# Graph-based Multiple-Line Outage Identification in Power Transmission Systems

Boyu Wang, Chao Gan, Jing Yang
School of EECS
The Pennsylvania State University
University Park, PA 16802
{boyuwang,cug203,yangjing}@psu.edu

Chinmay Hedge
Department of ECE
Iowa State University
Ames, IA 50011
chinmay@iastate.edu

Jingxian Wu
Department of EE
University of Arkansas
Fayetteville, AR 72701
wuj@uark.edu

***Abstract*—Fast and accurate detection and identification of power line outage is of paramount importance for the prevention of cascading failures in power systems, as well as prompt and effective restoration following the outage. Traditional approaches can only detect single and double line outages due to the combinatorial complexity challenges involved in the algorithms. A novel approach is to cast it to a sparse vector estimation problem, which can be solved efficiently by taking advantage of the recent progress in compressive sensing and variable selection. In this work, we adopt a similar approach to formulate the problem as a sparse binary-valued vector estimation problem, and leverage the cluster structure existing in most multiple-line outages to solve it. We propose two low-complexity graph-based algorithms to identify clustered line-outages. Simulated tests in IEEE-118 bus system confirm that the proposed algorithms can significantly improve the accuracy and efficiency of baseline algorithms that do not leverage the cluster structure of multiple-line outages.**

***Index Terms*—Line outage identification, cascading failures, structured sparse recovery.**

## I. Introduction

Fast and accurate power line outage identification is critical for the prevention of cascading failures, real-time contingency analysis, and prompt and effective restoration of the power system. A plethora of power line outage identification methods have been proposed in the literature. In [1], [2], Tate and Overbye utilize the pre-outage network topology information and real-time phasor measurements to detect single and double line outages. In [3], Abdelaziz et al. introduce the support vector machine (SVM) technique from machine learning to detect single line outages. A mixed-integer programming approach is proposed by Emami and Abur in [4] to deal with single line outages. Essentially, these methods formulate line outage identification as a combinatorial problem, and solve it by exhaustively searching all possible post-outage topologies. Due to the exponentially increasing complexity, such methods can only handle single or, at most, double line outages. The urge to cope with multiple line outages has motivated several existing works [5]–[8]. In [5], [6], the authors consider the calculation of line outage distribution factors in multiple-line outages case. In [8], He et al. adopts a Gaussian graphical model based approach to identify multiple-line outages at an affordable complexity, under the assumption that phasor angle measurements are conditionally independent with each other. In [7], Zhu and Giannakis formulate the power line outage identification problem as a sparse vector recovery problem, which is then solved efficiently by taking advantage of the recent progress in compressive sensing and variable selection. This approach has inspired a series of works, such as [9]–[11].

In this paper, we study the multiple line outage identification problem by taking the cluster structural feature of line outages into consideration. According to the outage records provided in [12], when multiple power line outages occur simultaneously, they tend to cluster in a local neighborhood. This can be intuitively explained in two ways. First, environment factors, such as lightning, wind, and snow, usually affect the transmission lines locally; Second, when outage occurs on certain transmission lines, the lines in conjunction with them usually have to bear more disturbance when power flow is automatically redistributed, thus are more susceptible to subsequent failures. In order to exploit this feature for more efficient and accurate line outage identification, we first formulate the problem as a sparse *binary*-valued vector estimation problem. Then, we propose two graph-based algorithms with reasonable computational complexity to solve it. One of the algorithm is a graph-based Orthogonal Matching Pursuit (OMP) algorithm, and the other is a graph-based Compressive Sampling Matching Pursuit (CoSaMP). Both of the algorithms are analyzed and evaluated through simulated tests on IEEE-118 bus systems. Simulation results confirm that the graph-based approaches is more efficient than baseline algorithms that do not exploit the cluster structure of multiple-line outages.

We adopt the following set of notations. We use bold-face lower-case, e.g., $\mathbf{x}$, and bold-face upper-case, e.g., $\mathbf{X}$, to denote vectors and matrices, respectively. $\|\mathbf{x}\|_p$ denotes the $\ell_p$-norm of $\mathbf{x}$. Sets and events are denoted with calligraphic font (e.g., $\mathcal{T}$). The cardinality of a finite set $\mathcal{T}$ is denoted as $|\mathcal{T}|$.

