Graph Algorithms for Preventing Cascading Failures in Networks

Pei Duo Yu†, Chee Wei Tan† and Hung-Lin Fu*
City University of Hong Kong†, National Chiao Tung University*
peiduoyu2-c@my.cityu.edu.hk, cheewtan@cityu.edu.hk, hlfu@math.nctu.edu.tw

***Abstract*—Cascading failures in critical networked infrastructures that result even from a single source of failure often lead to rapidly widespread outages as witnessed in the 2013 Northeast blackout in northern America. This paper examines the problem of minimizing the outage when a cascading failure from a single source occurs. An optimization problem is formulated where a limited number of protection nodes, when placed strategically in the network to mitigate systemic risk, can minimize the spread of cascading failure. Computationally fast distributed message-passing algorithms are developed to solve this problem. Global convergence and the optimality of the algorithm are proved using graph theoretic analysis. In particular, we illustrate how the poset-constrained graph algorithms can be designed to address the trade-off between complexity and optimality.**

***Index Terms*—Cascading failure, viral spreading, graph theory, large-scale optimization, message-passing algorithms**

I. Introduction

WHEN a cascading failure such as a widespread power outage [1] occurs, how to quickly contain the spread (even without knowing the source of failure) is of the essence, because this relates to how quickly a network can recover from temporal malfunction. Critical networked infrastructures such as the power grid and the Internet may have to evolve as networks that in addition to performing a basic homogeneous functionality (such as routing or forwarding) also have so-called *protection nodes* that can proactively protect the entire network from cascading spread that have high systemic risks. An example is the installation of special-purpose power circuit breakers that act as border gateways between subgrids that automatically decouple problematic grids from working ones in order to prevent the power outage from spreading. Conceivably, the protection cost comes at a high price and therefore such protection nodes when deployed ought to be limited in numbers. Hence, the problem to avert cascading failures by installing protection nodes in the network to mitigate systemic risk (i.e., finding the few protection nodes to guard against any impending cascades) is somewhat intertwined with the problem of cascading failure source detection (i.e., finding the most probable source after the cascade has happened).

The key to placing protection nodes to mitigate systemic risk is to identify the critical topological structure of the network related to the most probable source of failure given a snapshot observation of the failed nodes in the network. This snapshot observation of a cascade may be actual incidents in the past or it can even be a hypothetical one based on forecasting or simulations. Understanding how a cascade in complex networks evolves over time has received considerable attraction in the literature [1]–[4]. On the other hand, the problem of finding the source of a cascading is fairly recent and open [5]–[13]. We study the dual problem of predicting where to place a number of protection nodes in order to minimize the outage of cascading failure. We define outage of a cascading failure as the maximum number of nodes that can be adversely affected when the spread occurs. Protection nodes serve to stop the cascading failure from reaching other susceptible nodes when the spread reaches the protection nodes.

Indeed, protection nodes in the graph are ideally nodes that have *gateway-like* features, i.e., they can detect or even contain the cascading failure at critical subgraph junctions. This is similar to the placement of safeguards to detect the presence of or determining the exact location of an intruder in a network (so-called fault-tolerant locating-dominating sets, e.g., see [14] and the references therein), but these prior work do not consider safeguarding any cascading phenomenon. Due to systems considerations, protection cost is premium and this necessitates a limited number of protection nodes. This problem becomes computationally hard to solve when the size of the graph scales up and is the topic of this paper. We first provide network topological insights for predicting the most probably cascading failure source, that in turn leads naturally to feasible protection node placement strategies against cascading failures.

The main contributions of this paper are as follows:

- We introduce the notion of a partially ordered set (poset) as a means of modeling causality for inference problems with cascading failures and we exploit poset-constrained topological properties to derive equivalence relationship with the graph-theoretic centroid.
- We formulate an optimization problem of protection node placement in networks that have tree-graph topologies to mitigate the systemic risks of cascading failures and show that the graph centroid characterization is a feasible solution, and when it is optimal when we consider a single protection node. In the general case, by leveraging centroid decomposition, we propose a computationally

This work was in part supported by the Hong Kong Innovation and Technology Fund (ITF) Project (ITS/180/16), in part by the Science, Technology and Innovation Commission of Shenzhen Municipality, under Project JCYJ20170307090810981, and in part by the National Natural Science Foundation of China (NSFC), Grant No: 61771018.

