Detecting Critical Nodes in Interdependent Power Networks for Vulnerability Assessment

Dung T. Nguyen, Yilin Shen, *Student Member, IEEE*, and My T. Thai, *Member, IEEE*

***Abstract*—Power networks and information systems become more and more interdependent to ensure better supports for the functionality as well as improve the economy. However, power networks also tend to be more vulnerable due to the cascading failures from their interdependent information systems, i.e., the failures in the information systems can cause the failures of the coupled portion in power networks. Therefore, the accurate vulnerability assessment of interdependent power networks is of great importance in the presence of unexpected disruptive events or adversarial attacks targeting on critical network nodes. In this paper, we study the *Interdependent Power Network Disruptor* (IPND) optimization problem to identify critical nodes in an interdependent power network whose removals maximally destroy its functions due to both malfunction of these nodes and the cascading failures of its interdependent communication network. First, we show the IPND problem is NP-hard to be approximated within the factor of $(2-\epsilon)$. Despite its intractability, we propose a greedy framework with novel centrality functions based on the networks' interdependencies, to efficiently solve this problem in a timely manner. An extensive experiment not only illustrates the effectiveness of our approach on networks with different topologies and interdependencies, but also highlights some important observations which help to sharpen the robustness of interdependent networks in the future.**

***Index Terms*—Algorithm, computational complexity, experiments, interdependent power networks, vulnerability assessment.**

I. Introduction

THE RAPID development of technology has revolutionized the power networks and drastically increased their interdependencies with information systems. That is, power stations depend on communication networks for control and management and communication networks depend on power systems for their electricity support. In the meanwhile, such growing interdependencies also dramatically impact the vulnerability of power systems by being exposed to threats not only to themselves but also to the cascading failures induced by information systems. In a typical attacking point of view, an attacker would first exploit the network weaknesses, and then only needs to target on some critical nodes in either power networks or their interdependent communication networks, whose corruptions bring the whole network down to its knees. In other words, nodes from power networks depend heavily on nodes from their interdependent networks and vice versa. Consequently, when nodes from one network fail, they cause nodes in the other network to fail, too. For instance, an adversarial attack to any essential Internet hosts, e.g., tier-1 ISPs such as Qwest, AT&T or Sprint servers, once successful, may cause tremendous breakdowns to both millions of online services and the further large-area blackout because of the cascading failures. A real-world example is the wide-range blackout that affected the majority of Italy on 28 September 2003 [17], which resulted from the cascading failures induced by the dependence between power networks and communication networks. Therefore, in order to guarantee the robustness of power networks without reducing their performance by decoupling them from information systems, it is important to identify those critical nodes in interdependent power networks, beforehand.

There have been many studies assessing the network vulnerability [2], [3], [8], [10], [12], [15], [18]. Yet, these approaches are either designed only for single networks or heavily dependent on configuration models of interdependent networks. The existing approaches [1]–[3], [14] for single networks are based on various metrics, such as the degree of suspected nodes or edges [2], the average shortest path length [1], the global clustering coefficients [14], and the pairwise conductivity [2], [3] and so on. However, when applying into interdependent networks, their performances drop tremendously since these metrics fail to cast the cascading failures in interdependent networks. Later on, other researchers [8], [10], [12], [15], [18] studied the vulnerability assessment on interdependent networks, based on the size of largest connected component in power networks after cascading failures. Although they showed the effectiveness of this new metric, most of them focus on the artificial models of interdependent networks, i.e., random interdependency between networks, and ignore the detection of top critical nodes in real networks.

Let us consider a simple example of interdependent networks in Fig. 1, which illustrates a small portion of power network (lower nodes), communication network (upper nodes) and their interdependencies (dotted links). When we only take the single power network into account, the failure of u_7 destructs the power network more than that of u_1 since the largest connected component is of size 6 ($\{u_1, u_2, \ldots, u_6\}$) when u_7 fails, which is smaller than 9 ($\{u_2, \ldots, u_{10}\}$) when u_1 fails. However, if considering its interdependence upon the communication network, the failure of u_1 will destroy the power network more than that of u_7. This is because the failure of u_1 causes the failure of v_1 in the communication network, which further fails v_2, v_3, and v_4 since they are disconnected from the largest connected component. Due to their interdependence of the

Manuscript received April 01, 2012; revised September 05, 2012; accepted October 02, 2012. Date of publication January 08, 2013; date of current version February 27, 2013. This work is supported in part by DTRA, in part by the Young Investigator Award, and in part by the Basic Research Program HDTRA1-09-1-0061. Paper no. TSG-00176-2012.

