using comprehensive information," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 451–456, 2002.

[6] J. E. Tate and T. J. Overbye, "Line outage detection using phasor angle measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644–1652, 2008.

[7] Y. Zhao, J. Chen, A. Goldsmith, and H. V. Poor, "Identification of outages in power systems with uncertain states and optimal sensor locations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1140–1153, 2014.

[8] Y. Zhao, J. Chen, and H. V. Poor, "Learning to infer: A new variational inference approach for power grid topology identification," in *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, 2016.

[9] V. Kekatos and G. B. Giannakis, "Joint power system state estimation and breaker status identification," in *Proc. IEEE, North American Power Symposium (NAPS)*, 2012, pp. 1–6.

[10] H. Zhu and G. B. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2215–2224, 2012.

[11] J. Zhu, D. Lubkeman, and A. Girgis, "Automated fault location and diagnosis on electric power distribution feeders," *IEEE Transactions on Power Delivery*, vol. 12, no. 2, pp. 801–809, 1997.

[12] C. Fukui and J. Kawakami, "An expert system for fault section estimation using information from protective relays and circuit breakers," *IEEE Transactions on Power Delivery*, vol. 1, no. 4, pp. 83–90, 1986.

[13] R. Balakrishnan and A. Pahwa, "A computer assisted intelligent storm outage evaluator for power distribution systems," *IEEE Transactions on Power Delivery*, vol. 5, no. 3, pp. 1591–1597, 1990.

[14] R. Pereira, L. da Silva, M. Kezunovic, and J. Mantovani, "Improved fault location on distribution feeders based on matching during-fault voltage sags," *IEEE Transactions on Power Delivery*, vol. 24, no. 2, pp. 852–862, 2009.

[15] M. Kezunovic, "Smart fault location for smart grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 11–22, 2011.

[16] D. Srinivasan, R. Cheu, Y. Poh, and A. Ng, "Automated fault detection in power distribution networks using a hybrid fuzzy–genetic algorithm approach," *Elsevier Engineering Applications of Artificial Intelligence*, vol. 13, no. 4, pp. 407–418, 2000.

[17] S. E. LLC. (2016) Mm3 intelligent sensor. [Online]. Available: http://www.sentient-energy.com/mm3-intelligent-sensor/

[18] K. Schneider, Y. Chen, D. Chassin, R. Pratt, D. Engel, and S. Thompson, "Modern grid initiative distribution taxonomy final report," *arXiv preprint arXiv:1404.0058*, 2008.

[19] P. Cirillo, "Are your data really Pareto distributed?" *Physica A Statistical Mechanics and its Applications*, vol. 392, pp. 5947–5962, Dec. 2013.

[20] R. Sevlian, P. Siddarth, and R. Rajagopal, "Day ahead electricity load forecasting on varying levels of aggregation," *Power Engineering Letters*, 2014.

[21] A. Rodionov and H. Choo, "On generating random network structures: Trees," in *Computational Science ICCS 2003*. Springer, 2003, pp. 879–887.

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

[23] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*. Springer New York, 2006, vol. 1.

Appendix

A. Nomenclature Table

$\mathcal{T}$	Tree network representation of a distribution feeder
(V,E)	Vertex and edge set of tree $\mathcal{T}$
$V(H)$	Set of vertices that are connected to root under outage hypothesis H.
$\textbf{desc}(v)$	Descendants of vertex v.
$\textbf{child}(v)$	Children of vertex v.
x_n, $\mathbf{x}$	Scalar load and forecast value at vertex v
ϵ_n, ϵ	Forecast residual for load $l(v)$
σ_n^2, Σ	Forecast residual variance and covariance matrix.
$\mathcal{H}^1$	Single outage hypothesis and element and set.
$\mathcal{H}^k$	k-outage hypothesis.
$\mathcal{H}_u$	Unique hypothesis where no edges are downstream of any others.
A, $\mathcal{A}$, $\mathcal{A}^+$	Local area network $A \in \mathcal{A}$; Pruned areas under binary flow processing.
f	Binary flow indicator $f = \mathbb{I}_{\{\textbf{child}(s_i)>0\}}$
$\mathcal{H}_u^+(A_k, f)$	Set of local area hypotheses post binary flow processing.
$\mathcal{H}_{k,i(k)}$	$i(k)^{th}$ hypothesis in area A_k. Used to reconstruct global hypothesis H_i.
$\mathcal{M}$	Sensor placement ($\mathcal{M} \subset E$)
$\mathbf{s}$	Set of observations on edges of network
r_{ij}	Acceptance region for pairwise test of hypotheses: H_i and H_j
$R_{\mathcal{H}}(H)$	Acceptance region of hypothesis H over all alternatives in $\mathcal{H}$.
$P_{E,C}(H,\mathcal{M})$,	Probability of error (E) and correct detection (C) for hypothesis H, under placement $\mathcal{M}$.
$P_E^{max}(A)$	Maximum probability of incorrect detection over all hypothesis error in area A.
$P_C^{min}(A)$	Minimum probability of correct detection over all hypothesis error in area A.