978-1-5386-0579-0/18/$31.00 ©2018 IEEE

TABLE I
TABLE OF NOTATION

Notation	Remark
G_N	Underlying Network (consists of all susceptible vertices)
G_n	Subgraph of G_N of n vertices affected by the cascading failure
V_P	The set of vertices selected to be protected
$C(\{V_P\})$	Sequence of connected components after removing all vertices in V_P from G_N
v_c	*Centroid* of G_n
d_v	Degree of vertex v
$d(u,v)$	Distance between vertices v and u
$P(\hat{v}\|G_n)$	Probability that $\hat{v} = v^\star$, i.e., correct detection probability

efficient algorithm that can be parameterized recursively based on the branch weight centrality to place protection nodes to mitigate the systemic risk of cascading failures.

II. Modeling Cascading Failure using Posets

In this section, we model the occurrence of cascading failures in a networked infrastructure, e.g., a power grid network, using an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \dots\}$ is a set of nodes and E is the set of edges of the form (i, j) for vertices v_i and v_j in V. In other words, the nodes performing the basic functionality in this networked infrastructure are the vertices in G, and the edges model the conduit for interaction between nodes. For example, two substations in a power grid are connected by an edge so that power flows from one to the other. Another example is the Internet network where the nodes model routers that forward data packets in a hop-by-hop fashion. As such, G is general enough to model complex interaction in most man-made networked infrastructures. The degree of a node v_i is the number of its neighbors denoted by d_i.

To model the cascading failures for general networked infrastructures, we assume a basic epidemic model known as the susceptible-infectious (SI) model (e.g., see [15]) to model a cascading failure in a network. In this model, there are two types of nodes: (i) susceptible vertices that are susceptible to failure; and (ii) infected nodes that can cause their immediate susceptible neighbors to fail, e.g., v_i fails and in turn may cause v_j to fail when $(i, j) \in E$. Once a susceptible vertex fails, it remains in that failure state perpetually. In this way, spreading occurs in a cascading manner. We also assume a memoryless property in this spreading model: let τ_{ij} be the spreading time for an infected node v_i to infect its susceptible neighbor v_j for all $(i, j) \in E$, then τ_{ij}'s are mutually independent and exponentially distributed with a parameter λ (assume $\lambda = 1$). This SI model has also been used to analyze other kinds of viral spreading problems in the literature [6], [7], [9], [10], [12].

A. Preliminaries of Linear Extensions of Posets

Suppose that a single source of failure in this network G_N originates from a vertex $v^\star \in V(G_N)$ at a certain time $t = 0$ and spreads in the network G_N. Then, at a later time $t = T$, we observe that there are n failed nodes in the network G_N, and these n nodes collectively constitute a spread graph that we denote by G_n. Note that n represents the cardinality of the set of failed vertices in G_n.

Given only the spread graph G_n, the question of interests is to find out which node is the single source of cascading failure. Without observing how the cascade unravels, this is a computationally hard problem as the problem is mathematically equivalent to finding the sequence of nodes that fail in G_n. In the following, we connect the deduction of cascading failure to counting linear extensions of a given poset. Posets (Partially Ordered Sets) and its linear extensions are well-studied objects in order theory.

Definition II.1. A non-strict partial order is a relation $\leq_S$ over a set S satisfying the following rules, for all $v_1, v_2, v_3 \in S$:

- $v_1 \leq_S v_1$ (reflexivity)
- if $v_1 \leq_S v_2$ and $v_2 \leq_S v_1$, then $v_1 = v_2$ (antisymmetry)
- if $v_1 \leq_S v_2$ and $v_2 \leq_S v_3$, then $v_1 \leq_S v_3$ (transitivity)

A **total order** has one more rule that every two elements in the set must be assigned a relation. A **linear extension** $\leq_S^*$ of a partial order $\leq_S$ is a total order which preserve the relation in in $\leq_S$, i.e., for all $v_1 \leq_S^* v_2$ whenever $v_1 \leq_S v_2$. Given a network G_n and randomly pick a vertex v^* as the source of G_n, then there exists a partially order on $V(G_n)$. For example, let G_4 be a tree and v_2 is v^* . The following graph shows how a tree can be viewed as a partially ordered set [16].