The authors are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: dtnguyen@cise.ufl.edu; yshen@cise.ufl.edu; mythai@cise.ufl.edu). The first two authors, Dung T. Nguyen and Yilin Shen, contributed equally to this work.

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/TSG.2012.2229398

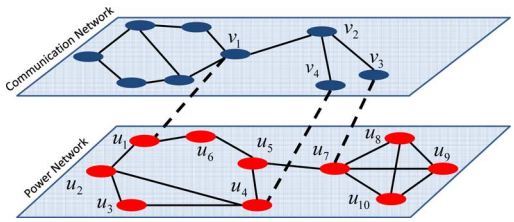


Fig. 1. Example of Interdependent Power Network and Communication Network.

nodes v_4 and u_7 in the power network, these cascading failures finally result in the largest connected component of the power network to be $\{u_8, u_9, u_{10}\}$ of size only 3. Yet, the largest connected component remains the same as in a single power network after the failure of u_7.

This example illustrates an important point that the role of one node could be totally different between single and interdependent networks with respect to the vulnerability assessment. Unfortunately, most of the existing works only illustrate the low disruption tolerance in random interdependent networks [8], [10], [12], [15], [18] and neglect to explore the significance of each node in general interdependent networks. In this paper, we study a new optimization problem, namely interdependent power network disruptor, to assess the vulnerability of generic power networks using the well-accepted measure, the size of the largest connected component [8], [10], [12], [15], [18], when a given number of nodes in power networks fail. We first show the non-trivial NP-hardness proof of the IPND problem, which also implies the inapproximability result. Due to its intractability, it seems unrealistic for one to quickly obtain optimal solutions for IPND problem within time constraint. To this end, we propose a greedy framework with novel centralities to effectively solve the IPND problem in a timely manner. The various novel greedy functions are further empirically evaluated to be very effective to various interdependent networks.

Our main contributions are summarized as follows:

— Show the $(2-\epsilon)$-inapproximability of the IPND problem on interdependent networks.
— Provide the greedy framework with various centralities, solving IPND problem with competitive results.
— Validate the performance of our proposed algorithm on a wide range of interdependent networks with different scales, topologies, and interdependencies.

The rest of the paper is organized as follows. In Section II, we introduce the interdependent network model, a well-accepted cascading failure model and the formal definition of the IPND problem. Section III includes the hardness and inapproximability results. The greedy framework is proposed in Section IV, along with various centrality metrics. The experimental evaluation is illustrated in Section V. Finally, Section VI provides some concluding remarks.

II. Network Models and Problem Definition

In this section, we first introduce our interdependent network model and the well-accepted model of cascading failure. Using these models, we study the *Interdependent Power Network Disruptor* problem, to minimize the size of largest connected component in the power network after cascading failures by selecting a certain number of target nodes.

A. Interdependent Network Model

Considering an interdependent system, we abstract it into two graphs, $G_s = (V_s, E_s)$ and $G_c = (V_c, E_c)$, and their interdependencies, E_{sc}. G_s and G_c represent the power network and communication network respectively. Each of them has a set of nodes V_s, V_c and a set of links E_s, E_c, which are referred to as *intra-links*. In addition, E_{sc} are *inter-links* coupling G_s and G_c, i.e., $E_{sc} = \{(u,v)|u \in V_s, v \in V_c\}$. A node $u \in V_s$ is functional if it is connected to the giant connected component of G_s and at least one of its interdependent nodes in G_c is in a working state. The whole interdependent system is referred to as $\Im(G_s, G_c, E_{sc})$.

B. Cascading Failures Model

In this paper, we use a well-accepted cascading failure model, which has been validated and applied in many previous works [8], [10], [12], [15], [18]. Initially, there are a few critical nodes failed in network G_s, which disconnect a set of nodes from the largest connected component of G_s. Due to the interdependency of two networks, all nodes in G_c only connecting to failed nodes in G_s are also impacted, and therefore stop working. Furthermore, the failures cascade to nodes which are disconnected from the largest connected component in G_c and cause further failures back to G_s. The process continues back and forth between two networks until there is no more failure nodes.