## II. System Model and Problem Formulation

We consider a power transmission network consisting of $N$ buses and $L$ lines. We use a graph $G = (\mathcal{N}, \mathcal{L})$ to represent it, where the node set $\mathcal{N} := \{1, ..., N\}$ denotes the buses and the edge set $\mathcal{L} := \{(m, n)\} \subset \mathcal{N} \times \mathcal{N}$ represents the lines. The system monitors the voltage phasor angles and injected real

This work was supported in part by the U. S. National Science Foundation (NSF) under Grants ECCS-1405403, ECCS-1650299 and CCF-1566281.

978-1-5386-2212-4/17/$31.00 ©2017 IEEE

power of all buses. Denote $p_m$ as the real power injection on bus $m$, $p_{mn}$ as the real power flow from bus $m$ to bus $n$, $\theta_m$ as the voltage phasor angle of bus $m$, $x_{mn}$ as the reactance of the line between bus $m$ and $n$, which is equal to $x_{nm}$. Then under the DC power flow model, the instant power flow must satisfy [13]:

$$p_m = \sum_{n \in \mathcal{N}_m} p_{mn} = \sum_{n \in \mathcal{N}_m} \frac{\theta_m - \theta_n}{x_{mn}} \tag{1}$$

where $\mathcal{N}_m := \{m : (m,n) \in \mathcal{L}\}$, i.e., the set of buses connected with bus $m$ in the graph $G$. Eqn. (1) can be equivalently expressed as

$$\mathbf{p} = \mathbf{B}\boldsymbol{\theta} \tag{2}$$

where $\boldsymbol{\theta} \triangleq [\theta_1, \theta_2 \ldots, \theta_N]^T \in \mathsf{R}^N$, $\mathbf{p} \triangleq [p_1, p_2 \ldots, p_N]^T \in \mathsf{R}^N$, $\mathbf{B}$ is a $N \times N$ matrix whose $(m,n)$-th entry is defined as

$$\mathbf{B}_{mn} = \begin{cases} -1/x_{mn} & \text{if } (m,n) \in \mathcal{L} \\ \Sigma_{v \in \mathcal{N}_m} 1/x_{mv} & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We note that $\mathbf{B}$ is uniquely determined by the topology and line reactance parameters of the power network. Each transmission line $(m,n) \in \mathcal{L}$ is related to four non-zero entries in $\mathbf{B}$, namely, $\mathbf{B}_{mn}$, $\mathbf{B}_{nm}$, $\mathbf{B}_{mm}$, and $\mathbf{B}_{nn}$.

When line outages happen in the system, the outaged lines are removed from the transmission network, resulting in a different matrix $\mathbf{B}'$. We assume that the power system works at a quasi-stable state during the events, i.e, the change of power injection $\mathbf{p}$ and load profile between the pre-event and post-event can be neglected under the timescale of outage events [1]. We also assume that the outage does not cause any islanding in the power grid. As a result of the line outage, voltage phasor angles will change automatically as the injected power is redistributed across the network to achieve another power flow balance. Denoting $\boldsymbol{\theta}' \triangleq [\theta'_1, \theta'_2 \ldots, \theta'_N]^T \in \mathsf{R}^N$ as the post-event phase angle measurements. Then, we have

$$\mathbf{B}\boldsymbol{\theta} = \mathbf{B}'\boldsymbol{\theta}' + \tag{4}$$

where, $\in \mathsf{R}^N$ consists of $N$ i.i.d Gaussian random variables with zero mean and $\sigma^2$ variance. We use it to account for the small perturbations between the pre- and post-event injected powers. Our objective is to identify $\mathbf{B}'$ based on the knowledge of $\mathbf{B}$ and measurements $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, and unveil the the outage locations based on the difference between $\mathbf{B}'$ and $\mathbf{B}$.

