Averting Cascading Failures in Networked Infrastructures: Poset-Constrained Graph Algorithms

Pei-Duo Yu, Chee Wei Tan, and Hung-Lin Fu

***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. The ensuing problem of containing future cascading failures by placement of protection or monitoring nodes in the network is complicated by the uncertainty of the failure source and the missing observation of how the cascading might unravel, be it the past cascading failures or the future ones. This paper examines the problem of minimizing the outage when a cascading failure from a single source occurs. A stochastic optimization problem is formulated where a limited number of protection nodes, when placed strategically in the network to mitigate systemic risk, can minimize the expected spread of cascading failure. We propose the vaccine centrality, which is a network centrality based on the partially ordered sets (poset) characteristics of the stochastic program and distributed message-passing, to design efficient approximation algorithms with provable approximation ratio guarantees. In particular, we illustrate how the vaccine centrality and the poset-constrained graph algorithms can be designed to tradeoff between complexity and optimality, as illustrated through a series of numerical experiments. This paper points toward a general framework of network centrality as statistical inference to design rigorous graph analytics for statistical problems in networks.**

***Index Terms*—Cascading failure, viral spreading, graph theory and algorithms, large-scale stochastic optimization, message-passing algorithms, approximation algorithm, network centrality.**

I. Introduction

THE propagation of failures in critical networked infrastructures and the viral spreading of rumors (or even computer virus) in online social networks share many common features. The cascade trigger originates from (typically, a single) unknown or clandestine sources in the network and then spreads rapidly across the network. In the case of critical networked infrastructures, this is usually due to poorly-designed network components, when unwittingly put together, can make the network fragile and easily vulnerable to cascading failures that typically lead to catastrophic consequences. For example, a software glitch on the supervisory control network of a power grid system can lead to widespread power outage [2]. A recent well-known case is the 2013 Northeast blackout in northern America that saw widespread power outages and damages happening within minutes or hours across many states that led to the network downtime on the order of days.

In cascading failures, the onset can originate from either a random failure or intentional attack of any source node in the network, and the reach of the ensuing cascade depends on many factors such as the underlying network connectivity and the interaction dynamics between nodes. One of the most striking features is the position of this source node in the network that determines the cascading amplification prowess and the extent of the spread. It has been shown recently in [3] that, even when a small number of control devices are unreliable, the network topology of interdependent and spatially embedded networks can greatly reduce the resilience of the system against cascading failures. This also means that unraveling the cascade to trace the source of failure is especially hard. In other words, when a cascading failure occurs, how to quickly contain the spread or inoculation (even without knowing the source of failure) is of essence, because this relates to how quickly a network can recover from temporal malfunction or network robustness against cyberattacks.

Critical networked infrastructures such as the power grid and the Internet are not merely a network performing a basic functionality (such as routing or forwarding). Rather, it is imperative to install so-called *protection nodes* that, besides their own basic functionality, can deliver some form of inoculation strategy to protect the entire network from undesirable cascading spread. For example, in online social networks or the Internet, information spread may be stopped through online content filtering or blocking by some pre-selected users in the network. Another example is the installation of special-purpose power circuit breakers that act as border or anti-islanding gateways between power grids to automatically decouple problematic grids from working ones in order to prevent power outage from spreading. Conceivably, protection cost may be high and thus limit the number of protection nodes that can be deployed. 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 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).

Manuscript received September 30, 2017; revised March 28, 2018 and May 20, 2018; accepted May 21, 2018. Date of publication June 7, 2018; date of current version July 27, 2018. This work was supported in part by the Research Grants Council of Hong Kong RGC 11212114, in part by the Hong Kong Innovation and Technology Fund 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 under Grant 61771017. The material in this paper was presented in part in [1] at the 52nd Annual Conference on Information Systems and Sciences, Princeton, NJ, USA, Mar. 2018. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Meng Wang. *(Corresponding author: Pei Duo Yu.)*

P.-D. Yu and C. W. Tan are with the City University of Hong Kong, Hong Kong (e-mail: peiduoyu2-c@my.cityu.edu.hk; cheewtan@cityu.edu.hk).

H.-L. Fu is with the National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: hlfu@math.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2018.2844813

The placement of protection nodes to mitigate systemic risks requires understanding the mathematical coupling between the stochastic process of spreading and the underlying topological network structure in order to identify probable sources of failure given a snapshot observation of the failed nodes. Snapshot observations of a cascade may be based on actual incidents in the past or it can even be a hypothetical one based on simulated forecasting. Understanding how a cascade in complex networks evolves over time has received considerable attraction in the literature [2]–[7]. In [5], Ganesh *et al.* analyzed the relation between the network topologies and the persistence of epidemics. In [7], Buldyrev *et al.* first considered the problem of identifying cascading failures in the interdependent networks, and in [3], Bashan *et al.* considered the vulnerability of the spatially embedded networks which can be modelled as a lattice networks. Khamfroush *et al.* [6] extend the analysis of the cascading failures to the interdependent networks. Another direction is the problem of finding the source of a cascading which is fairly recent and still an open problem. Shah and Zaman [8] first studied the single source detection problem by modeling the rumor spreading in a network as a discrete-time model and deriving a maximum likelihood estimator for the source. This source detection problem was subsequently extended to various problem settings, e.g., extension in [9] to considering a priori knowledge of suspect nodes, extension in [10]–[12] to multiple source detection, extension in [13] to detecting influential spreaders. In [14], Zheng and Tan provided a probabilistic characterization of rumor graph boundary for maximum likelihood estimation. Fuch and Yu [15] derived the correct detection probability for different types of increasing trees by considering the generating function of the number of the increasing trees. In [16], the authors studied graph-theoretic conditions to distinguish between epidemic and random spreading. In [17], Milling *et al.* studied how to hide the rumor source using adaptive diffusion based on Galton Watson tree analysis over infinite graphs.

On the other hand, how to proactively protect the networks from the cascading failures is also an important issue. In [18], Tootaghaj *et al.* considered the power grid and its communication network, and proposed a two-phase recovery approach to first alleviate the cascading failures and then to recover. A related problem is to predict where to place a number of protection nodes in order to minimize the outage of cascading failure in advance. In [19], Omi *et al.* considered another approach by breaking down the connections between the infected communities and the susceptible communities while minimizing the impact on the communities performance, i.e., preserving the inter-community connection. In [20], Drakopoulos *et al.* studied how to contain rapid contagion by dynamically curing some of the infected nodes under a budget constraint.

In this paper, we consider a discrete Susceptible-Infectious (>*SI*) spreading model over a homogeneous single network and we focus on how to place the protection node in advance. Examples of such homogeneous networks are the autonomous systems routers in the Internet. Another example is the transmission towers in the transmission grid of the power network. Assume that the network topologies is the only information provided due to the limited information about the network in the real world. We define outage of a cascading failure as the maximum number of nodes that can be adversely affected when the spread occurs. 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 [21] 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 probable cascading failure source, that in turn leads naturally to feasible protection node placement strategies against cascading failures. The inter-coupling of protection node placement and finding the most probable source of cascading also leads to new mathematical insights on restraining contagious 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, which is central to our algorithm design to mitigate the systemic risks of cascading failures.
- We formulate a stochastic optimization problem of protection nodes placement in networks that have tree-graph topology to minimize the expected size of cascade graphs. We show that the graph centroids are feasible solution, and consider special cases under which it is optimal for a single protection node. Finding the protection nodes amounts to computing the *vaccine centrality* for solving the stochastic program. Furthermore, a main result (e.g., see Theorem 6 in this paper) is to demonstrate that the vaccine centrality leads to approximation algorithms with provable approximation ratio guarantees.
- In the general case, by leveraging the centroid decomposition, we propose a computationally efficient algorithm that can be parameterized recursively based on the vaccine centrality to place protection nodes to mitigate systemic risk of cascading failures.
- For extremely large graphs, we use analytic combinatorics to prove an asymptotic result showing that the centroid is globally optimal in minimizing the systemic risk. Lastly, we highlight the conceptual simplicity of the *network centrality as statistical inference* framework to solve inferential statistical problems over large graphs.

II. Modeling Cascading Failure Using Posets

In this section, we model the occurrence of cascading failures in a networked infrastructure by using an undirected graph $G = (V(G), E(G))$, where $V(G) = \{v_1, v_2, \dots\}$ is a set of nodes and $E(G)$ is the set of edges of the form (v_i, v_j) for nodes v_i and v_j in $V(G)$ (cf. Table I). In other words, the nodes performing the basic functionality in this networked infrastructure are the nodes 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 interaction in most man-made networked infrastructures. The degree of a node v_i is the number of its neighbors denoted by d_{v_i}.

TABLE I
GLOSSARY OF KEY NOTATIONS USED IN TWO MAIN SECTIONS, NAMELY SECTION III AND SECTION IV, ON PARTIALLY ORDERED SET-BASED RESULTS FOR CAUSAL INFERENCE AND THE RELATED STOCHASTIC OPTIMIZATION SOLUTION TO MINIMIZE CASCADE SPREAD GRAPHS

Notation	Remark
G_N	Underlying network (consisting of all susceptible nodes)
G_n	A subgraph of G_N with n nodes affected by a probabilistic cascading model
t_b^a	Subtree of a rooted tree with root node a and with branch node b
$v^\star(G)$	*Graph centroid* of G
$\mathbf{Expect}(\lvert G_n\rvert)$	The expected size of a cascade spread graph G_n that occurs randomly in G_N
V_P	The set of nodes in G_N to be protected with vaccine
$C(\{V_P\})$	Sequence of connected components after removing all nodes in V_P from G_N
T_c	Centroid tree obtained recursively from the centroid decomposition of G_N. Particularly, T_c is a tree abstract data type and $v^\star(T_c) = v^\star(G_N)$.
$\lvert t_v^{v^\star(T_c)}\rvert$	Vaccine centrality of v defined as the size of the subtree $t_v^{v^\star(T_c)}$ in T_c.

To model the cascading failures for general networked infrastructures, we assume a basic epidemic model known as the susceptible-infectious model (e.g., see [22]) to model cascading failures in networks. In this model, there are two types of nodes: (i) susceptible nodes 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 if $(v_i, v_j) \in E(G)$. Once a susceptible node 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 $(v_i, v_j) \in E(G)$, 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 [8], [9], [11], [12]. In this paper, we add a new type of node which is protected node, to extend the original SI model.

Definition II.1: In the extended SI model, we have three types of nodes described as follows:

- Susceptible node: Nodes that are susceptible to failure.
- Infected node: Nodes that have failed.
- Protected node: Nodes that are protected and cannot spread the failure further.

A. Preliminaries of Linear Extensions of Posets

Suppose that a single source of failure in this network G_N originates from a node $v^\star \in V(G_N)$ 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 G_n. Obviously, the spread graph G_n is a connected subgraph of the underlying graph G_N as shown in Fig. 1.

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.