Now, consider a cascading failure with the order $\sigma = (v_2, v_3, v_1, v_4)$ starting from the node v_2 resulting in four failed nodes in the network, then σ can be viewed as a total order on $V(G_4)$ with the order $v_4 \leq_{V(G_4)}^* v_1 \leq_{V(G_4)}^* v_3 \leq_{V(G_4)}^* v_2$. Note that this total order $\leq_{V(G_4)}^*$ preserve the relation in $\leq_{V(G_4)}$, for example, if $v_1 \leq_{V(G_4)} v_2$ then we have $v_1 \leq_{V(G_4)}^* v_2$ by the transitivity of the total order, thus, this total order is a linear extension of this poset. According to the definition of the linear extension, we can conclude that a cascading failure order starting from v over a graph G_n can be viewed as one of the linear extensions of the poset constructed by the tree G_n rooted at v, moreover, the number of these orders starting from v over a graph G_n is equivalent to the number of linear extensions on the tree G_n rooted at v. When G_N is a general graph, the cascading failure is, in fact, a spanning tree of G_n, and we can compute the number of linear extensions on this spanning tree. The uncertainty is which spanning tree, i.e., diffusion tree? Intuitively, to protect a vertex with the maximum number of linear extensions (i.e., the number of possible sequences of vertices being affected by failure as the cascading failure spreads) on the given network is a good choice.

III. Network Centrality to Determine Maximum Number of Linear Extensions of a Poset

In this section, we provide an analytical characterization of the graph centroid as the node with the maximum number

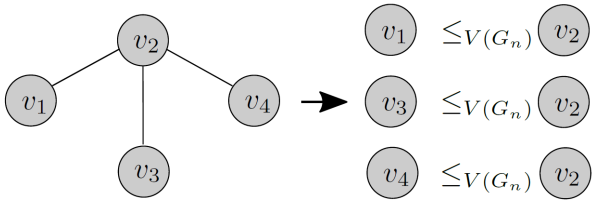


Fig. 1. An example of change a rooted tree structure to a partially ordered set. Note that, there is no relation between v_1, v_3 and v_4, since this order is a **partial** order. However, when considering a cascading failure on this graph with a specific order, for example $v_2 \to v_1 \to v_3 \to v_4$, then there is relation between any two vertices in this set.

of linear extensions of a poset, and propose a new message-passing algorithm to compute the centroid. This equivalence characterization also implies that the graph centroid is the most probable source of cascading failure, and forms the basis to tackle the protection node placement problem in Section IV.

A. Centroid as Network Center

Definition III.1. Let G_n be a tree with n vertices, for any $u, v \in G_n$, let t_v^u be the subtree rooted at v by removing the edge (u, v) from G_n and slightly abusing the notation of the subtree size t_v^u as t_v^u.

Now, we describe how to compute the number of linear extensions on a given poset. Since a linear extensions on a given poset is equivalent to a spreading order, we can leverage the work in [12] to compute the number of linear extensions on a rooted tree:

$$L(v, G_n) = \frac{(n-1)!}{t_{u_1}^v ! \cdot t_{u_2}^v ! \cdot \ldots \cdot t_{u_{d(v)}}^v !} \cdot \prod_{i=1}^{d_v} L(u_i, T_{u_i}^v). \quad (1)$$

In (1), $L(v, G_n)$ is called the **rumor centrality** of v in G_n, which is the number of spreading orders starting from v. For simplicity, in this paper, we also denote the number of linear extensions on the tree G_n rooted at v as $L(v, G_n)$. The vertex with the maximum rumor centrality is called the **rumor center**. The rumor centrality can be expanded recursively from the root v_r to all the leaves of G_n to yield [12]:

$$L(v, G_n) = n! \cdot \prod_{u \in G_n} \frac{1}{t_u^v}. \quad (2)$$

Using (2), we can find the rumor center, i.e., the vertex with maximum number of linear extensions. Now, consider two adjacent vertices u and v in G_n and a vertex $w \in G_n - \{u, v\}$, then we have $t_u^v = n - t_v^u$ and $t_w^v = t_w^u$. By using this recursion, it can be established that:

$$\frac{L(u, G_n)}{L(v, G_n)} = \frac{t_u^v}{n - t_v^u}, \quad (3)$$

which leads to the following result (see Proposition 1 in [12]).

Theorem 1. *Given a tree G_n with n vertices, $v \in G_n$ is a rumor center if and only if*

$$t_u^v \leq \frac{n}{2}$$

for all $u \in G_n - \{v\}$.