We follow the approach in [7] to reformulate (4) as a sparse vector recovery problem by representing matrix $\mathbf{B}$ in the form of a weighted Laplacian matrix. Denote $l$ as the index of line $(m,n)$, $\mathbf{m}_l \in \mathsf{R}^N$ as a column vector with all the entries equal to 0 except the $m$-th and $n$-th elements, which are 1 and $-1$, respectively. Define $\mathbf{M} \triangleq [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_L] \in \mathsf{R}^{N \times L}$, $\mathbf{D} \triangleq \text{diag}\{\frac{1}{x_1}, \frac{1}{x_2}, \ldots, \frac{1}{x_L}\} \in \mathsf{R}^{L \times L}$. Then, we have

$$\mathbf{B} = \sum_{l=1}^{L} \frac{1}{x_l} \mathbf{m}_l \mathbf{m}_l^T = \mathbf{M}\mathbf{D}\mathbf{M}^T \tag{5}$$

Substituting (5) into (4), we have

$$\mathbf{B}\Delta\boldsymbol{\theta} = (\mathbf{B} - \mathbf{B}')\boldsymbol{\theta}' + \tag{6}$$

$$= \sum_{l \in \mathcal{L}'} \frac{1}{x_l} \mathbf{m}_l \mathbf{m}_l^T \boldsymbol{\theta}' + \tag{7}$$

$$= \sum_{l \in \mathcal{L}'} \frac{\mathbf{m}_l^T \boldsymbol{\theta}'}{x_l} \mathbf{m}_l + \tag{8}$$

where $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}' - \boldsymbol{\theta}$, $\mathcal{L}'$ is the set of outaged lines. Eqn. (7) follows from the fact that, with the weighted Laplacian representation in (5), $\mathbf{B} - \mathbf{B}'$ can be expressed as the weighted summation of the matrices related to the outaged lines only.

Define $\mathbf{A} \triangleq [\frac{\mathbf{m}_1^T \boldsymbol{\theta}'}{x_1}\mathbf{m}_1, \frac{\mathbf{m}_2^T \boldsymbol{\theta}'}{x_2}\mathbf{m}_2, \ldots, \frac{\mathbf{m}_L^T \boldsymbol{\theta}'}{x_L}\mathbf{m}_L] \in \mathsf{R}^{N \times L}$, $\mathbf{y} \triangleq \mathbf{B}\Delta\boldsymbol{\theta}$, $\boldsymbol{\beta} \triangleq [\beta_1, \beta_2 \ldots, \beta_L]^T \in \mathsf{R}^L$, where

$$\beta_l = \begin{cases} 1 & \text{if } l \in \mathcal{L}' \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Then, (8) can be equivalently expressed as

$$\mathbf{y} = \mathbf{A}\boldsymbol{\beta} + \tag{10}$$

where $\mathbf{y}$ is the observation vector, which can be obtained based on $\mathbf{B}$ and the pre- and post-event phasor angle measurements, $\mathbf{A}$ is the measurement matrix, and $\boldsymbol{\beta}$ is a *binary*-valued vector to recover.

Assume that the number of the outaged lines is upper bounded by $s$, which is a small number compared with $L$. Then, the line outage identification problem can be formulated as a sparse recovery problem as follows:

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta} \in \{0,1\}^L, \|\boldsymbol{\beta}\|_1 \le s} \|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2 \tag{11}$$

Essentially, (11) is an integer programming problem, and its complexity scales exponentially in $s$. To overcome the combinatorial complexity, in the following, we leverage the recent progress on sparse recovery and the power network topological information to develop two algorithms to solve (11) efficiently.

## III. Graph-based Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) is a widely adopted iterative greedy algorithm for sparse recovery. The algorithm works as follows: In each iteration, it greedily selects the column which is most correlated with the current residual vector from the unselected columns of the measurement matrix. This column is then added into the set of selected columns. The algorithm then updates the residual vector by projecting the observation vector onto the linear subspace spanned by the selected columns. The algorithm iterates until the residual error is within a desired tolerance level. Compared with other alternative methods, a major advantage of the OMP is its simplicity and fast implementation.