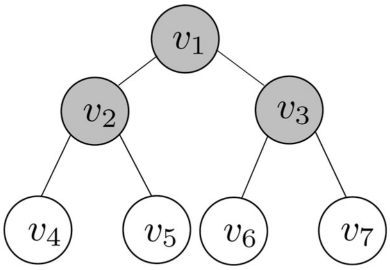

Fig. 1. An example of $V(G_N) = \{v_1, v_2, \ldots, v_7\}$, where $N = 7$, and the nodes in the shaded circles represent the nodes being affected by the cascading failure. Those nodes forms a connected subgraph G_3 of G_7.

Definition II.2: 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 $\leq_S$, i.e., for all $v_1 \leq_S^* v_2$ whenever $v_1 \leq_S v_2$. Given a spread graph G_n, and assume that $v' \in G_n$ is 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' . In Fig. 2, we use an example to illustrate how a tree can be viewed as a partially ordered set [23].

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

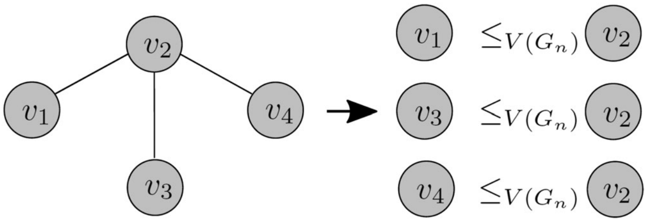

Fig. 2. An example of change a rooted tree structure to a partially ordered set. Note that, there is no relation between v_1, v_3 and v_4, since this order is a **partial** order. However, when considering a cascading failure on this graph with a specific order, for example $v_2 \to v_1 \to v_3 \to v_4$, then there is relation between any two nodes in this set.

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 spanning tree of 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. Assume that G_N is a tree, and G_n is a failure subgraph on G_N. For each node v in G_n, the rooted tree G_n rooted at v is a poset and we can compute the number of linear extensions on G_n. The number of linear extensions is equivalent to the number of cascading failure spreading orders starting from v and resulting in G_n. Intuitively, the node with the maximum number of linear extensions on G_n, is the node having the highest likelihood to cause the cascading failure [8]. Therefore, the node with the maximum number of linear extensions on the given network is an ideal node for protection placement. 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. However, the difficulty comes from identifying the right spanning tree of G_n. To compute the number of linear extensions of a node v on a given general graph G_n is computationally hard, a lower bound on the number of linear extensions is given in [23].

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

In this section, we provide analytical characterization of the graph centroid as the node with the maximum number of linear extensions of a poset, and propose 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. The graph centroid is the central concept that 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 nodes, for any $u, v \in G_N$, let t_v^u be the subtree rooted at v when u is the root of G_N and slightly abusing the notation of the size of the subtree $|t_v^u|$ as t_v^u (i.e., the context is clear whether t_v^u means the subtree or its size).

In this section, 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 [8] to compute the number of linear extensions on a rooted tree:

$$L(v, G_N) = \frac{(N-1)!}{t_{u_1}^v! \cdot t_{u_2}^v! \cdot \ldots \cdot t_{u_{d_v}}^v!} \cdot \prod_{i=1}^{d_v} L(u_i, T_{u_i}^v). \tag{1}$$

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. Note that $u_1, u_2, \ldots, u_{d(v)}$ are v's neighbors, and $t_{u_i}^v$ is the subtree size. A spreading order on G_N is equivalent to a numbering process that we number all nodes in G_N in an increasing order from 1 to n under the topological constraint of G_N. Hence, the fraction on the right-hand side in (1) is equivalent to the number of ways that we can allocate $N-1$ numbers into d_v piles, and each pile u_i is of size $t_{u_i}^v$. The remaining part of the right-hand side is the number of spreading orders in each pile. 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 node 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 [8]:

$$L(v, G_N) = N! \cdot \prod_{u \in G_N} \frac{1}{t_u^v}. \tag{2}$$

Using (2), we can find the rumor center, i.e., the node with maximum number of linear extensions. Now, consider two adjacent nodes u and v in G_N and a node $w \in G_N - \{u, v\}$, 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:

$$\frac{L(u, G_N)}{L(v, G_N)} = \frac{t_u^v}{N - t_v^u}, \tag{3}$$

which leads to the following result (see [8, Proposition 1]).

Theorem 1: Given a tree G_N with N nodes, $v \in G_N$ is a rumor center if and only if

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

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

Theorem 1 shows that each of the branch sizes rooted at the rumor center is less than or equal to $\frac{N}{2}$, which means that the rumor center is a node 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 node v in G_N by

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

The node of G_N with the *minimum weight* is called the *centroid* of G_N [24]. Denote the centroid of G_N by $v^\star(G_N)$. By its definition, removing $v^\star(G_N)$ 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 nodes v and j [25]. The node in G_N with the minimum distance centrality is

Algorithm 1: Message Passing Algorithm to Compute the Centroid of a Graph (whose size n can be an unknown).

```
  Input a tree $G$
  Choose a root $v_r$ from $G$
  for $u$ in T do
    if $u$ is a leaf then
      $M^{u\to\mathsf{parent}(u)} = 1$
    else
      if $u \neq v_r$ then
        $M^{u\to\mathsf{parent}(u)} = \Sigma_{j\in\mathsf{child}(u)} M^{j\to u} + 1$
      end if
    else
      $N = \Sigma_{j\in\mathsf{child}(v_r)} M^{j\to v_r}$
    end if
  end for
  Find the longest path $P = (p_1, \ldots, p_{k-1}, p_k)$ starting
from
  $v_r$, such that $\mathsf{Diff}(p_i, p_j)$ is decreasing along $P$, where
  $p_1 = v_r$ and $\mathsf{Diff}(v_i, v_j) = |2M^{v_i\to v_j} - N|$.
  $v^k$ is the centroid if $M^{v^k\to v^{k-1}} > M^{v^{k-1}\to v^k}$ else $v^{k-1}$ is
  the centroid.
```

called the *distance center*. We now provide an equivalent characterization to the rumor center in [8], [11], [12] for general tree graphs.

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

1) The node v is a rumor center of G_N.
2) The node v is a distance center of G_N.
3) The node v is a centroid of G_N, i.e., $v^\star(G_N)$.

In particular, the equivalence between the first two statements above has been proved in [8]. In words, statement 3, proved in this paper, characterizes the rumor center in terms of the sizes of its local subtrees. Moreover, Theorem 2 shows that, in the context of cascading failure in networks, the node with the maximum number of linear extensions is equivalent to the centroid (see, e.g., [24]). Intuitively, if we want to place a protection node, then choosing the centroid of G_N is a good choice, since after the centroid is protected, the cascade spread graph is at most $N/2$ in size (if we just pick one node to be protected). For example, the centroid of the spread graph in Fig.3 1 is v_1, and we can choose v_1 as the protected node.

It has been established in graph theory that a tree has either exactly one, or exactly two centroids joined by an edge (see, e.g., [24]). This implies that, there are *at most two* nodes with maximum number of linear extensions, and this scenario with two such nodes happens only when the maximum branch size is exactly $N/2$. Furthermore, that the centroid and the distance center coincides has been pointed out in [24].

B. A Message Passing Algorithm for Graph Centroid

Let $M^{i\to j}$ denote the message from node i to node j. To calculate the weight of all nodes 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 node $v \in V(G_N)$, we have $\mathsf{weight}(v) = \max\{M^{i\to v} | \forall i \text{ is adjacent to } v\}$. In the Algorithm 1 below, we first find all $M^{i\to j}$, and then use Theorem 3 to locate the $centroid$, finally we set weight to all nodes. 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 nodes, $\tilde{v} \in G_N$ is the centroid if and only if $\forall v$ adjacent to $\tilde{v}$ and $v_i, v_j \in V(G_N)$, $\min_{(v,\tilde{v})\in E(G_N)}\{\mathsf{Diff}(\tilde{v}, v)\} \le \{\mathsf{Diff}(v_i, v_j)\}$. Moreover, for any $u \in G_N$, on the path from $\tilde{v}$ to u say $(v_1, v_2, \ldots, v_D)$, where $v_1 = \tilde{v}$ and $v_D = u$. The sequence of $\mathsf{Diff}(v_i, v_{i+1})$ for $i = 1, 2 \ldots D$ is increasing.

Now, a practical implication of Theorem 3 is that this centroid for a given tree G_N can be found using graph algorithms (thereby providing alternative algorithms to the rumor source detection in [8]). Using graph-theoretic analysis, new algorithms can be designed with a computational time complexity $O(N)$, where N is the size of the input graph. Now, a key algorithmic design in statistical learning is the message passing algorithm framework (also known as belief propagation) whereby simple messages are exchanged between neighboring nodes in a graph and these local operations converge to the solution of a global problem iteratively [26]. We next propose such a new message passing algorithm to find the graph centroid in Algorithm 1.

The first part in Algorithm 1 is message passing where each node exchanges messages with their neighboring nodes that are updated in a recursive manner. As these messages are passed from the leaf nodes to their parent nodes who in turn aggregate the messages collected from their children nodes and pass the aggregated result to their parent nodes, this process iterates until each node computes their individual network centrality. The number of iterations is the same as the number of nodes in the graph, i.e., the computation complexity of this part is $O(N)$. The second part is to find the centroid, by leveraging Theorem 3. Since local optimality implies global optimality in the branch weight centrality, the length of the path in the second part is at most $N/2$, which implies the computation complexity is $O(N)$. Hence, the computation complexity for Algorithm 1 is $O(N)$. In the following, we highlight the key advantage of Algorithm 1 over that in [8]. First, the algorithm in [8] needs to compute $N!$ which is relatively larger than N, in Algorithm 1, the largest number is at most N. Second, Algorithm 1 is more scalable, because, when a new node has to be added into the network, we only need to update the nodes on the path from the newly-added node to the current centroid, and this modification complexity is at most the height of the tree. In Fig. 3, we use an example to illustrate these two steps in Algorithm 1.

Suppose the tree G_N is given, the message-passing Algorithm 1 ranks the importance for each node in terms of relative tree branch weight (or, equivalently, the number of linear extensions). The ranking of the nodes makes use of the relative centrality measure between adjacent nodes that is similar in measure to the rumor centrality, distance centrality or the branch weight centrality that we characterize as follows.

Theorem 4: Let G_N be a general tree with N nodes, and $u, v \in G_N$ are two adjacent nodes (neither u nor v needs to be the centroid). Then, the following statements are equivalent:

1) $L(v, G_N) \ge L(u, G_N)$.
2) $\mathscr{D}(v, G_N) \le \mathscr{D}(u, G_N)$.
3) $\mathsf{weight}(v) \le \mathsf{weight}(u)$.