Theorem1 shows that each of the branch sizes rooted at the rumor center is less or equal to $\frac{n}{2}$, which means that the rumor center is a vertex with subtree branches that are balanced in their sizes. We now introduce a graph-theoretic notion of G_n that provides an alternative equivalence characterization of the rumor center by using Theorem 1 to establish the link. This equivalence relationship will be leveraged later in Section IV to design poset-constrained centroid-based algorithms.

Definition III.2. Define the branch weight of a vertex v in G_n by

$$\mathsf{weight}(v) = \max_{c \in \mathsf{child}(v)} t_c^v.$$

The vertex of G_n with the *minimum weight* is called the *centroid* of G_n [17]. By its definition, removing this centroid from G_n results in disconnected components in which the size of the biggest component is the smallest possible. Furthermore, the size of the smallest component is the biggest possible. Let us also define the *distance centrality* of $v \in G_n$ as $\mathscr{D}(v, G_n) = \sum_{j \in G_n} d(v, j)$, where $d(v, j)$ is the distance (in terms of hop) between vertices v and j [18]. The vertex in G_n with the minimum distance centrality is called the *distance center*.

We now use graph convexity to provide an equivalent characterization to the rumor center in [7], [9], [10], [12] for general tree graphs.

Theorem 2. *Let G_n be a general tree graph and v is a vertex in G_n. Then, the following statements are equivalent:*

1) *The vertex v is a rumor center of G_n and also a distance center of G_n (proved in [10]).*
2) *The vertex v is a centroid of G_n.*

B. A Message Passing Algorithm for Graph Centroid

Let $M^{i \to j}$ denote the message from vertex i to vertex j. To calculate the weight of all vertices in G_N, we need to assign each $M^{i \to j}$ a number for all $(i, j) \in E(G_N)$. Let $M^{i \to j}$ be the size of t_i^j. So, we have $M^{i \to j} + M^{j \to i} = N$. And also for any vertex $v \in V(G_N)$, we have $weight(v) = max\{M^{i \to v} | \forall i \; is \; adjacent \; to \; v\}$. In the Algorithm 1 below, we first find all $M^{i \to j}$, and then use proposition 2 to locate the $centroid$, finally we set weight to all vertices. Let $\mathsf{Diff}(i, j)$ be defined by $\mathsf{Diff}(i, j) = |M^{i \to j} - M^{j \to i}|$.

Theorem 3. *Given a tree G_n with n vertices.*
$v_c \in G_n$ is the centroid if and only if $\forall v$ adjacent to v_c and $v_i, v_j \in V(G_n)$, $min_{(v, v_c) \in E(G_n)}\{\mathsf{Diff}(v_c, v)\} \leq \{\mathsf{Diff}(i, j)\}$. Moreover, for any $u \in G_n$, on the path from v_c to u say $(v_1, v_2, ..., v_D)$, where $v_1 = v_c$ and $v_D = u$. The sequence of $\mathsf{Diff}(v_i, v_{i+1})$ for $i = 1, 2...D$ is increasing.

IV. The Protection Node Placement Problem

In this section, we formulate our problem as an optimization problem on a network. We model the networked infrastructure as an acyclic connected graph with N vertices and denote as G_N. Let $V(G_N)$ be the vertex set of G_N and $E(G_N) = \{(v_i, v_j) | v_i, v_j \in G_N\}$ be the edge set of G_N. Let G_n be

the connected subgraph in G_N of vertices being affected by the cascading failure. Let V_P be the set of protected vertices. We let $\mathbf{E}(|G_n|)$ be the expectation of the number of failed vertices (i.e., the outage due to the spread of the cascading failure should it happens). Then, the protection node placement problem can be formulated as follows:

$$\begin{array}{ll} \underset{v\in G_n}{\text{minimize}} & \mathbf{E}(|G_n|) \\ \text{subject to} & |V_P| = k, \end{array} \tag{4}$$

where k is the cardinality of the number of protection nodes. Now, (4) is in general a stochastic program that is hard to solve. We next show that, when G_n has a tree topology, (4) can be simplified as a deterministic problem.