B. MAP Detection for Outage Hyptheses

Here we show how the general MAP detector rule can be evaluated for where we combine edge flows $\mathbf{s}$, load forecasts $\hat{\mathbf{x}}$ and candidate outages H.

$$\begin{aligned}
\hat{H} &= \arg\max_{H \in \mathcal{H}^k} \Pr\left(H \mid \mathbf{s},\ \hat{\mathbf{x}}\right) && (29) \\
&= \arg\max_{H \in \mathcal{H}^k} \frac{\Pr\left(\mathbf{s},\ \hat{\mathbf{x}} \mid H\right) \Pr\left(H\right)}{\Pr\left(\mathbf{s},\ \hat{\mathbf{x}}\right)} && (30) \\
&= \arg\max_{H \in \mathcal{H}^k} \Pr\left(\mathbf{s},\ \hat{\mathbf{x}} \mid H\right) \Pr\left(H\right) && (31) \\
&= \arg\max_{H \in \mathcal{H}^k} \Pr\left(\mathbf{s} \mid \hat{\mathbf{x}},\ H\right) \Pr\left(\hat{\mathbf{x}} \mid H\right) \Pr\left(H\right) && (32) \\
&= \arg\max_{H \in \mathcal{H}^k} \Pr\left(\mathbf{s} \mid \hat{\mathbf{x}},\ H\right) \Pr\left(\hat{\mathbf{x}}\right) \Pr\left(H\right) && (33) \\
&= \arg\max_{H \in \mathcal{H}^k} \Pr\left(\mathbf{s} \mid \hat{\mathbf{x}},\ H\right) \Pr\left(H\right) && (34) \\
&= \arg\max_{H \in \mathcal{H}^k} \Pr\left(\mathbf{s} \mid \hat{\mathbf{x}},\ H\right) && (35)
\end{aligned}$$

Lines (29) - (31) convert the MAP detector to a likelihood detector with prior weights. Line (32) conditions on the load forecast $\hat{\mathbf{x}}$. Since $\hat{\mathbf{x}}$ does not depend on the outage hypothesis, (only $\mathbf{s}$ does), the term can be removed leading to (34). In (35), we assume a uniform prior over all hypotheses, however this does not have to be the case.

Given the assumption of each edge going into outage with some fixed prior probability, a single edge outage hypothesis should have $\Pr(H) = \rho$, while a k outage condition should have a prior of $\Pr(H) = \rho^k$. This motivates enumerating fewer outage hypotheses when evaluating $\mathcal{H}_u$ in practice.

C. Extended Discussion of Recursive Evaluation of $\mathcal{H}_u$

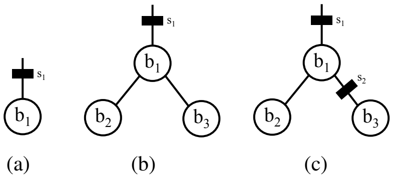


Fig. 11. 11(a) Worked example 1. 11(b) Worked example 2. 11(c) Worked example 3.

We present here additional worked out examples, that show how the recursive definition can enumerate all hypotheses and focus on some corner cases that must be defined. Recall the recursive definition:

$$\mathcal{H}_u(b) = E(b) \cup \left(\bigcup_{\mathbf{b}\in\mathbb{P}(\mathbf{child}(b))} \left(\bigwedge_{b\in\mathbf{b}} \mathcal{H}_u(b) \right) \right)$$

1) Example 1 (Figure 11(a)): This is the simplest case to evaluate and is:

$$\mathcal{H}_u = E(b_1) \cup \emptyset$$

The null hypothesis set arises from evaluating $\mathbb{P}(d(b_1)) = \emptyset$, since b_1 has no children.

2) Example 2 (Figure 11(b)): This is the simplest case to evaluate and is:

$$\begin{aligned}\mathcal{H}_u &= \{E(b_1) \cup \emptyset \cup \mathcal{H}_u(b_2) \cup \mathcal{H}_u(b_3) \cup \mathcal{H}_u(b_2) \times \mathcal{H}_u(b_3)\}\\ &= \{E(b_1) \cup \emptyset \cup \{\emptyset \cup E(b_2)\} \cup \{\emptyset \cup E(b_3)\} \cup \\ &\qquad \{\emptyset \cup E(b_2)\} \times \{\emptyset \cup E(b_3)\}\}\\ &= \{\emptyset \cup E(b_1) \cup E(b_2) \cup E(b_3) \cup (E(b_2) \times E(b_3))\}\end{aligned}$$

This example is reduced to it's final form in eq. (12), in Section V-A. However, the following equalities are omitted in the enumeration:

$$\emptyset \cup \emptyset = \emptyset \tag{36}$$
$$\emptyset \times \emptyset = \emptyset \tag{37}$$
$$\emptyset \times e_i = e_i \tag{38}$$

The final relation leads to $\emptyset \times E(b) = E(b)$.

3) Example 3 (Figure 11(c)): Consider the two binary flow indicators $\mathbb{I}_{\{s_2>0\}} = \{1\}$ and $\{0\}$. In the first case, we have the following product set:

$$\begin{aligned}\mathcal{H}^+(A_1, \{1\}) &= \{E(b_2) \cup \emptyset_1\} \times \{E(b_3) \cup \emptyset_2\}\\ &= \{E(b_2) \times E(b_3) \cup \emptyset_1 \times E(b_3) \cup E(b_2) \times \emptyset_2 \cup \emptyset_2 \times \emptyset_2\}\\ &= \{E(b_2) \times E(b_3) \cup E(b_2) \cup E(b_3) \cup \emptyset\}\end{aligned}$$

We use the fact that $\emptyset \times E(b) = E(b)$ and that the product $\emptyset_1 \times \emptyset_2 = \emptyset$, which is the global null hypothesis from the naive enumeration in Example 2. Similarly, enumerating the $f_1 = \{0\}$ case, we have that: $\mathcal{H}_u^+(A_1, \{0\}) = \{E(b_1)\}$. We see that splitting the hypotheses based on flow information, conserves the search space, since $\mathcal{H}_u^+(A_1, \{1\}) \cup \mathcal{H}_u^+(A_1, \{0\}) = \mathcal{H}_u(A_1)$.