Note that Theorem 4 implies Theorem 2. Also, Theorem 4 implies that the ranking among selected number of nodes can be determined using message passing in Algorithm 1. In the

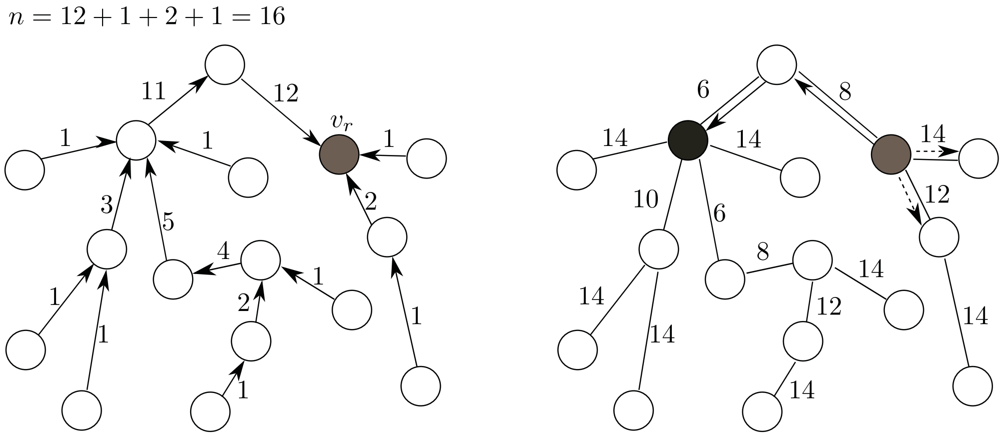

Fig. 3. Example of how Algorithm 1 works on a given tree G_{16}. The figure on the left is the first part of Algorithm 1. We first randomly pick a node as the root, which is the node in the grey color. The process of message passing starting from the leaves until the root receives messages from all of its children. The figure on the right compute $\mathsf{Diff}(v_i, v_j)$ for all $v_i, v_j \in G_{16}$. Finally, find a longest decreasing path P, where $P = (8 \to 6)$. Note that if $P = (8 \to 6 \to 6)$, we can still find the centroid by the last line of Algorithm 1.

previous section, we point out that the number of linear extensions of a node v is related to the number of spreading orders starting from v. In this section, we show that to rank a node v in accordance with $L(v, G_N)$ is equivalent to rank v based on its weight, moreover, ranking nodes based on weights is simpler. We shall leverage these graph-theoretic characterization to propose a *vaccine centrality* to address the problem of selecting nodes in the network to be vaccinated in order to minimize the expected outage of a cascading failure.

IV. Vaccine Centrality and Protection Node Placement

A. Problem Formulation

In this section, we formulate the problem of expected outage minimization from cascading failures as a stochastic optimization problem over a graph. Since the protection nodes are "immune" to the cascading failure, the cascading failure is not able to spread through the protection nodes, i.e, the graph is partitioned into several connected subgraphs after removing all the protection nodes. The optimization problem of interest is to evenly partition the graph into small subgraphs by placing the protection nodes. We model the networked infrastructure as an acyclic connected graph with N nodes denoted by G_N. Let V_P be the set of protection nodes in G_N protected by a vaccine. We let $\mathbf{Expect}(|G_n|)$ be the expectation of the number of failed nodes (i.e., the spread of the cascading failure should it happen). Then, the protection node placement problem can be formulated as follows:

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

where k is the cardinality of the number of protection nodes. Now, (4) is a stochastic program that is hard to solve in general. We shall show that, when G_N has a tree topology, (4) can be simplified as a deterministic problem that we can solve using the network centrality which uses partially ordered sets (poset) in graphs in Section III to identify the protection nodes. We call this the *vaccine centrality* for solving the stochastic program in (4).

B. Optimality Characterization and Bounds

Let the $C(\{V_P\}) = (C_1^{\{V_P\}}, C_2^{\{V_P\}}, \ldots, C_m^{\{V_P\}})$ be the sequence of connected components after removing nodes in V_P from G_N (cf. Table I). Assume that the failure starts from a node v uniformly picked in G_N, and that the cascading failure stops spreading once the failure affects all nodes in its connected component. In this case, the number of nodes being affected is the number of nodes in the connected component that contains v when all the nodes in V_P are removed from G_N. For example, in Fig. 1, if v_1 is protected, then $\mathbf{Expect}(|G_n|) = \frac{1}{7} \cdot (3 \cdot 3 + 3 \cdot 3 + 1 \cdot 0)$. This multiplicative factor $\frac{1}{7}$ is the probability of each node being picked initially. On therighthand-side, $3 \cdot 3$ is the number of nodes being affected by the cascading failure once it starts from v_2, v_4 or v_5 multiplied by $|\{v_2, v_4, v_5\}|$. Hence, the stochastic optimization problem in (4) can be equivalently expressed as the following deterministic problem:

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

where the variable in this optimization problem is a set of nodes 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 $v^\star(G_N)$ is defined as:

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

In (6), C_i^v is the i-th connected component after removing v from G_N and D is defined by $\max_{v \in G_N} d_v$. Note that, if there is a node 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

$$\begin{aligned} &\underset{v\in G_N}{\text{minimize}} \qquad \max_{\lambda\in\mathbf{R}^{D\times 1}} \sum_{i=1}^{D}\lambda_i \cdot C_i^v \\ &\text{subject to} \qquad \lambda^T\mathbf{1}=1, \\ &\qquad \lambda_i>0,\ i=1,\ldots,D. \end{aligned} \tag{7}$$

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 $\sum_{i=1}^{D} C_i^v = N-1$. Hence, $\sum_{i=1}^{D}\lambda_i = 1$, which implies λ is feasible in (7). Then (7) becomes an upper bound to the optimal value of the following problem:

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

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 node as the protection node. On the other hand, from the relationship between the ℓ_2-norm and ℓ_∞-norm,

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

Hence, we have

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

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 5: Let G_N be a graph such that the centroid $v^\star(G_N)$ of G_N is the only node with $d_{v^\star(G_N)}>2$, i.e., for all $v\in G_N$ and $v\neq v^\star(G_N)$, $d_v\le 2$, then $v^\star(G_N)$ is the optimal solution for (5) when $k=1$.

Theorem 5 can be proved by considering a sufficient condition of the optimality of $v^\star(G_N)$: For any $v\in V(G_n)$, there is an integer q such that, $C_i^v \ge C_i^{v^\star(G_N)}$ for $i=1,\ldots,q$ and $C_i^v \le C_i^{v^\star(G_N)}$ for $i=q+1,\ldots,d_{v^\star}$ where $(C_1^{v^\star}, C_2^{v^\star},\ldots,C_{d_{v^\star(G_N)}}^{v^\star})$ is defined as above but in a decreasing order, i.e, $C_i^{v^\star(G_N)} \ge C_j^{v^\star(G_N)}$ whenever $i>j$.

From Theorem 5, we deduce that, under some special cases, the centroid is indeed the optimal solution for (5). For example, in Fig. 4, the shaded node is the centroid with degree d and there are d paths connected to the centroid. Note that the length of each of these d paths need not be the same.

Corollary 1: If G_N is a tree, then the optimal solution of the optimization problem (5) when $|V_P|=1$ is the node v_B with the maximum betweenness centrality. (See (11) in the proof for the definition.)

Roughly speaking, the betweenness centrality [27] is proportional to the number of times that a node acts as a "bridge" on the shortest path for any two nodes in the graph. Since its inception in [27] in 1977, the betweenness centrality is often used as a routine in popular algorithms for clustering and community identification, and requires a complexity of $O(N+E(G_N))$ space and runs in $O(N\cdot E(G_N))$ time on general graphs [28]. The centroid $v^\star(G_N)$ can possibly be regarded as a heuristic approximation of the betweenness center v_B. However, particularly for the special case in Theorem 5, we have $d(v^\star(G_N),v_B)=0$. For general tree G_N, we have

$$d(v^\star(G),v_B) \le \frac{N+d_{v_B}-Nd_{v_B}+\sqrt{d_{v_B}I}}{2(d_{v_B}+1)},$$

where $I=(6N+5d_{v_B}+2Nd_{v_B}+2N^2d_{v_B}-2N^2+4)$.

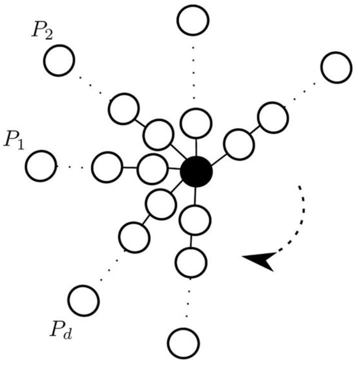

Fig. 4. An example illustrating that the centroid of G_N is the optimal solution for (5), the shaded node is the centroid of G_N.

V. Vaccine Centrality and Approximation Algorithm

A. Vaccine Centrality

In this section, we introduce the vaccine centrality of a given node in an induced tree abstract data type (*centroid tree*) based on the well-known graph decomposition method called the *centroid decomposition* [29], [30]. Let G_N be a tree and T_c be the corresponding centroid tree. The definition of the vaccine centrality of a given node v is defined by $|t_v^{v^\star(T_c)}|$ on T_c, where $v^\star(T_c)$ is the centroid of T_c. Note that the vaccine centrality is only defined on T_c instead of the original graph G_N, and each node in G_N has a corresponding node in T_c. Moreover, the centroid of T_c is also the centroid of G_N due to construction rules of T_c. Assume that v's parent node is removed from G_N, then the vaccine centrality of v measures how large a subtree of G_N can be decomposed when v is removed. Note that v can only be chosen after its parent node in T_c was chosen. For example, in Fig. 5, assume that node 1 is removed from G_N. We have 2_1, 2_2 and 2_3 are children node of 1 in T_c, and $|t_{2_1}^{v^\star(T_c)}|=2$, $|t_{2_2}^{v^\star(T_c)}|=3$ and $|t_{2_3}^{v^\star(T_c)}|=6$. Hence, after node 1 is chosen to be protected, the next choice is 2_3, since it has the maximum vaccine centrality, i.e., the protection of 2_3 can decompose the 6-nodes subtree into smaller subtrees.