In this section, we modify the OMP algorithm by incorporating the binary constraint on $\boldsymbol{\beta}$, and the topology information of the power system. We add graph $G$ as an input to the algorithm. In each iteration, we keep track of two sets of lines, $\mathcal{S}^k$ and $\mathcal{L}^k$. $\mathcal{S}^k$ is used to track the set of candidate columns that would be considered in the step of greedy selection, with its initial value set to be the full set including all columns of $\mathbf{A}$. The estimate outage set $\mathcal{L}^k$ is used to track the set of identified outaged lines, with its initial value to be empty. The algorithm works as follows. At the beginning of the $k$-th iteration, we greedily select the column from $\mathcal{S}^{k-1}$ to minimize the $\ell_2$ norm of the residual vector, and update the estimate outage set $\mathcal{L}^k$ by adding the new selected column index to $\mathcal{L}^{k-1}$. We point out that this index also correspond to a line in graph $G$. Next, we take the binary value constraint on $\boldsymbol{\beta}$ into consideration. We perform a Least Square Estimation (LSE) by restricting the values of $\beta_i, \forall i \in \mathcal{L}^k$ to be in $[0, 1]$ and letting the entries outside $\mathcal{L}^k$ to be zero. After obtaining $\boldsymbol{\beta}^k$, we force all of its entries to be binary through thresholding. Next, we update the estimate outage set $\mathcal{L}^k$ again by including the support of the thresholded $\boldsymbol{\beta}^k$ only. The residual vector $\mathbf{r}^k$ is updated accordingly. If $\|\mathbf{r}^k\|_2$ is still above the tolerance level, the algorithm will take the power network topology into consideration. Since multiple line outages often happen in clusters, if $\mathcal{L}^k$ correctly identifies some of the outaged lines, then, intuitively, the rest of the outaged lines will be in close proximity to those included in $\mathcal{L}^k$. This motivates us to update $\mathcal{L}^k$ by including the lines that share buses with those included in $\mathcal{L}^k$ only. The algorithm then move to the next iteration until a desired tolerance level is achieved, or until it reaches the maximum number of iterations. We summarize the algorithm in Algorithm 1.

**Algorithm 1** (Graph OMP) Input: $\mathbf{y}$, $\mathbf{A}$, graph $G$, threshold $\tau$. Output $\boldsymbol{\beta}$.

```
Initialize: r^0 = y, β^0 = 0, S^0 = L, L^0 = ∅, k = 1.
while k ≤ k_max or ||r^k||_2 ≤ δ do
    l^k := arg min_{l ∈ S^{k-1}} ||r^{k-1} − A(:, j)||_2^2
    L^k := L^{k-1} ∪ {l^k}
    Estimate β^k := arg min ||r^{k-1} − Aβ||_2^2 s.t. β_i ∈ [0, 1]
  for i ∈ L^k, and β_i = 0, ∀i ∉ L^k
    Update L^k := {l : β_l^k ≥ τ}
    Update residual vector r^k := r^{k-1} − Σ_{j ∈ L^k} A(:, j)
    if ||r^k||_2 ≤ δ then return
    else
        Update S^k by adding all lines sharing buses with
  the lines included in L^k;
        k = k + 1;
    end if
end while
```

## IV. Graph-based COSAMP

CoSaMP is an algorithm based on OMP. It uses an approach inspired by the restricted isometry property (RIP) [14] to accelerate the algorithm and to provide strong guarantees that OMP cannot. Specifically, if the measurement matrix $\mathbf{A}$ satisfies the RIP, the vector $\mathbf{A}^*\mathbf{A}\boldsymbol{\beta}$ can serve as a proxy for the sparse signal $\boldsymbol{\beta}$, since the largest components of the proxy point to the largest components of the original signal $\boldsymbol{\beta}$ with high probability. In each iteration of CoSaMP, it first identifies the largest components from the proxy of the residual, merges them into the set of selected component, and then solves a least-squares problem to approximate the target signal on the merged set of components. The algorithm then performs a pruning step to produce a new approximation by retaining only the largest entries in the least-squares approximation. The residual is then updated and fed to the next iteration until a stopping condition is satisfied.