C. Problem Definition

Definition 1 (IPND Problem): Given an integer k and an interdependent system $\Im(G_s, G_c, E_{sc})$, which consists of two networks $G_s = (V_s, E_s)$, $G_c = (V_c, E_c)$ along with their interdependencies E_{sc}. Let $LG_s(T)$ be the size of the largest connected component of G_s after the cascading failures caused by the initial removal of the set of nodes $T \subseteq V_s$ in G_s. The IPND problem asks for a set T of size at most k such that $LG_s(T)$ is minimized.

In the rest of paper, the pairs of terms interdependent, networks and coupled networks, node and vertex, as well as edge and link, are used interchangeably.

III. Computational Complexity

In this section, we first show the NP-completeness of IPND problem by reducing it from maximum independent set problem, which further implies that IPND problem is NP-hard to be approximated within the factor $2-\varepsilon$ for any $\varepsilon > 0$.

Theorem 1: IPND problem is NP-complete.

Proof: Consider the decision of IPND that asks whether the graph $G_s = (V_s, E_s)$ in an interdependent system $\Im(G_s, G_c, E_{sc})$ contains a set of vertices $T \subset V_s$ of size k such that the largest connected component in $G_s[V_s \setminus T]$ after cascading failures is at most z for a given positive integer z. Given $T \in V_s$, we can compute in polynomial time the size of the largest connected component in G_s after the cascade failures when removing T by iteratively identifying the largest

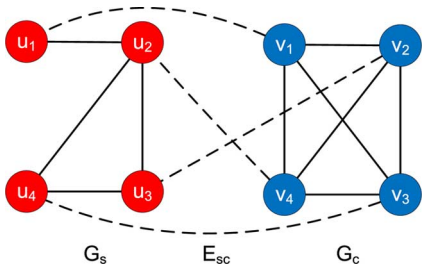


Fig. 2. An example of reduction from MIS to IPND.

connected component and removing disconnected vertices in G_s and G_c. This implies IPND $\in$ NP.

To prove that IPND is NP-hard, we reduce it from the decision version of the maximum independent set (MIS) problem, which asks for a subset $I \subseteq V$ with the maximum size such that no two vertices in I are adjacent. Let an undirected graph $G = (V, E)$ where $|V| = n$ and a positive integer $k \leq |V|$ be any instance of MIS. Now we construct the interdependent system $\mathfrak{I}(G_s, G_c, E_{sc})$ as follows. We select $G_s = G$ and G_c to be a clique of size $|V_s|$. Since G_s and G_c have the same size in our construction, to construct the interdependency between G_s and G_c, we randomly match each node in V_s to some arbitrary nodes in V_c so as to form a one-to-one correspondence between V_s and V_c. An example is illustrated in Fig. 2. We show that there is an MIS of size at most k in G iff G_s in $\mathfrak{I}(G_s, G_c, E_{sc})$ has an IPND of size $n - k$ such that the largest connected component in G_s after cascading failures is of size at most 1.

First, suppose $I \subseteq V$ is an MIS for G with $|I| \leq k$. By our construction, the largest connected component of $G_s[I]$ has the size 1 since there is no more cascading failure in the clique G_c. That is, $V_s \setminus I$ is also an IPND of $\mathfrak{I}(G_s, G_c, E_{sc})$.

Conversely, suppose that $T' \subseteq V_s$ with $|T'| = n - k$ is an IPND of $\mathfrak{I}(G_s, G_c, E_{sc})$, that is, the largest connected component of $G_s[V_s \setminus T']$ is of size 1. We show that $V_s \setminus T'$ is also an MIS of G. Since the failure of nodes in G_c will not cause any cascading failure in G_s, the largest connected component of $G_s[V_s \setminus T']$ is of size 1 iff $V_s \setminus T'$ is an independent set in G_s. That is, $V_s \setminus T'$ is also an MIS of G. ■

As IPND is NP-complete, one will question how tightly we can approximate the solution, leading to the theory of inapproximability. The inapproximability factor gives us the lower bound of near-optimal solution with theoretical performance guarantee. That said, no-one can design an approximation algorithm with a better ratio than the inapproximability factor. Then, we show that the above reduction implies the $(2-\varepsilon)$-inapproximability factor for IPND in the following corollary.

Corollary 1: IPND problem is NP-hard to be approximated into $2 - \varepsilon$ for any $\varepsilon > 0$.