B. Approximation Algorithm for 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 general topology, e.g., having cycles, we use the Breadth First Search (BFS) heuristic. In the BFS heuristic, we

Algorithm 2: Centroid Decomposition and Centroid Tree.

```
1:  Initially set $currentLV = 0$
2:  CENTROID-DECOMPOSITION ($T, currentLV, v^\star_{previousLV}$)
3:  $currentLV = currentLV + 1$
4:  Compute the centroid $v^\star(T)$ of $T$ (randomly pick one if
    there are two centroids)
5:  $v^\star.lv = currentLV$
6:  Decompose $T$ into several subtrees $T_j' s$ by
    removing $v^\star(T)$ from $T$
7:  $V(T_c) = V(T_c) \cup \{v^\star(T)\}$
8:  if $v^\star(T).lv \neq 1$ then
9:      $E(T_c) = E(T_c) \cup \{(v^\star(T), v^\star_{previousLV})\}$
10: end if
11: for each subtree $T_j$ do
12:     if $|T_j| > 1$ then
13:         CENTROID-DECOMPOSITION ($T_j, currentLV, v^\star(T)$)
14:     else
15:         $v.lv = currentLV + 1, \forall v \in V(T_j)$
16:         $V(T_c) = V(T_c) \cup \{v\}$
17:         $E(T_c) = E(T_c) \cup \{(v, v^\star_{previousLV})\}$
18:     end if
19: end for
```

Algorithm 3: Construct a Set of k Protection Nodes V_P.

```
1: Input: $T_c$, $k$, Set $V_P = \{\ \}$
2: Compute $t_v^{v^\star(G_N)}$ for each $v \in T_c$
3: Let $\mathsf{Sort}_v$ be the list of nodes in $T_c$ sorted in a
   decreasing order according to $t_v^{v^\star(G_N)}$
4: for $i = 1 \ldots k$ do
5:     $V_P = V_P \cup \mathsf{Sort}_v(i)$
6: end for
```

apply Algorithm 1, 2, and 3 on a BFS-induced spanning tree, which is denoted as T_{BFS}, of G_N. The intuition is that if the cascading failure starts from a node v, then this BFS spanning tree rooted at v would correspond to all the nearest neighbors of v being affected at the earliest time. Our k-protection placement algorithm contains three parts, the first part is graph decomposition using 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 a size less than or equal to $N/2$).

The second part is to construct a *centroid tree* T_c from the result of the centroid decomposition. Note that T_c is a tree rooted at the first centroid, i.e., the centroid $v^\star(T_{\mathsf{BFS}})$ of T_{BFS}, and the $\mathsf{height}(T_c) \leq \log_2 N + 1$. Besides, each node in T_{BFS} has a corresponding node in T_c and $v^\star(T_{\mathsf{BFS}}) = v^\star(T_c)$. In Algorithm 2, we denote the centroid found in the previous level as $v^\star_{previousLV}$.

The third part is selecting k nodes from T_{BFS} based on their vaccine centrality on T_c. We can use Algorithm 1 to compute $|t_v^{v^\star(T_c)}|$ for each $v \in T_c$. For example, in Fig. 5, $t^1_{2_2} = 3$ and $t^1_{2_3} = 6$. After computing $|t_v^{v^\star(T_c)}|$ for all v, we sort all the nodes in T_c according to their $|t_v^{v^\star(T_c)}|$ in a decreasing order. Let Sort_v be the ordered list. Lastly, select the first k nodes in Sort_v to be the protection nodes set.

In the following, we analyze the computational complexity and the optimality of Algorithm 2 and 3 when G_N is a tree. In Algorithm 2, the computational complexity of line 4 is $O(|T|)$ which is proved in the previous section. The recursion in line 13 executes at most $O(\log_2 N)$ times. Hence, the computational complexity of Algorithm 2 is $O(N \log_2 N)$. In Algorithm 3, line 2 is the message passing algorithm with complexity $O(N)$, and line 3 needs to sort N nodes with complexity $O(N \log_2 N)$. In summary, the total computational complexity is $O(N \log_2 N)$.

Theorem 6: Let $f(\{V_p\})$ denote the objective function in (5) and let V_p^* denote the optimal solution of (5). When G_N is a tree, the choice of V_p in Algorithm 3 guarantees that

$$1 \leq \frac{f(\{V_p\})}{f(\{V_p^*\})} \leq \frac{2}{c(1-c)},$$

where k is the size of the protection set V_p and $0 < c < 1$ is a constant such that $k = c \cdot N$.

In Theorem 6, it guarantees the performance of the Algorithm 3 in the worst case. For comparison, we use a *degree centrality*-based heuristic that sorts all the nodes according to their degrees. Thus, the *degree centrality* heuristic has $O(N \log N)$ computational complexity to select the protection set. In the following, we give an example illustrating that the performance of the degree-centrality heuristic cannot be bounded above as the size of the protection set increases.

Example 1: Suppose G_N is composed of two balanced tree graphs (e.g., the graph in Fig. 1) rooted at v and u respectively, and connected by a path $P = (p_1, p_2, \ldots, p_t)$, where p_i is a node on the path for $i = 1, \ldots, t$. Note that v is adjacent to p_1 and u is adjacent to p_t. Assume the length of P is close to N, i.e., t is much larger than the size of the two balanced trees on both sides. In this case the optimal strategy to place one protection node is to choose the node on the path P, and the cost $\mathbf{Expect}(|G_n|)$ will be $\frac{(N-1)^2}{2N}$ or $\frac{N-1}{2}$. The output of Algorithm 3 will be the same as the optimal solution. However, the degree centrality heuristic will output either v or u, and $\mathbf{Expect}(|G_n|)$ is around N. When the number of protection node increases, the output of degree centrality heuristic will not change too much until it starts to select the node on the path. On the contrary, $\mathbf{Expect}(|G_n|)$ of Algorithm 3 is bounded above by $\frac{2N}{k+1}$ due to the property of the centroid.

C. Asymptotically-Large Graph Regime

Networked infrastructures are typically very large and thus it is important to study mathematical behaviors that emerge when the number of vertices has an order of magnitude in hundred of millions or billions. In the following, we discuss how to solve (5) when the network size is infinitely large using analytic combinatorics. Such asymptotic analysis is especially useful for the case when the number of nodes in the networked infrastructure is not fixed but may grow sufficiently large. We make a few assumptions on the network structure and the growth process in deducing the asymptotic analysis by assuming that the network is a degree-regular tree (i.e., each node in the tree has the same degree except the leaves of the tree) while the tree grows asymptotically large as more nodes are attached to the leaves as time goes by (i.e., this can be attributed in part to the *network*

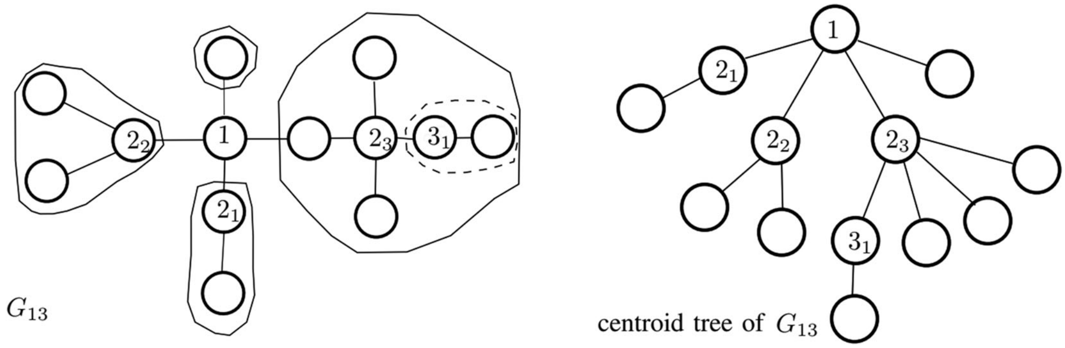

Fig. 5. Example of centroid decomposition of G_{13} and the tree on the right is the centroid tree from the centroid decomposition. After removing 1 from G_{13}, we have four connected components. For simplicity, the notation 2_i for $i = 1, 2, 3$ are equivalent to the notation $v^{\star}_{2,i}$ used in Section V which are the centroids in the second recursion, and node 3_1 is the centroid from the third recursion.

effect where the value of the network increases with the number of users).

We now consider the asymptotic relationship between the centroid and the other nodes in G_n as n grows to infinity. Note that Theorem 1 works for all N, hence, it provides an invariant characteristic of the centroid on a tree even when N goes to infinity. Define the ratio between the number of linear extensions of the centroid and the total number of linear extensions belonging to all the other nodes as

$$\frac{L(v^{\star}(G_N), G_N)}{\sum_{v\in G_N} L(v, G_N)}, \qquad v^{\star}(G_N) \text{ as the centroid of } G_N. \tag{10}$$

Let $k^{(d)}$ denote the ratio in (10), when G_N is a d-regular tree. We shall show that this ratio $k^{(d)}$ approaches a constant value when N goes to infinity. In particular, for the special case of G_N being a 3-regular tree, we can use a first-principle combinatorial counting method to prove the following Theorem 7 (please refer to the appendix for details):

Theorem 7: Let G_N be a 3-regular tree, then

$$\lim_{N\longrightarrow\infty} k^{(3)} = \frac{1}{4}.$$

To derive a general expression for d-regular tree graphs, we use analytic combinatorics to show that (please refer to the appendix for details):

Theorem 8: Let G_N be a d-regular tree, then the ratio $k^{(d)}$ as N grows to infinity is asymptotically:

$$\lim_{N\longrightarrow\infty} k^{(d)} = 1 - \frac{d}{2} + \frac{(d-2)\cdot\Gamma\left(\frac{d}{d-2}\right)}{2^{\frac{d}{d-2}}\cdot\Gamma\left(\frac{1}{d-2}\right)\Gamma\left(\frac{d-1}{d-2}\right)}.$$

Unlike Theorem 7, Theorem 8 shows that the number of linear extensions of posets in the centroid is proportional to those of other nodes by a factor analytically given by the Beta function (that is nonetheless still a constant value). The implication to the solvability of (5) is that the graph centroid is always a good choice of protection node even when N is growing asymptotically large. Finally, the following Theorem 9 provides an asymptotically general bound for $k^{(d)}$ when G_N is a d-regular tree.

Theorem 9: Let G_N be a d-regular tree, then we have,

$$\lim_{d\rightarrow\infty} k^{(d)} = 1 - \ln 2,$$

where $\ln 2$ is the natural logarithm of 2.

Moreover, for $d \geq 3$,

$$1/4 \leq k^{(d)} \leq 1 - \ln 2.$$

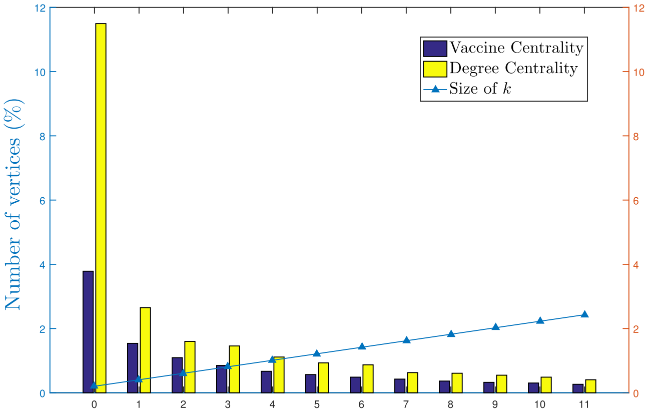

Fig. 6. Simulation results when G_N is a random tree generated by the "Barabási- Albert model". The y-axis represents the number of nodes and the x-axis represents each trial with different sizes of k in percentage.