Let the $C(\{V_P\}) = (C_1^{\{V_P\}}, C_2^{\{V_P\}}, ..., C_m^{\{V_P\}})$ be the sequence of connected components after removing vertices in V_P from G_N. Assume that the failure starts from a vertex v^* uniformly picked in G_N, and that the cascading failure stops spreading once the spreading reaches a protection node. In this case, the number of vertices being affected is the number of vertices in the connected component that contains v^* when all the vertices in V_P are removed from G_N. For example, in Figure 1, suppose v_1 is protected and one of $\{v_2, v_3, v_4\}$ is the source of the failure, then the failure will affect $\{v_2, v_3, v_4\}$, i.e., the expectation of the number of nodes affected by the failure is $\frac{1}{4}(3^2)$, where $\frac{1}{4}$ is the probability of each vertex being picked initially. Hence, the optimization problem in (4) can be expressed as the following equivalent problem:

$$\begin{array}{ll} \underset{V_P\subseteq V(G_n)}{\text{minimize}} & (C_1^{\{V_P\}})^2 + (C_2^{\{V_P\}})^2 + ... + (C_m^{\{V_P\}})^2 \\ \text{subject to} & |V_P| = k, \end{array} \tag{5}$$

where the variable in this optimization problem is a set of vertices in G_N, and m is the number of connected component after removing V_P from G_N.

In the following, we show how the centroid to (5) can be a feasible solution to (5) and also demonstrate when it solves (5) optimally. Now, the centroid is defined as:

$$\underset{v\in G_N}{\text{minimize}} \quad \max_{1\le i\le D}\{C_i^v\}. \tag{6}$$

In (6), C_i^v is the i-th connected component after removing v from G_N and D is defined by $\max\limits_{v\in G_N} d_v$. Note that, if there is a vertex v such that $d_v = j < D$, then we define $C_i^v = 0$ for $j \le i \le D$. In particular, after we have added a new auxiliary variable $\lambda \in \mathbf{R}^{D\times 1}$ on (6), we obtain

$$\begin{array}{ll} \underset{v\in G_N}{\text{minimize}} & \max\limits_{\lambda\in\mathbf{R}^{D\times 1}} \sum\limits_{i=1}^{D} \lambda_i \cdot C_i^v \\ \text{subject to} & \lambda^T \cdot \mathbf{1} = 1, \\ & \lambda_i > 0,\ i = 1, \ldots, D. \end{array} \tag{7}$$

Let λ_i be defined as $\frac{C_i^v}{N-1}$, for $i = 1 \ldots D$. By the definition of C_i^v, we have $\sum\limits_{i=1}^{D} C_i^v = N - 1$. Hence, $\sum\limits_{i=1}^{D} \lambda_i = 1$, which implies λ is feasible in (7). Then (7) becomes an upper bound to the optimal value of the following problem:

$$\underset{v\in G_N}{\text{minimize}} \quad \frac{1}{N-1}\sum_{i=1}^{D} C_i^v \cdot C_i^v. \tag{8}$$

Observe that (8) is the same as the form in (5) when $k = 1$. This means that the centroid of G_N is a feasible (but suboptimal) solution for the problem in (5) even if we only pick a single vertex as the protection node. On the other hand, from the relation between the ℓ_2-norm and $\ell_{\inf}$-norm, we have

$$\sqrt{(C_1^v)^2 + (C_2^v)^2 + ... + (C_D^v)^2} \ge \max_{1\le i\le D} C_i^v.$$

Hence,

$$\min_{v\in G_N} \sum_{i=1}^{D} (C_i^v)^2 \ge \min_{v\in G_N} \max_{1\le i\le D} (C_i^v)^2,$$

and we have thus established upper and lower bounds of the optimal value in (5) given by

$$\min_{v\in G_N} \max_{1\le i\le D} (C_i^v)^2 \le \sum_{i=1}^{D} (C_i^v)^2 \le (N-1) \min_{v\in G_N} \max_{1\le i\le D} C_i^v. \tag{9}$$

where, under the special case of k being 1, the centroid of G_N is the optimal solution corresponding to the optimization problems in the upper and lower bounds.

Theorem 4. *Let G_N be a graph such that the centroid v_c of G_N is the only vertex with $d_{v_c} > 2$, i.e., for all $v \in G_N$ and $v \ne v_c$, $d_v \le 2$, then v_c is the optimal solution for (5) when $k = 1$.*

Theorem 4 can be proved by considering a sufficient condition of the optimality of v_c. Let $(C_1^{v_c}, C_2^{v_c}, ..., C_{d_{v_c}}^{v_c})$ be defined as above but in a decreasing order, i.e, $C_i^{v_c} \ge C_j^{v_c}$ whenever $i > j$. The sufficient condition for optimality is: For any $v \in V(G_n)$, there is an integer q such that, $C_i^v \ge C_i^{v_c}$ for $i = 1, \ldots, q$ and $C_i^v \le C_i^{v_c}$ for $i = q+1, \ldots, d_{v_c}$.