Proof: We use the reduction in the proof of Theorem 1. Suppose that there is a $(2 - \varepsilon)$-approximation algorithm $\mathcal{A}$ for IPND. Then $\mathcal{A}$ can return the largest connected component in G_s of size less than 2 in our constructed instance if the optimal size is 1. Thus algorithm $\mathcal{A}$ can be applied to solve MIS on G in polynomial time because this size is integral. This contradicts to the NP-hardness of MIS. ■

IV. Greedy Framework for IPND Problem

In this part, we present different algorithms to detect the top critical nodes using the greedy framework, which has been illustrated as one of the most popular and effective approaches to solve hard problems. The idea is to iteratively choose the most critical node, whose removal degrades the functionality of the network as much as possible. In detail, we propose three following different strategies to select a critical node in the system at each iteration:

1) Select a node that maximizes the number of failed nodes after the cascading failure.
2) Select a node that decreases the structural strength of the system as much as possible. That is, when the number of removed nodes is large enough, the system will become weak. Therefore, the number of failed nodes increases considerably under the effect of cascading failures.
3) Select a node that not only increases the number of failed nodes but decreases the structural strength as well.

In the rest of this section, we describe three algorithms corresponding to the above strategies.

A. Maximum Cascading (Max-Cas) Algorithm

In maximum cascading (Max-Cas) algorithm, we iteratively select a node u that leads to the most number of new failed nodes, i.e., the maximum marginal gain to the current set of attacked nodes T. When a new node u fails, it results in a chain of cascading failures. The number of new failed nodes, referred to as *cascading impact number*, can be computed by simulating the cascading failures with the initial set $T \cup \{u\}$ on the interdependent system $\mathfrak{I}$ as described in Section II-B. However, the simulation of cascading failures is time-consuming due to its calculation of cascading failures between two networks. Each step in the cascading requires to identify the largest connected component of each network.

To this end, we further improve the running time of our algorithm by reducing the number of simulations. The idea is only to check potential nodes whose removal creates at least one more failed node in the same network due to the cascading failures. That is, this node (or its coupled node) disconnects the network which it belongs to, i.e., it (its coupled node) is an articulation node of G_s (or G_c), which is defined as any vertex whose removal increases the number of connected components in G_s (or G_c). The reason is illustrated in the following lemma.

Lemma 1: Given an interdependent system $\mathfrak{I}(G_s, G_c, E_{sc})$, removing a node $u \in V_s$ from the system causes at least one more node fail due to the cascading failure iff u (or its coupled node $v \in V_c$) is an articulation node in G_s (or G_c).

Proof: If u is an articulation node of G_s, the removal of u will increase the number of connected components in G_s at least to two. By the definition of cascading failures in G_s, all nodes disconnected from the largest connected component will be failed. Similarly, when v is an articulation node of G_c, removing u causes v fail, then there is at least one more node in G_c fail. After that, these nodes make coupled nodes in G_s fail as well. On the other hand, if neither u nor v are articulation nodes, the removal of u only makes v fail, and the rest of two networks are still connected, which terminates the cascading failures. ■

According to this property, the proposed algorithm first identifies all articulation nodes in both residual networks using

Hopcroft and Tarjan's algorithm [11]. Note that this algorithm runs in linear time on undirected graphs, which is faster than one simulation of cascading failures. Thus, the running time of each iteration is significantly improved especially when the number of articulation nodes is small. Denote $\text{Max-Cas}\,(G_s, T, \{u\})$ as the impact number of u, Algorithm 1 describes the details to detect critical nodes. In Algorithm 1 , since it takes $O(n)$ time to compute the cascading impact number for each node and at most $|A| < n$ articulation nodes will be evaluated, the running time is $O(kn^2)$ in the worst case. In practice, the actual running time is much less due to the small size of A, which is further illustrated in Section V.