TABLE II
DETAILS OF SIZE OF $|k|$ AND THE AVERAGE NUMBER OF NODES n AFFECTED BY THE CASCADING FAILURE OVER 2000 SIMULATIONS ON THE RANDOM TREE NETWORK

$\lvert k\rvert$	0.2	0.4	0.6	0.8	1	1.2
Algorithm3	3.78	1.54	1.09	0.85	0.67	0.57
DegreeCentrality	11.50	2.65	1.60	1.46	1.11	0.93
$\lvert k\rvert$	1.4	1.6	1.8	2.0	2.2	2.4
Average(n)	0.486	0.43	0.36	0.32	0.30	0.26
DegreeCentrality	0.87	0.63	0.61	0.55	0.49	0.40

D. Experimental Performance Evaluation

In this section, we evaluate the performance of the proposed k-protection placement algorithm. We provide several experimental results on four different underlying networks G_N. In each part of simulations, after k nodes are chosen to be protected, we simulate a cascading failure over G_N in which the spreading follows the SI spreading model. The spreading stops only when there are no more node that can be affected by cas-

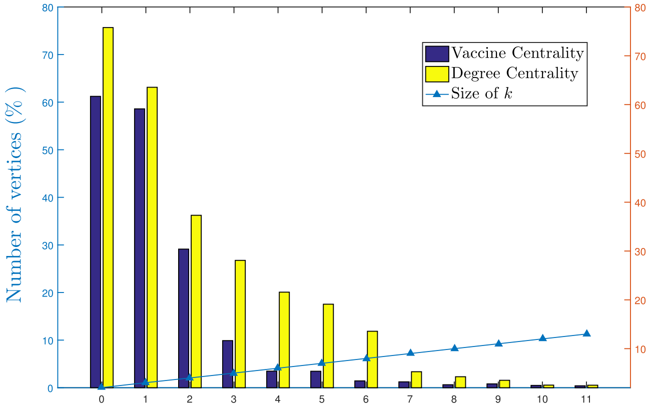

Fig. 7. Simulation results when G_N is a real world network: Western United State Power Grid Network. The y-axis represents the number of nodes and the x-axis represents each trial with different sizes of k.

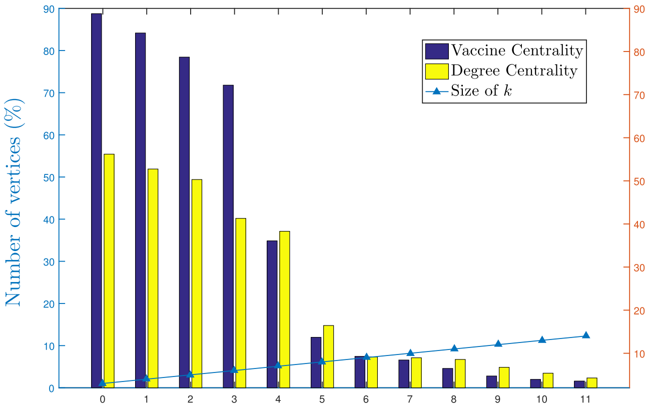

Fig. 8. Simulation results when G_N is a real world network: Europe renewable power system. The y-axis represents the number of nodes in percentages and the x-axis represents each trial with different sizes of k.

TABLE III
DETAILS OF SIZE OF $|k|$ AND THE AVERAGE NUMBER OF NODES n AFFECTED BY THE CASCADING FAILURE OVER 2000 SIMULATIONS ON THE REAL WORLD POWER GRID NETWORK

$\|k\|$	2	3	4	5	6	7
Algorithm3	61.2	58.6	29.1	9.90	3.48	3.46
DegreeCentrality	75.7	63.1	36.2	26.8	20.1	17.5
$\|k\|$	8	9	10	11	12	13
Algorithm3	1.44	1.23	0.95	0.75	0.49	0.40
DegreeCentrality	11.9	3.36	2.31	1.56	0.55	0.53

TABLE IV
DETAILS OF SIZE OF $|k|$ AND THE AVERAGE NUMBER OF NODES n AFFECTED BY THE CASCADING FAILURE OVER 2000 SIMULATIONS ON THE REAL WORLD EU RENEWABLE POWER NETWORK

$\|k\|$	3	4	5	6	7	8
Algorithm3	88.8	84.2	78.4	71.8	34.9	12.0
DegreeCentrality	55.4	51.9	49.1	40.2	37.1	14.8
$\|k\|$	9	10	11	12	13	14
Algorithm3	7.45	6.59	4.59	2.79	2.00	1.60
DegreeCentrality	7.31	7.12	6.72	4.86	3.46	2.33

cading failure. For example, in Fig. 1, if we protect v_3 and the cascading failure starts from v_2 then the spreading process stop when v_1, v_2, v_4, v_5 are affected by the cascading failure. For each given size of k, after placing the protection nodes, we simulate the cascading failure for 2,000 times, and in each simulation the source node is uniformly chosen from G_N, i.e., each with probability $1/N$. For comparison, we use the heuristic algorithm based on the *degree centrality* which is often used as a baseline algorithm for performance comparison in social network analysis [31], [32]. Now, nodes with a larger degree are similar to “hubs” as mentioned in [33]. For the degree-based heuristic, we first sort all the nodes in G_N based on their degree from the largest to the smallest, and then construct V_p by selecting the first k nodes in the sorted order. In the following simulation results, we use the percentage of number of nodes to display the results due to the difference in network size.

1) Tree Network With $N = 4941$: Here we provide simulation results on a general tree randomly generated by a well-known preferential attachment based model: “Barabási- Albert model”. We input different sizes of k to observe the performance of the two algorithms. We can observe that in the tree network, Algorithm 3 outperforms the degree centrality heuristic in each size of k (cf. Fig. 6 and Table II).

2) Real World Network With $N = 4941$: In this part of simulations, G_N is the western United State power grid network analyzed in [34]. Each node 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 and the average degree is 2.66. Note that the scale of y-axis is different from the random-tree simulation. When $|k| \le 0.05N$, the performance is worse than the tree network due to the connectivity of the power grid network. When $|k| \ge 0.06N$, the average number of n drop to $0.0348N$, which implies that we only need to protect 6% of G_N to reach such performance. In this network, 3 outperforms the degree centrality heuristic in each size of k (cf. Fig. 7 and Table III).

3) Real World Network With $N = 1508$: In this part of simulations, G_N is an Europe renewable power system provided in [35]. Each node in G_N represents a power plant and each edge represents a transmission line. Note that, the original graph is a directed graph but we ignore the direction on each edge here. The average degree in this network is around 2.88. We can observe that when $|k| \le 0.06N$, the performance of Algorithm 3 is worse than the degree-centrality heuristic. However, when $|k| \ge 0.07N$, the size of n drops extremely and Algorithm 3 outperforms the degree-centrality heuristic (cf. Fig.8 and Table IV).

4) Real World Network with $N = 1706$: In this part of simulations, G_N is a human protein interaction network analyzed in [36]. Each node in G_N represents a human protein and each edge represents there is an interaction between two human proteins. There are 6207 edges in this human protein interaction network and the average degree is around 7.28. Some of the critical infrastructure networks have small average degree due to the geographical constraint, i.e, it can be embedded on a plane (planar graph) [3]. Examples of such geographically-constrained networks are water supply networks, road traffic networks or some of the power transmission networks. The average degree of a planar graph is strictly less than 6, hence we select this human protein network that its average degree is much higher than the US power grid network or the EU power plant network but slightly larger than 6 to test out the performance of Algorithm 3.

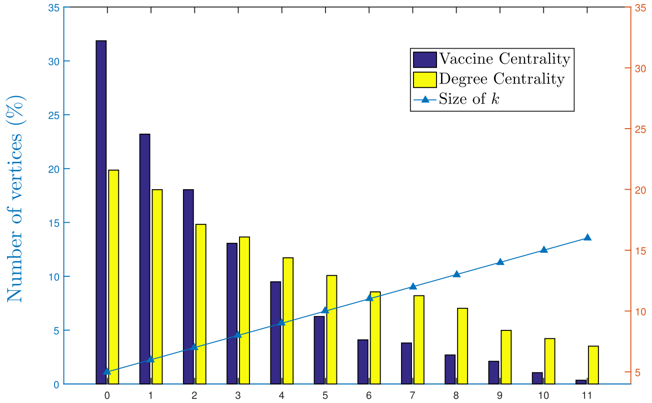

Fig. 9. Simulation results when G_N is a real world network: Human Protein Interaction Network. The y-axis represents the number of nodes in percentages and the x-axis represents each trial with different sizes of k.

TABLE V
DETAILS OF SIZE OF $|k|$ AND THE AVERAGE NUMBER OF NODES n AFFECTED BY THE CASCADING FAILURE OVER 2000 SIMULATIONS ON THE REAL WORLD HUMAN PROTEIN INTERACTION NETWORK

$\|k\|$	5	6	7	9	9	10
Algorithm3	31.9	23.2	18.0	13.1	9.50	6.26
DegreeCentrality	19.9	18.0	14.8	13.6	11.7	10.1
$\|k\|$	9	10	11	12	13	14
Algorithm3	4.10	3.81	2.69	2.11	1.05	0.35
DegreeCentrality	8.55	8.20	7.03	4.98	4.22	3.51

In Fig. 8, we can observe that when $|k| \geq 0.08N$, Algorithm 3 outperforms the degree-centrality heuristic, moreover, as the size of k growing the ratio between their performance is getting smaller. For example, when $|k| = 0.05N$, the ratio of the performance of two algorithms is $\frac{31.9}{19.9} = 1.6$, as k grows to $0.14N$ the ratio becomes $\frac{0.35}{3.51} = 0.1$, which implies Algorithm 3 is ten times better than the degree-centrality heuristic (cf. Fig. 9 and Table V).

VI. Further Modeling and Algorithmic Extensions

In this section, we discuss several extensions of the work in this paper. The first one is to extend the original homogeneous *SI* spreading model to aheterogeneous Susceptible-Infectious-Susceptible (*SIS*) spreading model. In a heterogeneous network, each node has different infection rate that can be modeled by a real value. The state change of a susceptible node can then be modeled by a continuous Markov chain and the probability of the state change is therefore related to its neighbors' assigned values. In this paper, each possible linear extension has the same spreading probability. However, in a heterogeneous network, each possible linear extension has a different spreading probability which is dependent on the values assigned to the nodes. Hence, simply choosing the node with the maximum number of linear extensions to be vaccinated may not be ideal. Instead, we need to find the vaccine nodes by considering both the number of linear extensions and the assigned values. If we use the discrete spreading model as considered in this paper, then this is a much harder computational problem, but our algorithms in this paper can potentially be leveraged to design efficient heuristics for the general heterogeneous problem.