4) Tree $\mathcal{T}_1$: For tree $\mathcal{T}_1$, we have:

$$\begin{aligned}\mathcal{H}_u(b_1) &= \{\emptyset \cup E(b_1) \cup E(b_2) \cup \mathcal{H}_u(b_3) \cup E(b_2) \times \mathcal{H}_u(b_3)\}\\ \mathcal{H}_u(b_3) &= \{\emptyset \cup E(b_3) \cup E(b_4) \cup E(b_5) \cup \{E(b_4) \times E(b_5)\}\}\end{aligned}$$

D. General Hypothesis Decoupling

These two cases provide the intuition for a general procedure which is as follows: Given binary information from flows, all areas A_k with rooted sensor with $s_k = 0$, are discarded in generating a local hypothesis. Each node in the branch graph is assigned a label, $l \in L$ $L = \{P, Z, U\}$ for (P) positive, (Z) zero, and (U) undetermined branches. These are defined as follows:

> **Positive Branch**: Branch is upstream from a sensor measuring positive flow, therefore can never be evaluated in any outage hypothesis. Also, it's immediate parent branch cannot be enumerated either.
>
> **Zero Branch**: This branch is directly upstream from a zero measurement therefore it's edges must always be enumerated in any outage hypothesis.
>
> **Undetermined Branch**: This branch has no information, so is enumerated without any restriction.

This definition leads to the following procedure to enumerate $\mathcal{H}^+(A_i, \mathbb{I}_{\{\mathbf{child}(s_i)>0\}})$. First each branch-node is labeled with the following procedure:

> **Initialization** Branch with descendent sensor (1) $s > 0$ assigned label P and (2) $s = 0$ assigned label Z, and (3) no descendants assigned label U.
>
> **Update** Given a current branch node b and the set of children, the node is assigned as follows: (1) If any descendent node is labelled P then it must be labelled P. (2) If descendants are U and Z, then it must be labeled U.

Once the the branch-nodes are labeled, enumerating $\mathcal{H}^+(A_i, \mathbb{I}_{\{\mathbf{child}(s_i)>0\}})$ can be done recursively using the following rules:

> **Positive Rule** Never enumerate a branch $(E(b))$ with positive flow label (P).
>
> **Zero Rule** When evaluating the recursive definition $\mathcal{H}(b)$ on an element of the power set. If any descendent is labelled Z, only evaluate product set elements that contain this branch.

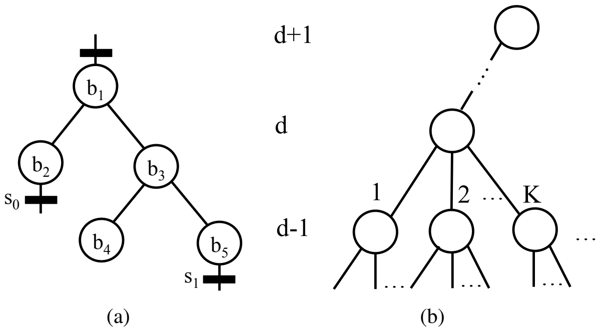


Fig. 12. 12(a) General network reduced to individual branches. 12(a) Worst case tree network of depth D and K children for each vertex.

An example local area is provided in Figure 12(a). The general method is applied under each of the binary flow cases, where the results are shown in Table II. The method is applied to each binary flow, and the branches to be enumerated are given.

TABLE II

Binary Flow Indicator $\mathbb{I}_{\{\mathbf{child}(s_i)>0\}}$	Branch Labels	Hypotheses Branches
$0, 0$	U, Z, U, U, Z	b_1, b_2, $b_2 \times b_3$, $b_2 \times b_5$, $b_2 \times b_4 \times b_5$
$0, 1$	P, Z, P, U, P	$b_2, b_2 \times b_4$
$1, 0$	P, P, U, U, Z	b_3, b_5, $b_4 \times b_5$
$1, 1$	P, P, P, U, P	b_4

E. Complexity Analysis

In analyzing the complexity of various algorithms we assume the tree in Figure 12(b). Each vertex has K children and is of depth D. It is clear that the number of edges is related to these quantities by $E = \frac{K^D - 1}{K - 1}$.

1) Evaluating $P_E^{max}(A)$: For simplicity, we focus on a binary tree, so $K = 2$ and $|E| = 2^D - 1$. The possible hypotheses using (8) where every edge is a single branch is given by

$$\mathcal{H}_u(e_i) = \{e_i \cup \mathcal{H}_u(\mathbf{child}(e_i, 1)) \cup \mathcal{H}_u(\mathbf{child}(e_i, 2)) \quad (39)$$
$$\cup\, \mathcal{H}_u(\mathbf{child}(e_i, 1)) \times \mathcal{H}_u(\mathbf{child}(e_i, 2))\} \quad (40)$$

The size of hypotheses at each depth $C(d)$ is therefore,

$$C(d+1) = 1 + 2C(d) + C(d)^2 \quad (41)$$
$$= (C(d) + 1)^2 \quad (42)$$

It can be shown from (42) that $C(d) = O(2^{2^d})$. Therefore the number of hypotheses are double exponential in the depth of the tree. For the entire tree, this leads to $C(D) = O(2^{|E|})$. Therefore $|\mathcal{H}_u| = O\left(2^{|E|}\right)$, exponential in the size of the graph, which is no better than the naive calculation without removing non-unique outages.