Algorithm 1 Max-Cas Greedy Algorithm

Input: Interdependent system $\mathfrak{I}(G_s, G_c, E_{sc})$, an integer k
Output: Set of k critical nodes in $T \in V_s$
$T \leftarrow \emptyset$
for $i = 1$ to k **do**
  $A_s, A_c \leftarrow$ set of articulation nodes of G_s and G_c respectively
  $A \leftarrow \{u \in V_s | u \in A_s \vee ((u,v) \in E_{sc} \wedge v \in A_c)\}$
  if $A \neq \emptyset$ **then**
    $u \leftarrow \text{argmax}_{u \in A}\,\text{Max-Cas}(G_s, T, \{u\})$
    $T \leftarrow T \cup \{u\}$
  else
    $u \leftarrow$ any node in $V_s \setminus T$
  end if
  Update $\mathfrak{I}[V_s \setminus T]$
end for
Return T

B. Iterative Interdependent Centrality (IIC) Algorithm

As one can see, Max-Cas algorithm prefers to choose nodes that can decrease the size of networks immediately. This can mislead the algorithm to select boundary nodes and affect its efficiency for large k since the residual networks still remain highly connected even many critical nodes have been removed. An alternative strategy is to identify the hub nodes which plays a role to connect other nodes together in the network. Actually, this strategy has been proved to be efficient in single complex networks by Albert *et al.* [2], in which the removal of a small fraction of nodes with highest degree centrality has been shown to fragmentize the network to small components. However, since this centrality method is only for single networks, the development of a new centrality measure is in an urgent need for interdependent systems.

Intuitively, this new centrality measure is required to capture both the intra-centrality (the centrality of nodes in each networks) and inter-centrality (the centrality formed by the interconnections between two networks). Given an interdependent system $\mathfrak{I}(G_s, G_c, E_{sc})$, node u in V_s is more likely to be critical if its coupled node $v \in G_c$ is critical. Furthermore, when node u is considered as a critical node, its neighbors are also more likely to become important since the failures of these nodes can cause u fail. That said, the criticality of these nodes implies the criticality of their coupled nodes. To capture this complicated relation in interdependent systems, we develop an iterative method to compute the centralities of nodes, called *Iterative Interdependent Centrality* (IIC). Initially, the centralities of all nodes in G_s are computed by the traditional centrality, e.g., degree centrality, betweenness centrality, etc. After that, these centralities of nodes in G_s are reflected to coupled nodes in G_c and the centralities of nodes in G_c are updated based on the reflected values. The centralities of nodes in G_c continue to be reflected on nodes of G_s and update centralities of these nodes. Two key points of IIC are the *updating function* and the *convergence*.

1) Updating Function: Considering the reflected values from the other network as the weight of nodes, we modify the *weighted degree* as the updated centrality of nodes, which is defined as

$$\mathcal{C}(u) = \alpha w(u) + (1-\alpha) \sum_{v:(u,v)\in E} \frac{w(v)}{d_v}$$

where $w(\cdot)$ is the reflected values (or the weight of nodes) and the reservation factor α lying in the interval $[0, 1]$. The underlying reason we use weighted degree is that a node is usually more critical if most of its neighbors are critical nodes. The reservation factor shows that the importance of each node is not only dependent on the reflected values from the other network, but also the role in its own network.

2) Convergence: Next, we show that the centralities of nodes can be computed based on matrix multiplications and prove the convergence via this property. Let $\mathbf{x}^t = [x^t_{v^s_1}, x^t_{v^s_2}, \ldots, x^t_{v^s_n}]$, $\mathbf{y}^t = [y^t_{v^c_1}, y^t_{v^c_2}, \ldots, y^t_{v^c_n}]$ be the normalized centrality vector after t^{th} iteration of G_s and G_c. Suppose that two interdependent nodes have the same position vectors $\mathbf{x}^t$ and $\mathbf{y}^t$, i.e., v^s_i and v^c_i are interdependent. Then, we have

$$x^t_u = \alpha y^{t-1}_u + (1-\alpha) \sum_{v:(u,v)\in E_s} \frac{y^{t-1}_v}{d_v}, \quad \forall u \in V_s$$

$$y^t_u = \alpha x^{t-1}_u + (1-\alpha) \sum_{v:(u,v)\in E_c} \frac{x^{t-1}_v}{d_v}, \quad \forall u \in V_c.$$

Note that if we divide these vectors by a constant, then they still represent the centralities of nodes in the system. Thus, after each iteration, these centrality vectors are divided by some constants C_s and C_c which are chosen later to prove the convergence.

$$\mathbf{x}^t = \frac{\mathbf{x}^t}{C_s},\ \mathbf{y}^t = \frac{\mathbf{y}^t}{C_c}.$$