The second direction is to consider incomplete information of the network topology due possibly to insufficient data. To address this, we can partition all edges into two sets, in which one is the set that cannot be observed and the other one that is observable. We can still find a protection set based on those observable edges, but it is challenging to determine performance guarantees due to the lack of information. One potential solution is to dynamically react to the ongoing cascading failure by combining our algorithms with existing heuristics, e.g., the work in [18]. The third direction is to consider a new definition of protected nodes where protected nodes are more "influential" than the one in this paper. Consider r_v to be the subset of v's neighbors that will be protected when v is chosen to be protected. This models the phenomenon in epidemiology where nodes with vaccine can affect neighbors. In other words, this paper only studies the special case with $|r_v| = 0$ for all $v \in G_N$. Consider another special case such that for any two vertices v_a and v_b, if $v_a \in r_{v_b}$ then $v_b \in r_{v_a}$, i.e., if we vaccine v_a then v_b will be protected and vice versa. Under this assumption, if $|r_v| = 1$ for all $v \in G_N$, then this new problem can be reduced to the one in this paper by performing edge contractions on the given graph. The edge contraction rule is as follows: For each vertex $v_a \in G$, if $v_b \in r_{v_a}$, then we do the edge contraction on the edge (a, b). The resultant graph G' is a graph with fewer vertices and edges, and we can apply the algorithms in this paper on G'. However, the case where $|r_v|$ is not a constant for each $v \in G_N$ remains open. It will be most interesting and challenging to extend the algorithms and analyses to handle more general cases.

VII. Network Centrality as Statistical Inference

As is the case with [8], this paper can be viewed as a case study of the "*network centrality as statistical inference*" approach to inferential statistical analysis. In particular, this approach addresses optimality guarantees to stochastic optimization problems in inferential statistics by transforming them to graph-theoretic problems. An appropriate network centrality induces a metric on each graph node, and brings graph algorithm machinery to bear on solving the stochastic program. If network centrality schemes are to be useful compact measures of the importance of nodes in the network, then they should have the statistical basis to accurately capture the optimality of stochastic optimization and to serve as a computational tool for finding the optimal solution. Network centrality as statistical inference may thus be quantified in both the *reverse engineering* and *forward engineering* directions.

A. Reverse Engineering

In the case of *reverse engineering*, we ask: Given a well-known network centrality, what are the statistical inference optimization problems that it implicitly solves? Both the distance centrality and the branch weight centrality, as shown in this paper, can solve the rumor source detection problem in [8]. The betweenness centrality solves the special case of a single-vaccine estimation problem in this paper. The appropriate network centrality can succinctly capture the effect from the addition or removal of nodes as well as changes in stochastic processes on the graph and thus be connected to perturbation analysis in stochastic programming. Network centrality algorithms can compute exact or approximate solution to the statistical inference optimization problems. It can provide guiding principles on algorithm design and computational complexity for statistical inference in the finite and asymptotically large graph regimes that are applicable to large data sets.

B. Forward Engineering

In the case of *forward engineering*, we ask: Given a stochastic optimization formulation over a network, how to transform it or to decompose it to one whose subproblems are graph-theoretic and can utilize network centrality, then solve or approximate the overall problem? In [8], the rumor centrality is defined over degree-regular networks to solve a maximum likelihood estimation problem. In the case of this work, vaccine centrality is defined on the *tree abstract data type* induced by the underlying network topology as an approximation algorithm to solve a statistical estimation problem. New algorithms can be designed based on message-passing (belief propagation) graph analysis.

Finally, this paper considers network centrality as statistical inference only from a static network viewpoint. It will be important to generalize this to time-dependent networks or even the availability of network data that changes over time [37]. Finding the appropriate network centrality to explain flow patterns or temporal scales of changes in the network to solve stochastic optimization problems concerning the network is especially interesting. Other than viewing network centrality as inferential statistics, there are broader implications. For example, the framework of network centrality as statistical inference can be applied to many abstract data types in computer science and to the study of probability on trees and graphs. Also, there are connections between network centrality as statistical inference and graph signal processing [38], [39], which include methods for sampling, filtering or learning over graphs. The confluence of these research directions can lead to mathematically rigorous graph analytics for analyzing statistics problems in large networks.

VIII. Conclusion

We studied the problem of averting cascading failures in networked infrastructures by the strategic placement of protection nodes to mitigate systemic risks due to cascading failures. A stochastic optimization problem over a network was formulated and then solved using the vaccine centrality, which is based on minimizing the expected size of epidemic spread graphs subject to partially ordered set constraints. We first studied the connection between the graph centroid and partially ordered sets as a means of modeling causality for inference of the most probable source of cascading failures. We then exploited the graph centroid as the central concept in our algorithms to solve the problem. Several special cases including the single vaccine case were solved optimally. For the general case, we leveraged the centroid decomposition to propose computationally-efficient message passing algorithms that can be parameterized recursively based on the vaccine centrality to place protection nodes. In addition, asymptotic results for infinitely-large graphs and thus applicable to large data set were established using analytic combinatorics. We also proposed efficient computational heuristics for the placement of protection nodes in general graphs that have cycles. Lastly, this paper illustrated the conceptual simplicity of network centrality as statistical inference for optimality characterization and graph algorithm design to solve stochastic optimization problems in large networks.

Appendix

A. Proof of Theorem 2

Let G_N be a tree of size N and $v \in G_N$. Observe the following directions. Let us prove ($1 \Rightarrow 2$): We prove it by the contrapositive argument. Suppose v is not a rumor center, by Theorem 1 there is a branch of v, say T_u^v, with order $> N/2$ and u is adjacent to v. Now, we need a relationship between $\sum_{s\in G_N} d(v,s)$ and $\sum_{s\in G_N} d(u,s)$ as described by

$$\sum_{s\in G_N} d(v,s) = \sum_{s\in G_N} d(u,s) + (t_u^v - 1) - (t_v^u - 1).$$

We have $\sum_{s\in G_N} d(v,s) > \sum_{s\in G_N} d(u,s)$, since $t_u^v > t_u^v$. This implies that v is not a distance center.

Next, let us prove ($2 \Rightarrow 3$): First, we need the following fact: If all v's branches are of size $\leq N/2$, then v is the centroid. Again, by contrapositive argument, suppose v is not a centroid, then there exists a branch of v whose size $> N/2$ by Theorem 1.

Lastly, let us prove ($3 \Rightarrow 1$): Suppose v is a centroid, then each of all its branches is of order $\leq N/2$. This implies that v is a rumor center. Let $u \in G_N$, if u is adjacent to v, then $\sum_{s\in G_N} d(v,s) < \sum_{s\in G_N} d(u,s)$ and we finish the proof. If u is not adjacent to v, then we can partition all the nodes in G_N into three sets. The first one is T_v^u, the second one is T_u^v and the last one contains all the nodes not in T_v^u and T_u^v, say R. Let l denote $d(u,v)$. Now, consider $\sum_{s\in G_N} d(v,s) - \sum_{s\in G_N} d(u,s) = (\sum_{s\in T_v^u} d(v,s) + \sum_{s\in T_u^v} d(v,s) + \sum_{s\in R} d(v,s)) - (\sum_{s\in T_v^u} d(u,s) + \sum_{s\in T_u^v} d(u,s) + \sum_{s\in R} d(u,s))$.

Since v is the centroid, we have :

1) $|R| + t_u^v \leq N/2$, and $t_v^u > N/2$;
2) $(\sum_{s\in T_u^v} d(v,s) + \sum_{s\in T_v^u} d(v,s)) - (\sum_{s\in T_u^v} d(u,s) + \sum_{s\in T_v^u} d(u,s)) = l \cdot (t_u^v - t_v^u)$;
3) $|\sum_{s\in R} d(v,s) - \sum_{s\in R} d(u,s)| \leq l \cdot |R|$.

Combining these three properties, we conclude that $\sum_{s\in G_N} d(v,s) - \sum_{s\in G_N} d(u,s) < 0$, for any $u \in G_N$, that is, v is the distance center. ■

B. Proof of Theorem 4

Let G_N be a tree of size N, and $u, v \in G_N$. Observe the following directions. Let us prove ($1 \Rightarrow 2$): Suppose $L(v, G_N) \geq L(u, G_N)$, we have $\mathscr{D}(v, G_N) = \mathscr{D}(u, G_N) - t_v^u + t_u^v$ and $t_v^u \geq t_u^v$, and so we conclude that $\mathscr{D}(v, G_N) \leq \mathscr{D}(u, G_N)$.

Next, let us prove ($2 \Rightarrow 3$): Suppose $\mathscr{D}(v, G_N) \leq \mathscr{D}(u, G_N)$, we have $\mathscr{D}(v, G_N) - \mathscr{D}(u, G_N) = t_u^v - t_v^u \leq 0$. This implies that $t_u^v \leq t_v^u$. Note that $\mathsf{weight}(u) = t_v^u$. If not, then there is a branch of u with size larger than t_v^u thereby implying $t_u^v \geq t_v^u$, which is a contradiction. Hence, we have $\mathsf{weight}(u) = t_v^u \geq \mathsf{weight}(v)$.

Lastly, let us prove ($3 \Rightarrow 1$): Suppose $\mathsf{weight}(v) \leq \mathsf{weight}(u)$, and note that $\mathsf{weight}(u) = t_v^u$. Since u is not the centroid, we have $t_v^u > N/2$ and so $t_u^v \leq N/2$, this implies that $L(v, G_N) \geq L(u, G_N)$. ■

C. Proof of Corollary 1

Let G_N be a tree with N vertices. Let σ_{su} denote the total number of shortest paths from node s to u and $\sigma_{su}(v)$ denote the number of those paths which pass through v. Then the betweenness centrality $\mathcal{B}(v)$ of v can be defined as

$$\mathcal{B}(v) = \sum_{s \neq u \neq v} \frac{\sigma_{su}(v)}{\sigma_{su}}. \tag{11}$$

We have the fact that $\sigma_{su}(v)$ is either 1 or 0 for all s, t and $v \in G_N$, since G_N is a tree. Also, we have σ_{su} is either 1 or 0 for all s, $u \in G_N$. Let $t_{v_j}^{v_i}$ be defined as in previous sections. Then, we can rewrite (11) as

$$\mathcal{B}(v) = \frac{1}{2} \sum_{s,u \in \mathsf{neighbor}(v), s \neq u} t_s^v \cdot t_u^v. \tag{12}$$

Note that the objective function $f(\{V_P\})$ in (5) is equivalent to $\sum_{s \in \mathsf{neighbor}(v)} (t_s^v)^2$ when G_N is a tree and $V_P = \{v\}$. By combining $f(\{V_P\})$ and $\mathcal{B}(v)$, we have

$$\begin{aligned} 2\mathcal{B}(v) + f(v) &= \sum_{s,u \in \mathsf{neighbor}(v)} t_s^v \cdot t_u^v \\ &= \left(\sum_{s \in \mathsf{neighbor}(v)} (t_s^v) \right)^2 \\ &= (N-1)^2. \end{aligned}$$

Hence, the optimization problem (5) can be written as

$$\underset{v \in G_N}{\text{minimize}} \quad (N-1)^2 - 2\mathcal{B}(v),$$

which is equivalent to

$$\underset{v \in G_N}{\text{maximize}} \quad \mathcal{B}(v).$$

We can conclude that the node v with the maximum betweenness centrality is the optimal solution for (5) when G_N is a tree and $|V_P| = 1$. ■

D. Proof of Theorem 6

Let G_N, V_p and k be defined as in Theorem 6. By the definition of the graph centroid of a given tree T, at each time that we remove the centroid from T, the size of the maximally connected component is at most $T/2$. This implies that when $k = 1$, the worst-case optimal value of the objective function in (5) is $(\frac{N}{2})^2 + (\frac{N}{2} - 1)^2$. When $k = 2$, the expectation $\mathbf{Expect}(|G_n|)$ in (5) is $(\frac{N}{4})^2 + (\frac{N}{4} - 1)^2 + (\frac{N}{2} - 1)^2$. To simplify the calculation, we ignore the constant term, so that $\mathbf{Expect}(|G_n|)$ in (5) is $(\frac{N}{2})^2 + (\frac{N}{2})^2$.