This cost may be averted due to the following:

Fixed Hypothesis Size. The MAP detector requires a prior probability of hypotheses. Since multiple edge outages are less likely, they don't always have to be enumerated. For example, considering only single edge outages leads to $|\mathcal{H}_u| = O(|E|)$ complexity for an area.

Small Area Sizes, and Binary Flow Segmentation. The number of hypotheses are exponential in $|E|$ for an area. Given many sensors, this can divide the number of edges for an area considerably. Additionally, the binary flow information from downstream sensors will on average divide each $\mathcal{H}_u^+(A, f)$ by a factor of $2^{|\mathbf{child}(s)|}$.

2) Evaluation of Algorithm 2 using greedy strategy: The following analysis is in terms of the evaluation of $T = P_E^{max}(A)$. The greedy strategy will have to evaluate all 2^K subproblems at each vertex, and choose the minimum $P_E^{max}(A)$. The worst case complexity is therefore $O(T|E|\ 2^K)$, which reduces to $O(|E|T)$, since K is a constant.

3) Evaluation of Algorithm 2 using optimal strategy: The optimal strategy will expand the problem size by a factor 2^K when the area network root moves upstream. This will occur D times, so sub-problems to consider after a depth of D will be $(2^K)^D$ which for a binary tree becomes $(2^D)^K$ leading to the final complexity $O(T|E|^K)$.

F. Proxy Function Optimization

The placement problem OPT-1 will output the optimal sensor locations and minimizing maximum missed detection error $\alpha^\star(\mathcal{M}^\star)$, as:

$$\alpha^\star(\mathcal{M}^\star) = \min_{|\mathcal{M}|=M} \max_{H \in H_u} P_E(H, \mathcal{M})$$

This can be approximated by using the decoupling of hypotheses in different areas. Recall the solution to OPT-1 repeated above, is the distributed detector in Algorithm 1 where each area performs a local hypothesis $\hat{H}, \ldots, \hat{H}_M$. The complete hypothesis is only correct if every local detection output is correct. So for any hypothesis, we have following lower bound:

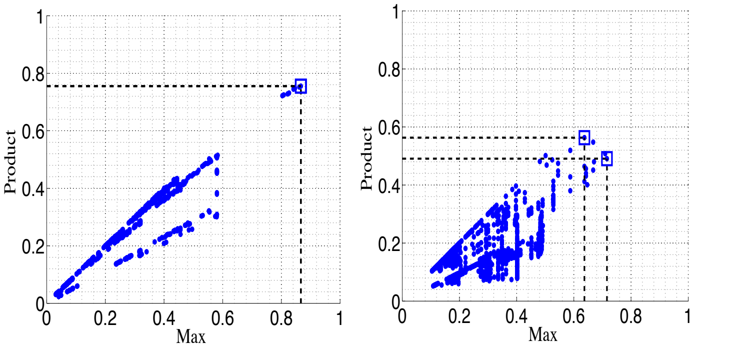
$$\begin{aligned}
&\min_{H \in H_u} Pr(\hat{H} = H; \mathcal{M}) \\
&\qquad = \min_{H \in H_u} \left(\prod_{\forall A_i,\ H_i} Pr(\hat{H}_i = H_i; \mathcal{M}) \right) && (43) \\
&\qquad \geq \prod_{\forall A \in \mathcal{A}} \min_{H \in \mathcal{H}(A)} Pr(\hat{H} = H; A) && (44) \\
&\qquad = \prod_{\forall A \in \mathcal{A}} P_C^{min}(A). && (45)
\end{aligned}$$

Line (43) follows from the decoupling of the decentralized detector. The overall MAP decision can be correct only if each local MAP decision is correct. For any $H \in \mathcal{H}_u$ the probability of each area making a correct decision is always greater than the worst case probability of correct decision for each area. We interchange the sensor placement and area notation since local areas are constructed from sensor placements. Here $P_C^{min}(A)$ is the minimum probability of correct detection within a local area A. This lower bound can be used to first upper bound the optimal $\alpha^\star$. Finally, only an approximate solution to the upper bound is formulated.

$$\begin{aligned}
\alpha(\mathcal{M}^\star) &= \min_{|\mathcal{M}|=M} \max_{H \in H_u} P_E(H, \mathcal{M}) && (46) \\
&= \min_{|\mathcal{M}|=M} \max_{H \in H_u} (1 - P_C(H, \mathcal{M})) && (47) \\
&= 1 - \max_{|\mathcal{M}|=M} \min_{H \in H_u} P_C(H, \mathcal{M}) && (48) \\
&\leq 1 - \max_{|\mathcal{M}|=M} \prod_{\forall A \in \mathcal{A}} P_C^{min}(A) && (49) \\
&\approx 1 - \max_{|\mathcal{M}|=M} \min_{\forall A \in \mathcal{A}} P_C^{min}(A) && (50) \\
&= 1 - \max_{|\mathcal{M}|=M} \max_{\forall A \in \mathcal{A}} (1 - P_E^{max}(A)) && (51) \\
&= \min_{|\mathcal{M}|=M} \max_{\forall A \in \mathcal{A}} P_E^{max}(A) && (52)
\end{aligned}$$