In [15], Hegde et. al propose a Graph-based CoSaMP algorithm to account for sparsity structures defined via graphs. They introduce a novel general sparsity model called weighted graph model (WGM). Specifically, they define $G = (\mathcal{V}, \mathcal{E})$ as an undirected, weighted graph where the node set $\mathcal{V}$ corresponds to the unknown sparse vector $\boldsymbol{\beta}$, and each edge has a real-valued weight. In order to control the sparsity patterns, the WGM offers three parameters on the desired supports $\mathcal{S} \subseteq \mathcal{V}$, i.e., the sparsity level $s$; the maximum number of connected components (trees) corresponding to $\mathcal{S}$, denoted as $g$; and the bound on the total weight of edges in the forest corresponding to $\mathcal{S}$, denoted as $B$. By choosing different edge weights and those three parameters, different sparsity structures can be encoded with the WGM. Essentially, the WGM captures sparsity structures with a small number of trees in $G$.

The algorithmic core of the graph-CoSaMP algorithm is a computationally efficient procedure to project an arbitrary vector into the WGM. Mathematically, for any given vector $\boldsymbol{\beta}$, the algorithm looks for the best vector $\boldsymbol{\beta}'$ whose support is compliant with the sparsity structure encoded by the WGM to minimize $\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|^2$. However, searching for the best $\boldsymbol{\beta}'$ is a NP-hard combinatorial optimization problem [16]. In order to reduce the computational complexity, the framework propose two approximate model-projection algorithms, namely tail approximation and head approximation, to obtain a sub-optimal $\boldsymbol{\beta}'$ with complementary approximation guarantees. The approximation algorithms are based on a connection to the prize-collecting Steiner Forest problem (PCSF).

For the completeness of this paper, in the following, we will first introduce the PCSF problem, and then discuss how our problem can be related to PCSF.

### *A. The Prize Collecting Steiner Forest (PCSF) Problem*

PCSF is a variant of the Prize Collecting Steiner Tree (PCST) problem. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected, weighted graph with edge costs $c(e) \in \mathsf{R}, \forall e \in \mathcal{E}$ and node prizes $\pi(v) \in \mathsf{R}, \forall v \in \mathcal{V}$. For a subset of edges $\mathcal{E}' \subset \mathcal{E}$, its cost $c(\mathcal{E}') = \sum_{e \in \mathcal{E}'} c(e)$. Similarly, for a subset of nodes $\mathcal{V}' \subset \mathcal{V}$, its prize $\pi(\mathcal{V}') = \sum_{v \in \mathcal{V}'} \pi(v)$. We also denote $\bar{\mathcal{V}}'$ as the complement of $\mathcal{V}'$. Then the goal of the PCST problem is to find a subtree $\mathcal{T} = (\mathcal{V}', \mathcal{E}')$ to minimize $c(\mathcal{T}) + \pi(\bar{\mathcal{T}})$.

This problem is a generalization of the classical Steiner tree problem [17], and it is also NP-hard. In the seminal work of Goemans and Williamson (GW) [18], a primal-dual algorithm is developed with certain approximation guarantee.

The objective of PCSF is to find a subgraph $\mathcal{F} = (\mathcal{V}', \mathcal{E}')$ that minimizes $c(\mathcal{F}) + \pi(\bar{\mathcal{F}})$ with at most $g$ connected compo-

978-1-5386-2212-4/17/$31.00 ©2017 IEEE

nents (trees) in $\mathcal{F}$. In [15], a fast GW algorithm is proposed, which returns a forest with the following guarantee:

$$c(\mathcal{F}) + 2\pi(\bar{\mathcal{F}}) \leq 2c(\mathcal{F}_{OPT}) + 2\pi(\bar{\mathcal{F}}_{OPT}) \qquad (12)$$

where $\mathcal{F}_{OPT}$ is the optimal solution of the PCSF problem.