Now, let us consider the general case of the worst-case optimal value. Given any integer $k > 0$, then there is an integer t such that $2^t \leq k \leq 2^{t+1}$. In particular, the expectation $\mathbf{Expect}(|G_n|)$ of the worst-case corresponding to the given k satisfies

$$\begin{aligned} &N^2 \left[(2^{t+1} - 1 - k) \cdot \frac{1}{2^{2t}} + (k + 1 - (2^{t+1} - 1 - k)) \cdot \frac{1}{2^{2t+2}} \right] \\ &= N^2 \left[\frac{2^{t+2} - 2^t - k - 1}{2^{2t+1}} \right] < N^2 \left[\frac{3 - \frac{k}{2^t} - \frac{1}{2^t}}{k+1} \right] < \frac{2N^2}{k+1}. \end{aligned}$$

In the optimal case, all the protection nodes evenly divide G_N into several connected components each having a size equal to 1. Therefore, the expectation $\mathbf{Expect}(|G_n|) \leq \sum_{v_i \notin V_P} 1^2$ which is $N - k$. Let $0 < c < 1$ be a constant such that $k = c \cdot N$. Then, the ratio between the worst-case $\mathbf{Expect}(|G_n|)$ and the optimal $\mathbf{Expect}(|G_n|)$ is at most $\frac{2}{(c + \frac{1}{N})(1-c)}$, hence is bounded above by $\frac{2}{c(1-c)}$. ■

E. Proof of Theorem 7

Consider a degree regular tree G_N with degree d, let A_d and B_d be two sets such that

$$A_d = \left\{ (a_1, a_2, \ldots, a_d) | 0 \leq a_i \leq \frac{N}{2}, \sum_{i=1}^{d} a_i = N - 1 \right\}$$

$$B_d = \left\{ (b_1, b_2, \ldots, b_d) | b_i \in \mathbf{N} \cup \{0\}, \sum_{i=1}^{d} b_i = N - 1 \right\}.$$

Note that, these two sets correspond to the orders of the branch sizes of a node. And if the branch size is in A_d then it is the centroid. We have $|B_d| = \binom{N-1+d-1}{d-1}$, and let $S_k = \{(s_1, s_2, \ldots, s_d) | s_k > \frac{N}{2}, 0 \leq s_j < \frac{N}{2}, \sum_{i=1}^{d} s_i = N - 1\}$, so $|S_i| = \binom{d + \lceil \frac{N}{2} \rceil - 3}{d-1}$ and

$$\begin{aligned} |A_d| &= |B_d| - \sum_{k=1}^{d} |S_k| \\ &= \binom{N+d-2}{d-1} - d \cdot \binom{\lceil \frac{N}{2} \rceil + d - 3}{d-1} \end{aligned}$$

since $S_i \cap S_j = \phi$ for $i \neq j$.

Let A_3 and B_3 be defined as above and $z_d(i) = (d-2)(i-1) + 1$. By (2), we have

$$\begin{aligned} k^{(3)} &= \frac{\sum_{(t_{u_1}^v, t_{u_2}^v, t_{u_3}^v) \in A_3} \left(\prod_{k=1}^{3} \frac{\prod_{i=1}^{t_{u_k}^v} (z_d(i))}{t_{u_k}^v !} \right)}{\sum_{(t_{u_1}^v, t_{u_2}^v, t_{u_3}^v) \in B_3} \left(\prod_{k=1}^{3} \frac{\prod_{i=1}^{t_{u_k}^v} (z_d(i))}{t_{u_k}^v !} \right)} \\ &= \frac{\sum_{(t_{u_1}^v, t_{u_2}^v, t_{u_3}^v) \in A_3} 1}{\sum_{(t_{u_1}^v, t_{u_2}^v, t_{u_3}^v) \in B3} 1} = \frac{|A_3|}{|B_3|}. \end{aligned}$$

Finally, we have

$$\frac{|A_3|}{|B_3|} = \frac{\binom{N+1}{2} - 3 \cdot \binom{\lceil \frac{N}{2} \rceil}{2}}{\binom{N+1}{2}}.$$

Thus, $\lim_{N \longrightarrow \infty} k^{(3)} = \frac{1}{4}$. Note that it is immaterial whether N is even or odd in the asymptotic regime. ■

F. Proof of Theorem 8

Proof: We consider a cascading failure on d-regular graphs. First, we introduce some notations. Let $\tilde{\mathcal{T}}_N$ denote the tree after the cascading failure has spread to N nodes. We label the nodes with numbers from the set $\{1, 2, \dots, N\}$ according to the very first instance when a node is affected by the cascading failure. Thus, the node with label 1 is the source and it is the only node with an out-degree d whereas all the other nodes have out-degree $d-1$ (if $\tilde{\mathcal{T}}_n$ is drawn as a rooted tree in the usual way). Next, observe that all the d subtrees of this source are $(d-1)$-ary increasing trees (increasing trees are labeled trees with label sequences from the root to any leaf increasing). Define the notations:

$$T_N = \text{number of } (d-1)\text{-ary increasing trees},$$

$$T(z) = \sum_{N \geq 1} T_N \frac{z^N}{N!},$$

where $T(z)$ is the generating function of T_N. It is easy to derive that

$$T(z) = -1 + (1-(d-2)z)^{-1/(d-2)}. \tag{13}$$

Next, define the notations:

$$\tilde{T}_N = \text{number of possible } \tilde{\mathcal{T}}_N,$$

$$\tilde{T}(z) = \sum_{N \geq 1} 2\tilde{T}_N \frac{z^N}{N!}.$$

Then,

$$\tilde{T}_N = \frac{1}{2} \prod_{i=1}^{N} [2 + (d-2)(i-1)]$$

$$\tilde{T}(z) = \ -1 + (1-(d-2)z)^{-2/(d-2)} \tag{14}$$

We are interested in the case that the labeled-1 node, i.e., v_1, is the centroid of $\tilde{\mathcal{T}}_N$. We have

$$\frac{L(v_1, G_N)}{\sum\limits_{v \in G_N} L(v, G_N)}$$
$$= 1 - d(\text{ratio of the case that one branch size of } v_1 \geq N/2).$$

Next, we fix one subtree of the source of $\tilde{\mathcal{T}}_n$, say t_v^1, and denote by I its size. Let j_i represent the size of the ith subtree of $\tilde{\mathcal{T}}_n$ and $R(I=j)$ denote the ratio mentioned in (10) but with the case that v_1 is not the centroid since the branch size $t_v^1 > N/2$. Obviously,

$$R(I=j) \tag{15}$$
$$= \frac{1}{\tilde{T}_N} \sum_{j+j_2+j_3+\ldots+j_d=N-1} \binom{N-1}{j, j_2, j_3, \dots, j_d} T_j T_{j_2} T_{j_3} \dots T_{j_d}$$
$$= \frac{(N-1)!T_j}{j!\tilde{T}_N} \sum_{j_2+j_3+\ldots+j_d=N-1-j} \frac{T_{j_2}}{j_2!} \frac{T_{j_3}}{j_3!} \cdots \frac{T_{j_d}}{j_d!}$$
$$= \frac{(N-1)!T_j}{j!\tilde{T}_N} [z^{N-1-j}](1+T(z))^{d-1}$$
$$= \frac{(N-1)!T_j}{j!\tilde{T}_N} [z^{N-1-j}](1-(d-2)z)^{-\frac{d-1}{d-2}}$$
$$= \frac{(N-1)!T_j}{j!\tilde{T}_N} (d-2)^{N-1-j} [z^{N-1-j}](1-z)^{-\frac{d-1}{d-2}}$$
$$= \frac{(N-1)!T_j}{j!\tilde{T}_N} (d-2)^{N-1-j} \frac{(N-1-j)^{\frac{1}{d-2}}}{\Gamma(\frac{d-1}{d-2})}. \tag{16}$$

In the sequel, we need the following important lemma from analytic combinatorics.

Theorem 10 ([40, Thm. VI.1]): For $\alpha \in \mathbb{C} \setminus \mathbb{Z}_{\leq 0}$ set

$$f(z) := (1-z)^{-\alpha}.$$

Then, as $N \to \infty$,

$$[z^N]f(z) \sim \frac{N^{\alpha-1}}{\Gamma(\alpha)} \left(1 + \sum_{k=1}^{\infty} \frac{e_k(\alpha)}{N^k}\right),$$

where $e_k(\alpha)$ is a polynomial of degree $2k$.

a) Asymptotics: We now turn to deducing the asymptotical result in the following computation based on $N \to \infty$.

First, we apply Theorem 10 to (13) to obtain

$$T_N \sim \frac{N! \cdot N^{\frac{2-d}{d-1}}}{\Gamma(\frac{1}{d-1})} (d-1)^N \qquad \text{as } N \to \infty.$$

Similarly, we apply Theorem 10 to (14) to obtain

$$\tilde{T}_N \sim \frac{N! \cdot N^{\frac{4-d}{d-2}}}{2\Gamma(\frac{2}{d-2})} (d-2)^N \qquad \text{as } N \to \infty.$$

By Theorem 1, we need to compute

$$\sum_{N/2 \leq j \leq N-1} R(I=j),$$

where $R(I=j)$ is given by (16). Therefore, we again use Theorem 10 and the expansions for T_N and $\tilde{T}_N$ from above.