Let M^s and M^c be $n \times n$ matrices such that

$$M^s_{uv} = \begin{cases} \alpha & \text{if } u = v \\ d_v^{-1} & \text{if } (u,v) \in E_s \\ 0 & \text{otherwise,} \end{cases}$$

$$M^c_{uv} = \begin{cases} \alpha & \text{if } u = v \\ d_v^{-1} & \text{if } (u,v) \in E_c \\ 0 & \text{otherwise.} \end{cases}$$

Then the relationship between $\mathbf{x}^t$ and $\mathbf{y}^t$ is rewritten as:

$$\mathbf{x}^t = \frac{M^s \mathbf{y}^{t-1}}{C_s},\ \mathbf{y}^t = \frac{M^c \mathbf{x}^{t-1}}{C_c}.$$

Therefore

$$\mathbf{x}^t = \frac{M^s M^c \mathbf{x}^{t-2}}{C_s C_c} = \frac{M \mathbf{x}^{t-2}}{C_s C_c}$$

where $M = M^s M^c$ is called the characteristic matrix. Next, we analyze the condition of this matrix to guarantee that $\mathbf{x}^t$ con-

verges by using the Jordan canonical form of M, defined as follows.

Theorem 2 (Jordan Canonical Form [19]): Any $n \times n$ matrix M with n eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ can be represented as $M = PJP^{-1}$ where P is an invertible matrix and J is Jordan matrix which has form

$$J = \mathsf{diag}(J_1, \ldots, J_P)$$

where each block J_i, called Jordan block, is a square matrix of the form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}.$$

According to its above definition, the power of the matrix M can be computed as follows

$$M^k = (PJP^{-1})^k = PJ^kP^{-1}.$$

Hence, M^k converges when $k \to \infty$ if and only if J^k converges. The powers of J is computed via the powers of Jordan block $J_1^k, J_2^k, \ldots, J_p^k$.

$$J^k = \mathsf{diag}(J_1^k, \ldots, J_P^k)$$

where

$$J_i^k = \begin{bmatrix} \lambda_i^k & \binom{k}{1}\lambda_i^{k-1} & \binom{k}{2}\lambda_i^{k-2} & \cdots & \binom{k}{d_i-1}\lambda_i^{k-(d_i-1)} \\ & \lambda_i^k & \binom{k}{1}\lambda_i^{k-1} & \cdots & \binom{k}{d_i-2}\lambda_i^{k-(d_i-2)} \\ & & & \ddots & \\ & & & & \lambda_i^k \end{bmatrix}.$$

Note that the powers of J converges if and only if the powers of all Jordan blocks converge. Thus, we focus on the convergence of a block J^k as stated in the following lemma.

Lemma 2: The convergence of a $d \times d$ Jordan block J_i only depends on d and λ_i:

(1) If $|\lambda_i| > 1$ then J_i^k does not converge when $k \to \infty$.
(2) If $|\lambda_i| < 1$ then J_i^k converges to 0 when $k \to \infty$.
(3) If $|\lambda_i| = 1$ and $\lambda_i \neq 1$, then J_i^k does not converge when $k \to \infty$.
(4) If $\lambda_i = 1$ and $d = 1$, then $J_i^k = [1]$.
(5) If $\lambda_i = 1$ and $d > 1$, then J_i^k does not converge when $k \to \infty$.

Proof: Cases (1), (3), (4), and (5) are trivial, thus we only show the proof for case (ii). With $|\lambda_i| < 1$, every element of J_i^k has form $\binom{k}{j}\lambda_i^{k-j}$ which converge to 0 as $k \to \infty$. ∎

According to this lemma, when normalized factors C_s, C_c satisfies $C_sC_c = |\lambda_1|$, we will have

$$\left(\frac{M}{C_sC_c}\right)^{t/2}\mathbf{x}^0 = P\left(\frac{J}{|\lambda_1|}\right)^{t/2}P^{-1}\mathbf{x}^0.$$

Clearly, $\mathbf{x}^t$ will converge when $(J/|\lambda_1|)^{t/2}$ converges. Then, we have the following theorem.

Theorem 3: The centrality vector converges if and only if the characteristic matrix has exactly one maximum magnitude eigenvalue.