Optimization OPT-1 is identical to (47), since the probability of a single hypothesis error can be exchanged for it's compliment. The min-max to max-min change is due to the negative sign in 48. In line 48, instead of maximizing the minimum correct probability over all hypotheses we maximize a computationally tractable lower bound which is the product $\prod_{\forall A \in \mathcal{A}} P_C^{min}(A)$. In 50 we introduce a close approximate solution which is the following: Instead of maximizing the product of $P_C^{min}(A)$ for each A, it is sufficient to maximizing the minimum of each $P_C^{min}(A)$ Experimentally the two solutions have been shown identical for a large number instances, and only sub-optimal in a small number of cases where the gap is small.

Experimentally the two solutions have been shown identical for a large number instances, and only sub-optimal in a small number of cases where the gap is small. Figure 13(a), 13(b), shows a pair of randomly generated trees with $N = 15$ nodes



(a)

(b)

Fig. 13. 13(a) Brute force placement evaluation where the two objectives are identical. 13(b) Brute force placement evaluation where the two objectives differ.

with random loads and a forecast coefficient of variation of 0.02. In both cases, the bottom up placement was used to determine $\mathcal{M}^g$, where $|\mathcal{M}^g| = 5$.

A brute force enumeration of all $\binom{15}{5}$ placements is evaluated for $\max_{\forall A \in \mathcal{A}} P_C^{min}(A, \mathcal{M})$ and $\prod_{\forall A \in \mathcal{A}} P_C^{min}(A, \mathcal{M})$.

In both cases, there is a strong correlation between the two solutions. The two solutions are not however equal Figure 13(a), the solutions are identical, while in 13(b), the two solutions differ by 7.2%. Intuitively they should intuitively be very close to each other. Decreasing one area error will increase the other area's error due to the monotonic growth of $P_E^{max}(A)$ and the finite tree size. Therefore, maximizing the product of all the terms tends to a solution where each area error is as close to each other as possible. Minimizing the maximum error often leads to such a solution, since we must trade off one area error for another.

G. Proof of Theorem 1

We prove Theorem 1 by showing the following:

1) The objective function $P_E^{max}(A)$ monotonically increases for nested areas.
2) Algorithm 2, will recover the solution to OPT-3

First we prove propositions 1 and state a conjecture shown to hold in large scale simulation experiments. These are needed to prove Lemma 2 which is needed to prove Theorem 1.

First consider the following:

Definition 1. *For a single pairwise test, H_i vs H_j we have the following decision region:*

$$r_{i,j} = \{\mathbf{s} \in R^M : \Pr(\mathbf{s}|H_i) \geq \Pr(\mathbf{s}|H_j)\}. \tag{53}$$

The observation space is therefore partitioned into two regions. So the detector is the following:

$$\hat{H} = \begin{cases} H_i, & \mathbf{s} \in r_{ij} \\ H_j, & \mathbf{s} \notin r_{ij}. \end{cases} \tag{54}$$

For the one-to-many ML test: H_i vs $\forall H_j \in H$, we have an acceptance region defined as:

$$R_{\mathcal{H}}(H_i) = \{\mathbf{s} \in R^M : \Pr(\mathbf{s}|H_i) \geq \Pr(\mathbf{s}|H_j)\ \forall H_j \in \mathcal{H}\}. \tag{55}$$

Lemma 1. *An equivalent definition is*

$$R(H_i) = \bigcap_{j:H_j \in \mathcal{H}} r_{i,j}. \tag{56}$$

Proof: Using the definition of the right hand side, we have

$$\begin{aligned} \bigcap_{j \in H} r_{i,j} &= \{s : s \in r_{i,1} \cap \ldots \cap s \in r_{i,N}\} \\ &= \{s : \Pr(\mathbf{s}|H_i) \geq \Pr(\mathbf{s}|H_1) \cap \ldots \\ &\qquad \cap \Pr(\mathbf{s}|H_i) \geq \Pr(\mathbf{s}|H_N)\} \\ &= \{s : \Pr(\mathbf{s}|H_i) \geq \Pr(\mathbf{s}|H_j)\ \forall H_j \in \mathcal{H}\} \\ &= R_{\mathcal{H}}(H_i). \end{aligned}$$

■

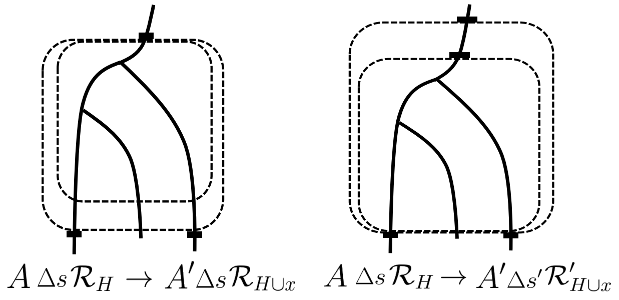
Note, that this statement is for the ML classifier. Under an arbitrary classifier, a procedure of constructing one-to-many classifier will lead to ambiguity. See [23] (pg. 183) for discussion.