This gives

$$\begin{aligned}
&\sum_{N/2\le j\le N-1} R(I=j)\\
&\sim \frac{(N-1)!}{\tilde{T}_N}\sum_{N/2\le j\le N-1}\frac{T_j}{j!\Gamma(\frac{d-1}{d-2})}(d-2)^{N-1-j}(N-1-j)^{\frac{1}{d-2}}\\
&\sim \frac{(N-1)!\cdot 2\Gamma(\frac{2}{d-2})}{N!\cdot N^{\frac{4-d}{d-2}}(d-2)^N}\sum_{N/2\le j\le N-1}\left[\frac{\frac{j!\cdot j^{\frac{3-d}{d-2}}}{\Gamma(\frac{1}{d-2})}(d-2)^j}{j!\Gamma(\frac{d-1}{d-2})}\cdot\right.\\
&\left.(d-2)^{N-1-j}(N-1-j)^{\frac{1}{d-2}}\right] \sim \frac{2\Gamma\frac{2}{d-2}}{(d-2)\cdot N^{\frac{2}{d-2}}\Gamma(\frac{1}{d-2})\Gamma(\frac{d-1}{d-2})}\\
&\cdot\sum_{N/2\le j\le N-1}[j^{\frac{1}{d-2}-1}(N-1-j)^{\frac{1}{d-2}}]\\
&\sim \frac{2\Gamma\frac{2}{d-2}}{(d-2)\cdot N^{\frac{2}{d-2}}\Gamma(\frac{1}{d-2})\Gamma(\frac{d-1}{d-2})}\cdot N^{\frac{2}{d-2}-1}\\
&\cdot\sum_{N/2\le j\le N-1}\left(\frac{j}{N}\right)^{\frac{1}{d-2}-1}\left(\frac{N-1-j}{N}\right)^{\frac{1}{d-2}}\\
&\sim \frac{2\Gamma(\frac{2}{d-2})}{(d-2)\Gamma(\frac{1}{d-2})\Gamma(\frac{d-1}{d-2})}\int_{1/2}^{1}x^{\frac{1}{d-2}-1}(1-x)^{\frac{1}{d-2}}\,dx
\end{aligned}$$

as $N \to \infty$.

Lemma 1: For $\alpha > 0$,

$$\int_{1/2}^{1} x^{\alpha-1}(1-x)^{\alpha}\mathrm{d}x = \frac{1}{2}\left(B(\alpha,\alpha+1) - \frac{1}{\alpha 2^{2\alpha}}\right),$$

where $B(\cdot,\cdot)$ denotes the beta function.

Proof: First, observe that

$$\begin{aligned}
&B(\alpha,\alpha+1)\\
&= \int_0^1 x^{\alpha-1}(1-x)^{\alpha}\mathrm{d}x\\
&= \int_0^{1/2} x^{\alpha-1}(1-x)^{\alpha}\mathrm{d}x + \int_{1/2}^{1} x^{\alpha-1}(1-x)^{\alpha}\mathrm{d}x.
\end{aligned}$$

Now, call the first and second integral on the right-hand side $\mathscr{L}$ and $\mathscr{R}$, respectively. Using integration by parts and substitution, we have

$$\mathscr{L} = \frac{1}{\alpha}x^{\alpha}(1-x)^{\alpha}\Big|_0^{1/2} + \mathscr{R}.$$

Thus,

$$\mathscr{R} = \frac{1}{2}\left(B(\alpha,\alpha+1) - \frac{1}{\alpha 2^{2\alpha}}\right)$$

which is the claimed result. ■

Finally, we combine the above results to yield the asymptotic expression of the ratio $k^{(d)}$ in d-regular trees in the limit:

$$\begin{aligned}
\lim_{N\longrightarrow\infty} k^{(d)} &= 1 - d\cdot\sum_{N/2\le j\le N-1} R(I=j)\\
&= 1 - \frac{d}{2} + \frac{(d-2)\cdot\Gamma\left(\frac{d}{d-2}\right)}{2^{\frac{d}{d-2}}\cdot\Gamma\left(\frac{1}{d-2}\right)\Gamma\left(\frac{d-1}{d-2}\right)}
\end{aligned}$$

■

References

[1] P. D. Yu, C. W. Tan, and H. L. Fu, "Graph algorithms for preventing cascading failures in networks," in *Proc. 52nd Annu. Conf. Inf. Syst. Sci.*, 2018, pp. 1–6.

[2] S. Erjongmanee, C. Ji, and J. Momoh, "Inferring network-power cascading disruptions and sustainability," in *Proc. Int. Conf. Neural Netw.*, Aug. 2011, pp. 3063–3067.

[3] A. Bashan, Y. Berezin, S. V. Buldyrev, and S. Havlin, "The extreme vulnerability of interdependent spatially embedded networks," *Nature Phys.*, vol. 9, pp. 667–672, Aug. 2013.

[4] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Network forensics: Random infection vs. spreading epidemic," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2012, pp. 223–234.

[5] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE INFOCOM*, 2005, pp. 1455–1466.

[6] H. Khamfroush, N. Bartolini, T. F. L. Porta, A. Swami, and J. Dillman, "On propagation of phenomena in interdependent networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 3, no. 4, pp. 225–239, Oct./Dec. 2016.

[7] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, pp. 1025–1028, Apr. 2010.

[8] D. Shah and T. Zaman, "Rumors in a network: Whos's the culprit?," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, Aug. 2011.

[9] W. Dong, W. Zhang, and C. W. Tan, "Rooting out the rumor culprit from suspects," in *Proc. IEEE Int. Symp. Inf. Theory*, 2013, pp. 2671–2675.

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

[11] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rumor source detection with multiple observations: Fundamental limits and algorithms," *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2014, pp. 1–3.

[12] 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 J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 663–677, Jun. 2015.

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

[14] L. Zheng and C. W. Tan, "A probabilistic characterization of the rumor graph boundary in rumor source detection," in *Proc. IEEE Digital Signal Process.*, 2015, pp. 765–769.

[15] M. Fuch and P. D. Yu, "Rumor source detection for rumor spreading on random increasing trees," *Electron. Commun. Probab.*, vol. 20, no. 2, p. 12, 2015.

[16] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Distinguishing infections on different graph topologies," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3100–3120, Jun. 2015.

[17] G. Fanti, P. Kairouz, S. Oh, K. Ramchandran, and P. Viswanath, "Hiding the rumor source," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6679–6713, Oct. 2017.

[18] D. Z. Tootaghaj, N. Bartolini, H. Khamfroush, and T. L. Porta, "Controlling cascading failures in interdependent networks under incomplete knowledge," in *Proc. IEEE 36th Symp. Reliable Distrib. Syst.*, Sep. 2017, vol. 3, pp. 54–63.

[19] J. Omi, J. Martin-Hernandez, and P. V. Mieghem, "Network protection against worms and cascading failures using modularity partitioning," in *Proc. 22nd Int. Teletraffic Congr.*, Sep. 2010, vol. 3, pp. 1–8.

[20] K. Drakopoulos, A. Ozdaglar, and J. Tsitsiklis, “An efficient curing policy for epidemics on graphs,” *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 2, pp. 67–75, Jul./Dec. 2015.

[21] P. J. Slater, “Fault-tolerant locating-dominating sets,” *Discrete Math.*, vol. 249, pp. 179–189, Apr. 2002.

[22] N. T. Bailey, *The Mathematical Theory of Infectious Diseases and Its Applications*. London, U.K.: Griffin, 1975.

[23] B. Iriarte, “Graph orientations and linear extensions,” *Math. Oper. Res.*, vol. 42, no. 4, pp. 1219–1229, 2017.

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

[25] O. Ore, *Theory of Graphs*. Providence, RI, USA: Am. Math. Soc., 1962.

[26] D. J. C. Mackay, *Information Theory, Inference and Learning Algorithms*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[27] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Soc. Netw.*, vol. 1, pp. 215–239, 1978–1979.

[28] U. Brandes, “A faster algorithm for betweenness centrality,” *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.

[29] A. J. Goldman, “Optimal center location in simple networks,” *Transp. Sci.*, vol. 5, pp. 212–221, May 1971.

[30] N. Megiddo, “An $o(nlog_2 n)$ algorithm for the kth longest path in a tree with applications to location problems,” *SIAM J. Comput.*, vol. 10, pp. 328–337, May 1979.

[31] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2003, pp. 137–146.

[32] G. Erkan and D. R. Radev, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, Apr. 2004.

[33] Z. Dezso and A. L. Barabasi, “Halting viruses in scale-free networks,” *Phys. Rev. E*, vol. 65, 055103(R) – Published, May 2002.

[34] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol. 393, pp. 440–442, Jun. 1998.

[35] T. V. Jensen and P. Pinson, “Re-europe, a large-scale dataset for modeling a highly renewable european electricity system,” *Nature Sci. Data*, vol. 4, Nov. 2017, Article no: 170175.

[36] U. Stelzl *et al.*,“A human protein-protein interaction network: a resource for annotating the proteome,” *Cell*, vol. 122, pp. 957–968, Sep. 2005.

[37] D. Domenico, G. Manlio, C. Granell, M. Porter, and A. Arenas, “The physics of spreading processes in multilayer networks,” *Nature Phys.*, vol. 12, pp. 901–906, 2016.

[38] A. Ortega, P. Frossard, J. Kovacevic, J. M. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges and applications,” *Proc. IEEE*, vol. 106, no. 5, pp. 108–828, May 2018.

[39] X. Yan, B. M. Sadler, R. J. Drost, P. L. Yu, and K. Lerman, “Graph filters and the z-laplacian,” *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 774–784, Sep. 2017.

[40] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

Pei-Duo Yu received the B.Sc. and M.Sc. degrees in applied mathematics from the National Chiao Tung University, Hsinchu, Taiwan in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include combinatorics counting, graph algorithms, optimization theory, and its applications.

Chee Wei Tan (M’08–SM’12) received the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 2006 and 2008, respectively. He is an Associate Professor of computer science with the City University of Hong Kong, Hong Kong. He was a Postdoctoral Scholar with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA. He was a Visiting Faculty with Qualcomm R&D, San Diego, CA, USA, and the Tencent AI Lab, and is a Senior Fellow with the Institute for Pure and Applied Mathematics for the program on “Science at Extreme Scales: Where Big Data Meets Large-Scale Computing.” His research interests include networks and graph analytics, algorithms at the interface of computer science and statistics, and convex optimization theory and its applications. He was the recipient of the 2008 Princeton University Wu Prize for Excellence. He was the Chair of the IEEE Information Theory Society Hong Kong Chapter and received the Chapter of the Year Award. He was twice selected to participate at the U.S. National Academy of Engineering China-America Frontiers of Engineering Symposium. He currently serves as an Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING.

Hung-Lin Fu (M’13) received the B.S. degree in mathematics from National Taiwan Normal University, Taipei City, Taiwan, in 1973 and the Ph.D. degree in mathematics (major in combinatorics) from Auburn University, Auburn, AL, USA, in 1980. He is currently a Fellow of the Institute of Combinatorics and Its Applications, and a Professor with the Department of Applied Mathematics, National Chiao Tung University, Hsinchu, Taiwan, since 1988. His interests include graph theory, combinatorial designs, and their applications.