V. General k-protection Placement Algorithm

A. Algorithm to place k protection nodes

So far, our results in the previous sections apply to acyclic graphs, i.e., networks with a tree topology. For the general case of a graph with a general topology, e.g., having cycles, we propose a Breadth-First Search (BFS) heuristic method to solve (4) by finding a k-protection set on a given graph G_N based on the centroid decomposition technique. Our k-protection placement algorithm contains three parts, the first part is a graph decomposition algorithm. In the first part, we use a well known graph decomposition method called the *centroid decomposition*. By leveraging the properties of the centroid of a tree, at each recursion, we can decompose the tree into components that are roughly balanced in size (i.e., each subtree component has size $\le N/2$).

The second part is to construct a *centroid tree* from the result of the centroid decomposition. Let $v_c^{i,j}$ denote those centroids from the i-th recursion and j is the number of

centroids in the i-th recursion. For example, with $v_c^{1,1}$ as the centroid of G_N, and if $d_{v_c^{1,1}} = 3$, then there are $v_c^{2,1}$, $v_c^{2,2}$ and $v_c^{2,3}$, respectively, three centroids in the second recursion from the three connected components after $v_c^{1,1}$ has been removed from G_N. Let the centroid tree be denoted as T_c. Note that T_c is a tree rooted at the first centroid, and that $height(T_c) \leq log(N)$. The *height* of a vertex v in a rooted tree is defined by $height(v) = \max_{v_i \in desc(v)} d(v_i, v)$, where $desc(v)$ is a set of all descendants of v in the rooted tree. The height of the rooted tree is defined as the height of the root.

The third part is selecting k vertices from G_N based on T_c. Let $t_v^{\mathsf{parent}(v)}$ be defined as in Definition III.1 on T_c. We compute $t_v^{\mathsf{parent}(v)}$ for each $v \in T_c$. For example, in Figure 2, $t_{2_2}^1 = 2$ and $t_{2_3}^1 = 6$. After computing $t_v^{\mathsf{parent}(v)}$ for all v, we sort all the vertices in T_c according to their $t_v^{\mathsf{parent}(v)}$ in a decreasing order. Let Sort_v be the ordered list. Lastly, select the first k vertices in Sort_v to be the protection vertices set as the output of the algorithm.

Algorithm 1 Centroid Decomposition and Centroid Tree

Initially set $currentLV = 0$
CENTROID-DECOMPOSITION(T,$currentLV$,$v_c^{previousLV}$)
$currentLV = currentLV + 1$
Compute the centroid v_c of T (if there is two centroids, then randomly pick one)
$v_c.lv = currentLV$
Decompose T into several subtrees $T_j's$ by removing v_c from T
$V(T_c) = V(T_c) \cup \{v_c\}$
if $v_c.lv \neq 1$ **then**
  $E(T_c) = E(T_c) \cup \{(v_c, v_c^{previousLV})\}$
end if
for each subtree T_j **do**
  if $|T_k| > 1$ **then**
    CENTROID-DECOMPOSITION(T_j,$currentLV$,v_c)
  else
    $v.lv = currentLV + 1$, $\forall v \in V(T_j)$
    $V(T_c) = V(T_c) \cup \{v\}$
    $E(T_c) = E(T_c) \cup \{(v, v_c^{previousLV})\}$
  end if
end for

Algorithm 2 Construct V_P

Input: T_c, k, Set $V_P = \{\ \}$
Compute $t_v^{\mathsf{parent}(v)}$ for each $v \in T_c$
Let Sort_v be the list of vertices in T_c sorted in a decreasing order according to $t_v^{\mathsf{parent}(v)}$
for $i = 1 \ldots k$ **do**
  $V_P = V_P \cup \mathsf{Sort}_v(i)$
end for

B. Experimental Results for Finite d-regular Tree Networks

In this section, we evaluate the performance of the proposed k-protection placement algorithm against cascading outage.

TABLE II
DETAILS OF SIZE OF $|k|$ AND THE AVERAGE NUMBER OF VERTICES n AFFECTED BY THE CASCADING FAILURE OVER 2000 TIMES SIMULATION ON THE REAL WORLD POWER GRID NETWORK.