Conjecture 1. *Given a ML detection problem with a set of hypothesis of the form: $s_k \sim N(\mu_k, \sigma_k^2 + \Delta)$ for $k = 1, \ldots, K$. The missed detection error for each hypothesis will monotonically increase w.r.t Δ.*

This is seen to hold with 200,000 random problem instantiations. We now state Lemma 2.

Definition 2. *Two area networks are nested $A \subset A'$ if the vertices of each area V, V' are such that $V \subset V'$.*

Lemma 2. *Given two area networks A and A' where $A \subset A'$, $P_E^{max}(A) \leq P_E^{max}(A')$.*


$A\,\Delta s \mathcal{R}_H \rightarrow A' \Delta s \mathcal{R}_{H \cup x}$
$A\,\Delta s \mathcal{R}_H \rightarrow A' \Delta s' \mathcal{R}'_{H \cup x}$

(a)

(b)

Fig. 14. 14(a) Case 1 showing growth by adding new nodes by moving *terminal sensor down*. The conditional pdf $\Delta s|H\ \forall h \in H$ does not change. However the acceptance region shrinks from $R_{\mathcal{H}}$ to $R_{\mathcal{H} \cup x}$. 14(b) Case 2 showing growth by adding new nodes by moving *root sensor up*. The conditional pdf $\Delta s'|H\ \forall h \in H$ will change. Again, the acceptance region shrinks from $R_{\mathcal{H}}$ to $R'_{H \cup x}$.

Proof: Given some $\alpha^* = P_{\text{err}}^{\max}(A)$, which is the maximum missed detection probability of the local hypothesis in area A which is evaluated at some hypothesis H^*. We aim to show that for an enlarged area A', the error probability for $H^\star$ will *always* be larger, which is quite intuitive. Therefore the maximum error $P_E^{\max}(A')$ regardless if H^* maximizes the missed detection in the enlarged area.

Expansion of A is analyzed in two cases:

Case 1 Terminal sensors expand downstream (away from v_0) shown in Figure 14(a).

Case 2 Root sensor of A moves upstream (closer to v_0) shown in Figure 14(b).

We use the following shorthand: Hypotheses for area A and A': $\mathcal{H} \triangleq \mathcal{H}(A)$, $\mathcal{H} \cup x_i \triangleq \mathcal{H}(A')$. The set x_i differ in how the area is enlarged, where $i = 1, 2$ for case 1 and case 2. Therefore under case 1, and 2 we have $R_{\mathcal{H} \cup x_1}(H_i)$ and $R_{\mathcal{H} \cup x_2}(H_i)$. Now we show how the effective measurement distribution (as defined in (28)) and acceptance regions changes under each case.

Case 1 For all hypothesis $H_i \in \mathcal{H}$ we have that $\mu'_i = \mu_i$ and $\sigma'^2_i = \sigma^2_i$. The distribution $\Delta s|H_i$ is unchanged. Given that $R_{\mathcal{H}}(H_i) = \bigcap_{j:H_j \in \mathcal{H}} r_{ij}$, the new acceptance region is $R_{\mathcal{H}\cup x_1}(H_i) = \bigcap_{j:H_j \in \mathcal{H}\cup x_1} r_{ij}$, with r_{ij} from the original alternatives unchanged.

Case 2 For all hypothesis $H_i \in \mathcal{H}$ we have that $\mu'_i = \mu_e + \mu_i$ and $\sigma'^2_i = \sigma^2_e + \sigma^2_i$. The distribution and acceptance region change $\Delta s|H_i \rightarrow \Delta s'|H_i$. The new acceptance region is the following: $R_{\mathcal{H}\cup x_2}(H_i) = \{\bigcap_{j:H_j \in \mathcal{H}} r'_{ij}\} \bigcap \{\bigcap_{j:H_j \in \mathcal{H}} r_{ij}\}$, where the acceptance regions under the previous area alternatives are now different.

To see why the distribution $\Delta s|H_i$ changes, first recall that:

$$\Delta s_i|H_k \sim N\left(\sum_{v \in V_i \setminus V_k} \mu(v) - \mu_T, \sum_{v \in V_i \setminus V_k} \sigma^2(v) - \sigma^2_T\right) \quad (57)$$

$$\sim N\left(\mu_k, \sigma^2_k\right) \forall H \in \mathcal{H} \quad (58)$$

The terms μ_T and σ^2_T are the sum of loads forecasts and variances of all terminal sensors. Now moving the root node upstream leads to:

$$s'_i|H_k \sim N\left(\sum_{v \in V'_i \setminus V_k} \mu(v) - \mu_T, \sum_{v \in V'_i \setminus V_k} \sigma^2(v) - \sigma^2_T\right) \quad (59)$$

$$\sim N\left(\mu_e + \mu_i, \sigma^2_e + \sigma^2_i\right) \forall H \in \mathcal{H}. \quad (60)$$

Therefore changing the position of s_i so as to add additional vertices will increase every original hypothesis mean and variance by the same amount.