To compute the converged centrality vector, we first choose α such that M has $\lambda_1 > \lambda_2$. In practice, we choose $\alpha = 0.5$ and centrality vectors always converge. Although it seems necessary to compute the largest eigenvalue of M, we propose an alternative method to avoid this time-consuming computation as follows. Suppose that $\mathbf{x}^{2t}$ converges to a vector $\mathbf{x}$ after t_0 iterations i.e., $\mathbf{x} = M^{t_0}\mathbf{x}^0/|\lambda_1|^{t_0}$. Now we define the sequence of vectors $z_0 = \mathbf{x}^0$, $z_{i+1} = Mz_i/|Mz_i|$, then:

$$z_{t_0} = \frac{M^{t_0}z_0}{\prod_{i=0}^{t_0-1}|Mz_i|} = \frac{M^{t_0}\mathbf{x}^0}{\prod_{i=0}^{t_0-1}|Mz_i|}.$$

It means that $\mathbf{x} = Az_{t_0}$ where A is a scalar value. Therefore $z_{t_0} = \mathbf{x}/\|\mathbf{x}\|$. Thus we can compute the centrality vector using the recursive formula of z as described in Algorithm 2, then use this algorithm as sub-routine to detect critical nodes in Algorithm 3.

Algorithm 2 Iterative Interdependent Centrality

Input: Characterize matrix M and allowed error ϵ
Output: Centrality vector
$\mathbf{x} \leftarrow \mathbf{1}$
$error \leftarrow +\infty$
while $error > \epsilon$ **do**
  $\mathbf{y} \leftarrow M\mathbf{x}$
  $norm \leftarrow ||\mathbf{y}||$
  $\mathbf{y} \leftarrow \mathbf{y}/norm$
  $error \leftarrow ||\mathbf{y} - \mathbf{x}||$
  $\mathbf{x} \leftarrow \mathbf{y}$
end while
Return $\mathbf{x}$

Algorithm 3 IIC-based Algorithm

Input: Interdependent system $\mathfrak{I}(G_s, G_c, E_{sc})$, an integer k
Output: Set of k critical nodes in $T \in V_s$
$T \leftarrow \emptyset$
for $i = 1$ to k **do**
  $\alpha \leftarrow 0.5$, Compute M
  $\epsilon \leftarrow 10^{-8}$
  Compute centrality vector $\mathbf{x}$ using Algorithm 2.
  $u \leftarrow \mathrm{argmax}_{V_s \setminus T}\,\mathbf{x}[u]$
  $T \leftarrow T \cup \{u\}$
  Update $\mathfrak{I}[V_s \setminus T]$
end for
Return T

Time Complexity: Since two matrices M^s and M^c have only $(2|E_s| + |V_s|)$ and $(|E_c| + |V_c|)$ non-zero elements, the product $M\mathbf{x} = M^sM^c\mathbf{x}$ takes $O(2|E_s| + |V_s| + 2|E_c| + |V_c|)$ time using sparse matrix multiplication. The convergence speed is $|\lambda_1|/|\lambda_2|$, thus the number of iterations is $O(\log(1/\epsilon)/\log|\lambda_1|/|\lambda_2|)$. Therefore, the total running time to compute iterative interdependent centrality is $O((|E_s + E_c|)\log(1/\epsilon)/\log|\lambda_1|/|\lambda_2|)$. Thus, the total time to detect critical nodes is $O(k(|E_s + E_c|)\log(1/\epsilon)/\log|\lambda_1|/|\lambda_2|)$.

C. Hybrid Algorithm

Motivated by the advantages of Max-Cas and IIC algorithms, we further design a hybrid algorithm by taking advantage of them. As one can see, Max-Cas only works well when networks

are loosely connected since it mainly aims to create as many failed nodes as possible instead of making the network as weak as possible. On the other hand, IIC algorithm can make the network weak but it does not work well as Max-Cas when networks are loosely connected. Thus, the idea of hybrid algorithm is to remove as many nodes as possible and make networks weaker in turn. That is, we use Max-Cas and IIC algorithms in odd and even iterations respectively, as described in Algorithm 4 . Since the running time of IIC is much smaller than Max-Cas, its running time is about a half of Max-Cas, which will be empirically shown in Section V.

Algorithm 4 Hybrid Algorithm

Input: Interdependent system $\Im(G_s, G_c, E_{sc})$, an integer k
Output: Set of k critical nodes in $T \in V_s$
$T \leftarrow \emptyset$
for $i = 1$ to k **do**
  if i is odd **then**
    Select u as Max-Cas algorithm
  else
    Select u as IIC algorithm
  end if
  $T \leftarrow T \cup \{u\}$
  Update $\Im[V_s \setminus T]$
end for
Return T

V. Experimental Evaluation

A. Dataset and Metric

In the experiment, we evaluate Max-Cas, IIC, and Hybrid algorithms, with respect to the size of giant connected component (GCC) and the running time, on various real-world and synthetic datasets.