$\|k\|$	100	200	210	220	230	250	300
Average(n)	3025	1439	1137	1043	680	489	172
$\|k\|$	400	500	600	700	800	900	1000
Average(n)	71	31	17	12	8	6	5

We provide two experimental results on two different underlying networks G_N. In each simulation, we first apply Algorithm 1 and 2 on a given G_N to find the k-protection set. Second, we simulate a cascading failure over G_N, the spreading of the cascading failure follows the SI spreading model. The spreading will stop when there is no vertex can be affected by cascading failure. For each given size of k, after placing the protection vertices, we simulate the cascading failure for $2,000$ times, and in each simulation, the source vertex is uniformly chosen from G_N, i.e., each with probability $1/N$.

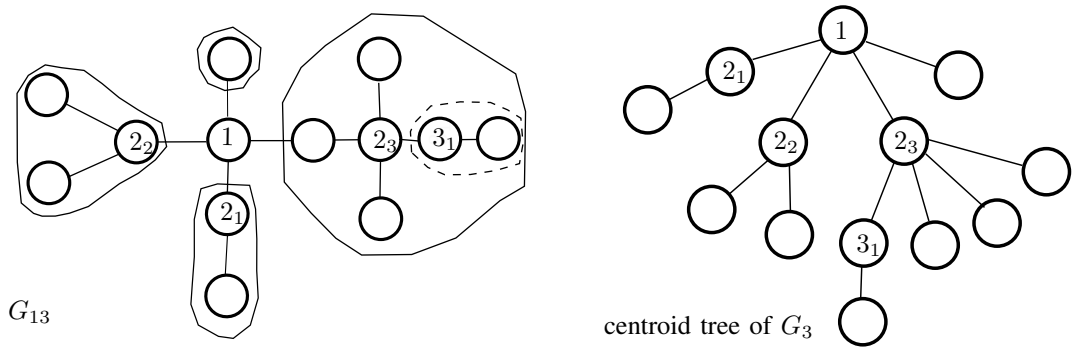
1) Tree Network with $N = 4941$: Here we provide a simulation result on a general tree randomly generated by a well-known preferential attachment based model: "BarabsiAlbert model". We evaluate the performance of the algorithm based on different sizes of k and observe the average outage incurred. We can observe that in the tree network, the average number of n decreases extremely fast when $|k| > 20$.

2) Real World Network with $N = 4941$: In this simulation, G_N is the western United State power grid network analyzed in [19]. Each vertex in G_N represents a substation in the country and each edge represents a high voltage line. There are 6594 edges in this power grid network. Note that the scale of y-axis is different from the random-tree simulation. When $|k| \leq 250$, the outage performance is bad due to the connectivity of the power grid network. When $|k| \geq 300$, the average number of n drops to 172, which implies that we only need to protect 6% of G_N to reach the desired performance to mitigate outage.

VI. CONCLUSION

We studied the problem of averting cascading failures in networked infrastructures by formulating an optimization problem of placing protection nodes to mitigate systemic risks due to cascading failures. We first introduced the notion of a partially ordered set as a means of modeling causality for inference of the most probable source of cascading failures. We then exploited poset-constrained topological properties to derive equivalence relationship with the graph-theoretic centroid, which we demonstrated to be feasible and optimal for the single protection node case. For the general case, we leveraged centroid decomposition to propose computationally-efficient message-passing algorithms that can be parameterized recursively based on the branch weight centrality to place protection nodes. In addition, we derived asymptotic results on the solution when the graph becomes infinitely large, and also proposed a suboptimal heuristic for the general case of

Fig. 2. Example of centroid decomposition of G_{13} and the tree on the left is the centroid tree from the centroid decomposition. After removing 1 from G_{13}, we have four connected components. For simplicity, the notation 2_i for $i = 1, 2, 3$ are equivalent to the notation $v_c^{2,i}$ used in Section V which are the centroids in the second recursion. Vertex 3_1 is the centroid from the third recursion.