Now consider Case 1 first, where we merely add new alternatives, keeping the distributions of $\Delta s|H$ unchanged. Here we have:

$$\Pr(\Delta s \in R_{\mathcal{H}\cup x}(H^*)|H^* \text{ true}) \quad (61)$$

$$= \Pr(\Delta s \in \bigcap_{j:H_j \in \mathcal{H}\cup x} r_{i,j} \mid H^* \text{ true}) \quad (62)$$

$$= \Pr(\{\Delta s \in \bigcap_{j:H_j \in \mathcal{H}} r_{i,j}\} \cap \{\bigcap_{j:H_j \in x} \Delta s \in r_{i,j}\}| \; H^* \text{ true}) \quad (63)$$

$$\leq \Pr(\Delta s \in \bigcap_{H_j \in \mathcal{H}} \in r_{i,j}|H^* \text{ true}) \quad (64)$$

$$= \Pr(\Delta s \in \mathcal{R}_{\mathcal{H}}(H^*)|H^* \text{ true}). \quad (65)$$

Line 62 defines the area $R_{\mathcal{H}}(H^*)$ using proposition 1. This is split into the the intersection of two separate events using our definitions of $\mathcal{H}$ and $\mathcal{H}\cup x$. Next we use the fact that $\Pr(A \cap B) \leq \Pr(A)$. Therefore if $\Pr(\Delta s \in R_{\mathcal{H}}(H^*)|H^* \text{ true}) \leq \Pr(\Delta s \in R_{\mathcal{H}\cup x_2}(H^*)|H^* \text{ true})$ then $P^{max}_E(A) \leq P^{max}_E(A')$.

We next prove Case 2 where not only are more alternatives considered for H^*, but $\Delta s|H^*$ is translated by fixed amount μ_e, σ^2_e in mean and variance. This implies that:

$$\Pr(\Delta s' \in R_{\mathcal{H}\cup x}|H^* \text{ true})$$

$$\leq \Pr(\Delta s' \in R_{\mathcal{H}}(H^*)|H^* \text{ true}) \quad (66)$$

$$= \Pr(\Delta s' - \mu_e \in R_{\mathcal{H}}(H^*) - \mu_e|H^* \text{ true}) \quad (67)$$

$$\leq \Pr(\Delta s \in R_{\mathcal{H}}|H^* \text{ true}). \quad (68)$$

The inequality in line 66 uses the identical procedure in 62 - 64. In line 67 we are merely shifting the gaussian Δs and the acceptance region by μ_e using a shorthand notation. This can be done since the MAP test is scalar. Finally the inequality in line 68, follows from conjecture 1. ■

We can now prove Theorem 1.

Proof: The bottom up solution $\mathcal{M}^g$ moved to the root node enlarging each area network so that each $P^{max}_E(A) < P^{target}$ but any further up will violate the target area. Consider some other method produces a solution $\mathcal{M}'$ which minimizes the number of sensors the error constraint, where $|\mathcal{M}'| < |\mathcal{M}^g|$. This implies that some area must increase in size, as compared to the $\mathcal{M}^g$ solution, and from Lemma 2, some area will violate the error constraint. ■

H. Greedy and Optimal Tree Action

optimal-tree-action: The correct action at a node-junction is to enumerate each $2^{|\textbf{child}(v_r)|-1}$ possible trees and process them until only one remains closest to the root. In the example in Figure 4, we must process to the root node processing both A_2 and A_3 in parallel as separate problem instances, with its own v_t and M^g. This is in contrast to the greedy strategy that chooses one placement and moves on. Each problem instance is then processed, until the area objective function violates P^{target}. All but one problem instance is kept; the one where v_t was closest to the root vertex.

It turns out that the **greedy-tree-action** and the **optimal-tree-action** procedures are in practice extremely close as discussed in Appendix H. Algorithm 2 can only implement this technique, since we do not grow the search space with multiple bottom up scenarios.

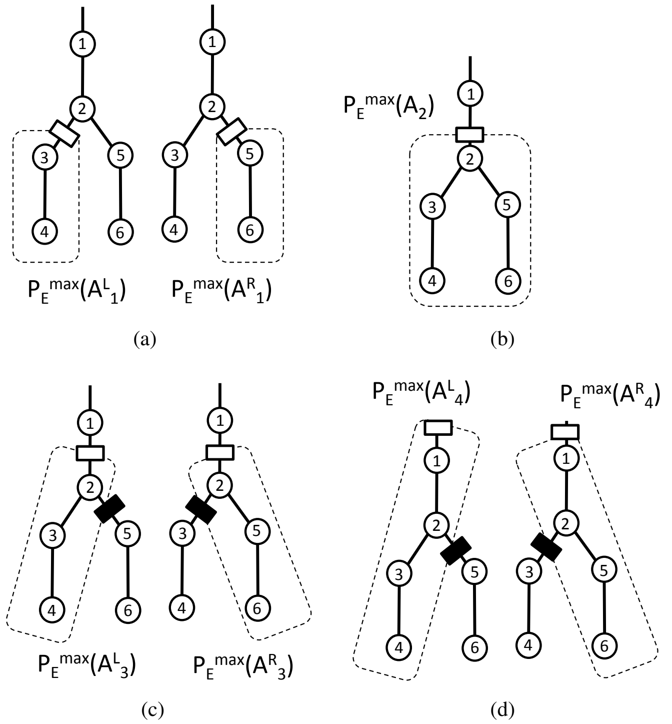


Fig. 15. 15(a) Both areas have maximum error smaller than target error: $P^{max}_E(A^L_1) < P^{target}$, $P^{max}_E(A^R_1) < P^{target}$. 15(b) The area's are combined, and tested where $P^{max}_E(A_2) > P^{target}$. 15(c) The greedy choice will choose A^L_3 as the candidate: $P^{max}_E(A^L_3) < P^{max}_E(A^R_3) < P^{target}$. 15(d) The correct choice is A^R_4 since $P^{max}_E(A^R_3) < P^{target} < P^{max}_E(A^L_3)$.