By setting $\pi(i) = \beta_i^2$ and $c(e) = w(e) + 1$, we have $c(\mathcal{F}) = w(\mathcal{F}) + (|\mathcal{F}| - g)$, $\pi(\bar{\mathcal{F}}) = \|\boldsymbol{\beta} - \boldsymbol{\beta}_{\mathcal{F}}\|^2$. After multiplying the edge costs with a parameter $\lambda$, the PCSF objective function essentially becomes a Lagrangian relaxation of the WGM-constrained optimal projection problem. The tail and head approximation algorithms in [15] adaptively search for the optimal $\lambda$ to ensure that the PCSF-GW algorithm returns a good approximation of $\boldsymbol{\beta}$ with theoretical guarantees.

### B. Graph CoSaMP Algorithm

In this subsection, we adapt the Graph CoSaMP proposed in [15] for the multiple-line outage identification problem. For the sparse recovery problem in (11), the unknown vector $\boldsymbol{\beta}$ represents the statuses of power lines. The corresponding WGM graph $M = (\mathcal{V}, \mathcal{E})$ is defined as follows. First, we define the node set $\mathcal{V}$ as the set of the power lines. If two power lines are connected through a bus in the power grid, there is an edge connecting the corresponding nodes in $M$. We let the node prize $\pi(i) = \beta_i^2$, and the edge cost $c(e) = c$. As we will explain later, we fix $c$ be to a constant in $(0.5, 1)$.

The adapted Graph CoSaMP is summarized in Algorithm 2. Specifically, after forming a proxy of the residue vector, we adopt the tail approximation algorithm in [15] to find an approximation of the proxy with the desired sparse forest structure. After we merge the support of the returned forest with the support from previous step, we update the estimation of the signal vector by performing LSE over the support under the constraint that $\beta_i \in [0, 1]$, and then threshold it to make it binary. In the pruning step, we use PCSF-GW algorithm to get an approximated signal vector whose support has the desired forest structure.

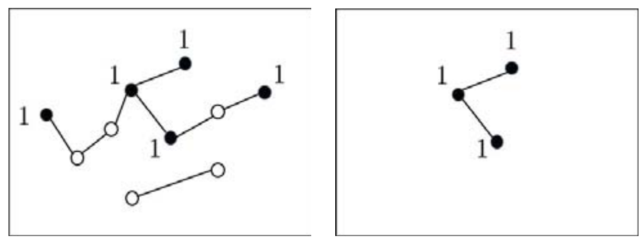
The input of PCSF-GW comes from the thresholded LSE result over the support $\mathcal{Z}^k$. Therefore, the node prizes can only be zero or one, as shown in Fig. 1(a). We use black dots to represent the nodes with prize one. Consider the case where $g$ is set to be one. If the edge cost is greater than one, PCSF-GW will select only one node with prize one to optimize the objective function. On the other hand, if the edge cost is very small, all nodes with prize one will be connected by edges to form one cluster. In order to obtain the desired cluster structure shown in Fig. 1(b), we choose a constant edge cost $c \in (0.5, 1)$.

## V. Simulation Results

In this section, we compare our proposed algorithms with baseline algorithms OMP and CoSaMP on an IEEE 118-Bus system. The parameters of this system are fully described in the toolbox of MATPOWER [19]. We simulate the line outages by eliminating some transmission lines in the power network. We collect pre- and post-event measurements, namely, the