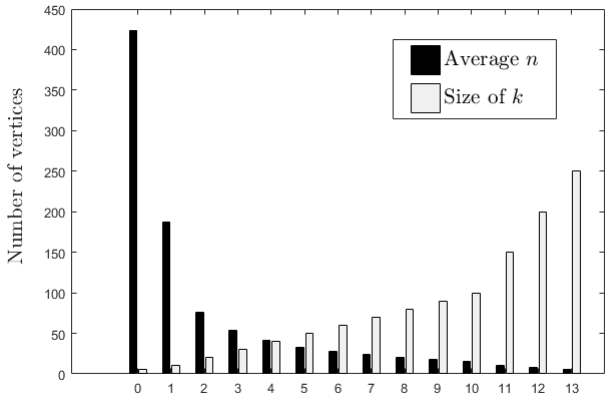


Fig. 3. A simulation result when G_N is a random tree. The y-axis represents the number of vertices and the x-axis represents each trial with different size of k.

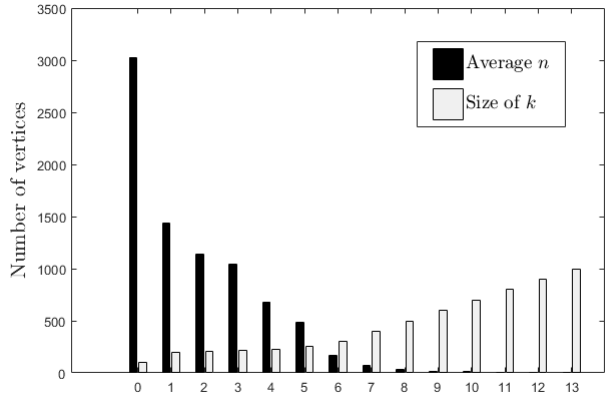


Fig. 4. A simulation result when G_N is a real world network: Western United State Power Grid Network. The y-axis represents the number of vertices and the x-axis represents each trial with different size of k.

placement of poset-constrained protection nodes in general graphs that have cycles.

REFERENCES

[1] C. J. S. Erjongmanee and J. Momoh, "Inferring network-power cascading disruptions and sustainability," *Proc. of International Conference of Neural Network*, August 2011.

[2] S. M. C. Milling, C. Caramanis and S. Shakkottai, "Network forensics: Random infection vs. spreading epidemic," *Proc. ACM SIGMETRICS*, 2012.

[3] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," *Proc. IEEE INFOCOM*, 2005.

[4] S. V. B. A. Bashan, Y. Berezin and S. Havlin, "The extreme vulnerability of interdependent spatially embedded networks," *Nature Physics*, vol. 9, pp. 667–672, August 2013.

[5] W. Luo, W. P. Tay, and M. Leng, "Identifying infection sources and regions in large networks," *IEEE Trans. Signal Processing*, vol. 61, no. 11, pp. 2850–2865, 2013.

[6] W. Dong, W. Zhang, and C. W. Tan, "Rooting out the rumor culprit from suspects," *Proc. of IEEE ISIT*, 2013.

[7] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rumor source detection with multiple observations: Fundamental limits and algorithms," *Proc. of ACM SIGMETRICS*, 2014.

[8] C. W. Tan, P. D. Yu, C. K. Lai, W. Zhang, and H. L. Fu, "Optimal detection of influential spreaders in online social networks," *Proc. of CISS*, pp. 145–150, 2016.

[9] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rooting out rumor sources in online social networks: The value of diversity from multiple observations," *, IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 663–677, 2015.

[10] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: theory and experiment," *Proc. of ACM SIGMETRICS*, 2010.

[11] D. Shah and T. Zaman, "Rumor centrality: a universal source detector," *Proc. of ACM SIGMETRICS*, 2012.

[12] D. Shah and T. Zaman, "Rumors in a network: Whos's the culprit?," *IEEE Trans. Information Theory*, vol. 57, pp. 5163–5181, 2011.

[13] P. Y. M. Fuch, "Rumor source detection for rumor spreading on random increasing trees," *Electronic Communications in Probability*, vol. 20, no. 2, 2015.

[14] P. J. Slater, "Fault-tolerant locating-dominating sets," *Discrete Mathematics*, vol. 249, April 2002.

[15] N. T. Bailey, "The mathematical theory of infectious diseases and its applications," *Griffin*, 1975.

[16] B. Iriarte, "Graph orientations and linear extensions," *Discrete Mathematics and Theoretical Computer Science*, 2017.

[17] B. Zelinka, "Medians and peripherians of trees," *Arch. Math.*, vol. 4, no. 2, pp. 87–95, 1968.

[18] D. West, *Introduction to Graph Theory*. Englewood Cliffs, New Jersy: Prentice Hall, 1996.

[19] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440–442, June 1998.