The bottom up placement using **greedy-tree-action** relies on moving the current node as close to root as possible while keeping the error $< P^{\text{target}}$. At a juncture, recall that the current network is chosen as the one which minimizes the maximum error, and continues with that choice. To see why this is sub optimal, consider Figure 15 shows a typical subtree where we have the following events:

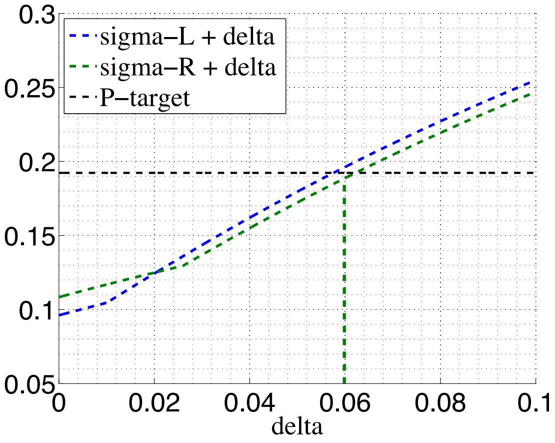
1) (Figure 15(a)) The bottom up algorithm will evaluate $P_E^{max}(A_1^L) < P^{target}$ and $P_E^{max}(A_1^R) < P^{target}$.
2) (Figure 15(b)) Move onto evaluating the combined area network with the parent node. Evaluating $P_E^{max}(A_2) > P^{\text{target}}$, we must choose in a greedy manner via **greedy-tree-action**.
3) (Figure 15(c)) Evaluating both $P_E^{max}(A_3^L)$ and $P_E^{max}(A_3^R)$, where $P_E^{max}(A_3^L) < P_E^{max}(A_3^R) < P^{target}$. The greedy choice will keep A_3^L and discard the A_3^R.
4) (Figure 15(d)) The optimal choice is in fact A_4^R since $P_E^{max}(A_4^R) < P^{target} < P_E^{max}(A_4^R)$.

The greedy choice will choose A_3^L and be forced to place a sensor in A_4^L, while the optimal can continue upstream. The following numerical example will lead to this: $\mu_i = 1, \forall i$ and $\sigma^2 = \{0.0599,\ 0.0125,\ 0.0835,\ 0.0945,\ 0.0906,\ 0.0607\}$ with $P^{target} = 0.1923$.

The triplet of scalar hypotheses which cause this are shown in Table III.

TABLE III
PNNL Test feeders used in case study

Area	Flow	$\mathcal{H}^+(A,b)$	μ_k	σ_k^2	$P_E^{max}(A)$
A_3^L	$s_2 > 0,$ $s_5 > 0$	$e_3, e_4, \emptyset$	1, 2, 3	0.0125, 0.1031, 0.1637	0.1083
A_3^R	$s_2 > 0,$ $s_3 > 0$	$e_5, e_6, , \emptyset$	1, 2, 3	0.0125, 0.0960, 0.1905	0.0961
A_4^L	$s_1 > 0,$ $s_3 > 0$	$e_3, e_4, \emptyset$	2, 3, 4	0.0724, 0.1558, 0.2504	0.1885
A_4^R	$s_1 > 0,$ $s_3 > 0$	$e_5, e_6, , \emptyset$	2, 3, 4	0.0724, 0.1630, 0.2236	0.1960



(a)

Fig. 16. The maximum probability of error over three hypotheses as a function of a translation of each hypothesis variance. At the point $\Delta = 0$, $P(A_3^L) < P(A_3^R)$, and in point $\Delta = 0.0599$, $P(A_4^L) > P(A_4^R)$

Notice that in the hypothesis means and variances, we have the following:

$$\begin{aligned} \mu_k(A_4^L) &= \mu(v_1) + \mu_k(A_3^L) \quad \sigma^2(A_4^R) = \Delta + \sigma_k(A_3^R) \\ \mu_k(A_4^R) &= \mu(v_1) + \mu_k(A_3^R) \quad \sigma^2(A_4^L) = \Delta + \sigma_k(A_3^L) \end{aligned}$$

Where $\mu(v_1) = 1$, and $\Delta = \sigma^2(v_1) = 0.0599$. The translation of mean and variance causes the maximum error over an area to switch from A^L to A^R. Recall that conjecture 1 stated that the maximum error in such a case will monotonically increase with respect to some translation in variances. Regardless, there is no domination between any pair of triplets, whereby one will always be greater than another under the same translation.

This can be seen in Figure 16(a) where the maximum hypothesis error between the tuples is shown with respect to translation Δ. As indicated, any $\Delta > 0.02$ will cause their ordering to change, with the counterexample shown to be beyond this point.

However, we should not that although the transition does occur, the gap is quiet small so any realistic gap will be very small. Since P^{target} is between $P(A_4^L)$ and $P(A_4^R)$ it is very unlikely to occur. In finding a counterexample 10000 monte carlo runs produced 5 examples. For this reason, the greedy and optimal placement strategies will have identical outcomes of random instances.