**Algorithm 2** (Graph CoSaMP) Input: $\mathbf{y}$, $\mathbf{A}$, $s$, $g$, $M$, $\tau$. Output: $\boldsymbol{\beta}$.

Initialize: $\mathbf{r}^0 = \mathbf{y}, \boldsymbol{\beta}^0 = \mathbf{0}, \mathcal{S}^0 = \mathcal{L}$, $\mathcal{L}^0 = \emptyset$, $k = 1$.
**while** Stopping criterion is not met **do**
  Form a proxy $\mathbf{t} = \mathbf{A}^{\square} \times \mathbf{r}^{k-1}$
  Identify support $\mathcal{T} = \text{TailApprox}(M, \mathbf{t}, s, g)$
  Merge $\mathcal{Z}^k = \mathcal{T} \cup \text{supp}(\boldsymbol{\beta}^{k-1})$
  Estimate $\boldsymbol{\beta}^k = \arg\min \| \mathbf{b} - \mathbf{A}\boldsymbol{\beta} \|_2$ s.t. $\text{supp}(\boldsymbol{\beta}) \in \mathcal{Z}^k$, $\beta_i \in [0, 1]$
  **if** $\beta_i^k \geq \tau$ **then** $\beta_i^k = 1$
  **else** $\beta_i^k = 0$
  **end if**
  Prune $\boldsymbol{\beta}^k = \text{PCSF-GW}(M, \boldsymbol{\beta}^n, g)$
  $\mathbf{r}^{k+1} = \mathbf{y} - \mathbf{A}\boldsymbol{\beta}^k$
  $k = k + 1$
**end while**
**return** $\boldsymbol{\beta}$



(a) Input of PCSF. (b) Output of PCSF.

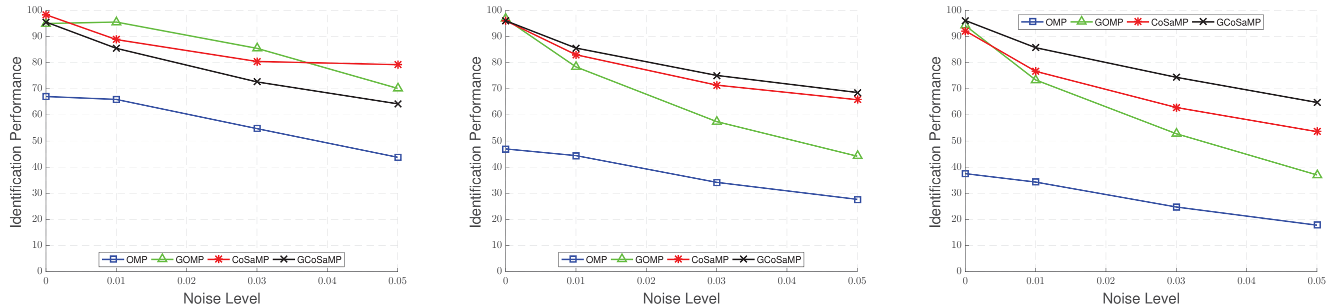
Fig. 1: A pruning example of PCSF with $g = 1$.

voltage phasor angle of every bus after running the power flow function.

We consider the result returned by an algorithm a *success* if it matches with the set of lines removed from the system exactly. We randomly generate the outage patterns, and evaluate the performances of the algorithms by counting their portion of success.

First, we study the performance of the proposed algorithms on identifying a single cluster of line outages. The size of the cluster ranges from one to three lines. For the single line case, all the 179 lines in the system are tested and repeated for 10 times, totally 1790 cases. For the double and triple line clustered outage, we randomly choose 1500 cases. We set the standard deviation of the noise to be 0, 1%, 3%, and 5% of the power injection, and evaluate the performances of the algorithms at those noise levels. The percentage of successful identification is recorded and compared in Fig. 2. As we observe, Graph OMP has significant improvement over OMP for all three scenarios. Graph CoSaMP has better performance in multiple line outage cases over CoSaMP, especially in the three-line outage case.

Then we study the performances of the proposed algorithms with two clusters of line outages. We consider the case where both clusters consist of two outaged lines, and three outaged lines, respectively. The identification performances are shown




(a) Single-line outage. (b) Double-line outage. (c) Three-line outage.

Fig. 2: Outage identification for a single cluster.

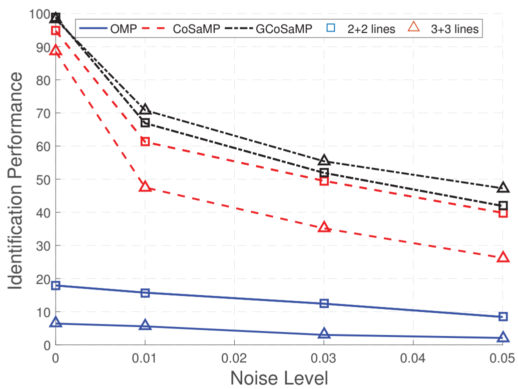



Fig. 3: Outage identification for two clusters.

in Fig. 3. We note that the Graph-CoSaMP has much better performance than OMP and CoSaMP. The results of Graph OMP is not included since it cannot handle multiple clusters.

Finally, we compare the running times of the algorithms. All algorithms are run in MATLABr R2016a, on a MacBook Pro with a 2.2GHz CPU and a 16GB RAM. Table I shows the average running times of the algorithms. We note that the Graph OMP has slight improvement over OMP on the average running time, while Graph CoSaMP takes much more time than other algorithms due to its increased complexity in the graph model based projection in the tail approximation and pruning steps.

TABLE I: Average running times in seconds.

| | OMP | Graph OMP | CoSaMP | Graph CoSaMP |
|---|---|---|---|---|
| 1 line | 2e-3 | 1.8e-3 | 8e-4 | 1.60 |
| 2 lines | 2.4e-3 | 2.3e-3 | 6e-4 | 1.60 |
| 3 lines | 4.1e-3 | 3.9e-3 | 7e-4 | 1.82 |
| 2+2 lines | 3.7e-3 | – | 7e-4 | 1.57 |
| 3+3 lines | 6.4e-3 | – | 9e-4 | 1.75 |

## References


[1] 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.

[2] ——, "Double line outage detection using phasor angle measurements," in *2009 IEEE Power Energy Society General Meeting*, July 2009, pp. 1–5.

[3] A. Y. Abdelaziz, S. F. Mekhamer, M. Ezzat, and E. F. El-Saadany, "Line outage detection using support vector machine (svm) based on the phasor measurement units (pmus) technology," in *IEEE Power and Energy Society General Meeting*, July 2012, pp. 1–8.

[4] R. Emami and A. Abur, "Tracking changes in the external network model," in *North American Power Symposium (NAPS)*, Sept 2010, pp. 1–6.

[5] T. Guler, G. Gross, and M. Liu, "Generalized line outage distribution factors," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 879–881, May 2007.

[6] J. Guo, Y. Fu, Z. Li, and M. Shahidehpour, "Direct calculation of line outage distribution factors," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1633–1634, Aug 2009.

[7] 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.

[8] M. He and J. Zhang, "A dependency graph approach for fault detection and localization towards secure smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 342–351, June 2011.

[9] M. Babakmehr, M. G. Simes, M. B. Wakin, A. A. Durra, and F. Harirchi, "Smart grid topology identification using sparse recovery," in *Industry Applications Society Annual Meeting, 2015 IEEE*, Oct 2015, pp. 1–8.

[10] B. Donmez and A. Abur, "Sparse estimation based external system line outage detection," in *2016 Power Systems Computation Conference (PSCC)*, June 2016, pp. 1–6.

[11] G. Feng and A. Abur, "Fault location using wide-area measurements and sparse estimation," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2938–2945, July 2016.

[12] M. LLC. (2016) Bpa operations information. [Online]. Available: https://transmission.bpa.gov/Business/Operations/Outages/default.aspx

[13] M.-S. Chen and W. E. Dillon, "Power system modeling," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 901–915, 1974.

[14] D. Needell and J. A. Tropp, "Cosamp: iterative signal recovery from incomplete and inaccurate samples," *Communications of the ACM*, vol. 53, no. 12, pp. 93–100, 2010.

[15] C. Hegde, P. Indyk, and L. Schmidt, "A nearly-linear time framework for graph-structured sparsity," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 928–937.

[16] R. M. Karp, "Reducibility among combinatorial problems," in *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.*, 1972, pp. 85–103.

[17] I. Ljubic, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti, "Solving the prize-collecting steiner tree problem to optimality," in *ALENEX/ANALCO*, 2005, pp. 68–76.

[18] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 296–317, 1995.

[19] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, "Matpower: A matlab power system simulation package," *Manual, Power Systems Engineering Research Center, Ithaca NY*, vol. 1, 1997.