In terms of power networks, we use both real Western States power network of the US [20] with 4941 nodes and 6594 edges, and the synthetic scale free networks. This network as well as other communication networks belong to a class of networks called scale-free networks in which the number of nodes with degree d, denoted by $\mathsf{P}(d)$, is proportional to $d^{-\beta}$ i.e., $\mathsf{P}(d) \sim d^{-\beta}$ for some exponential factor $\beta > 0$. According to [4], power networks are found to have their exponential factors β between 2.5 and 4. In order to do a more comprehensive experiment, we further generate more types of synthetic power networks with different exponential factors, using igraph package [9].

In addition, due to the lack of data describing interdependencies between any communication networks and the real-world power network, we use the synthetic scale-free networks, representing communication networks, e.g., Internet, telephone network, etc. Since most communication networks are observed to have the scale free property with their exponential factors β between 2 and 2.6 [6], [21], we generate communication networks with component factors of 2.2 or 2.6.

For the sake of coupling method, motivated by the observation from real-world interdependent systems in [16], we develop a realistic and practical coupling approach, Random Positive Degree Correlation Coupling (RPDCC) scheme. In this scheme, nodes with high degrees tend to coupled together and so do nodes with low degrees, thus the degree correlation of coupled nodes is positive as described in [16]. The detail of RPDCC strategy will be discussed in the Appendix.

Finally, each experiment on synthesized systems is repeated 100 times to compute the average results.

B. Performance of Proposed Algorithms

In order to show the effectiveness of our proposed algorithms, due to the intractability of IPND problem and the time consumption to obtain optimal solutions, we focus on comparing them with traditional centrality approaches which are often used in network analysis [5], including degree centrality (DC), closeness centrality (CC), betweenness centrality (BC) [7], and bridgeness centrality (BRC) [13]. In these approaches, the k nodes of largest centralities in power networks are selected as critical nodes. Particularly, we test our approaches on the following three types of datasets:

1) WS System: US Western states power network — Scale-free communication network with $\beta = 2.2$.
2) SS System: Scale-free power networks with $\beta = 3.0$— Scale-free communication network with $\beta = 2.2$.
3) Eq-SS System: Scale-free power and communication networks with the same $\beta = 2.6$.

Here we choose the exponential factor β according to the real-world power networks and communication networks, as mentioned above in V-A .

Fig. 3 reports the comparison of performance between different approaches in these three interdependent systems. In these figures, all of three proposed algorithms outperform CC (the best traditional centrality approach) for any number of k critical nodes. When k becomes larger, the interdependent systems have totally destroyed by choosing these critical nodes using our algorithms, while more than 60% of nodes remain intact if selecting nodes with highest traditional centralities. Especially in WS interdependent system consisting of real-world US Western states power system, the number of functional nodes remains nearly 5000 even 50 nodes are identified using CC, whereas it is sufficient to destroy the whole system only by removing 20 nodes using our Hybrid or Max-Cas approach. That is, these traditional approaches perform much worse compared with our algorithms, especially when the number of attacked nodes is large. Thus, the traditional methods cannot identify a correct set of critical nodes in interdependent systems. The reason is that these approaches can only reflect the importance of each node in single power networks rather than interdependent systems, and they ignore the impact of cascading failures to interdependent systems.

Comparing our three proposed approaches, as revealed in Fig. 3, IIC runs fastest in spite of its worst performance, roughly 1000 times faster than Max-Cas in WS interdependent system. We also notice that the performance of Max-Cas and Hybrid algorithms is very close while Hybrid algorithm runs about 2 times faster than Max-Cas algorithm. In particular, Max-Cas has better performance than Hybrid algorithm in SS interdependent system, yet worse performance in the other two systems. This is because the power network with $\beta = 3.0$ is very loosely connected and fragile in SS interdependent systems. Thus, Max-Cas strategy can destroy the system quickly and easily. However, since nodes are better connected in the other two systems, especially Eq-SS, Hybrid algorithm is more efficient due to its strategy that makes networks weak first and then destroys them.

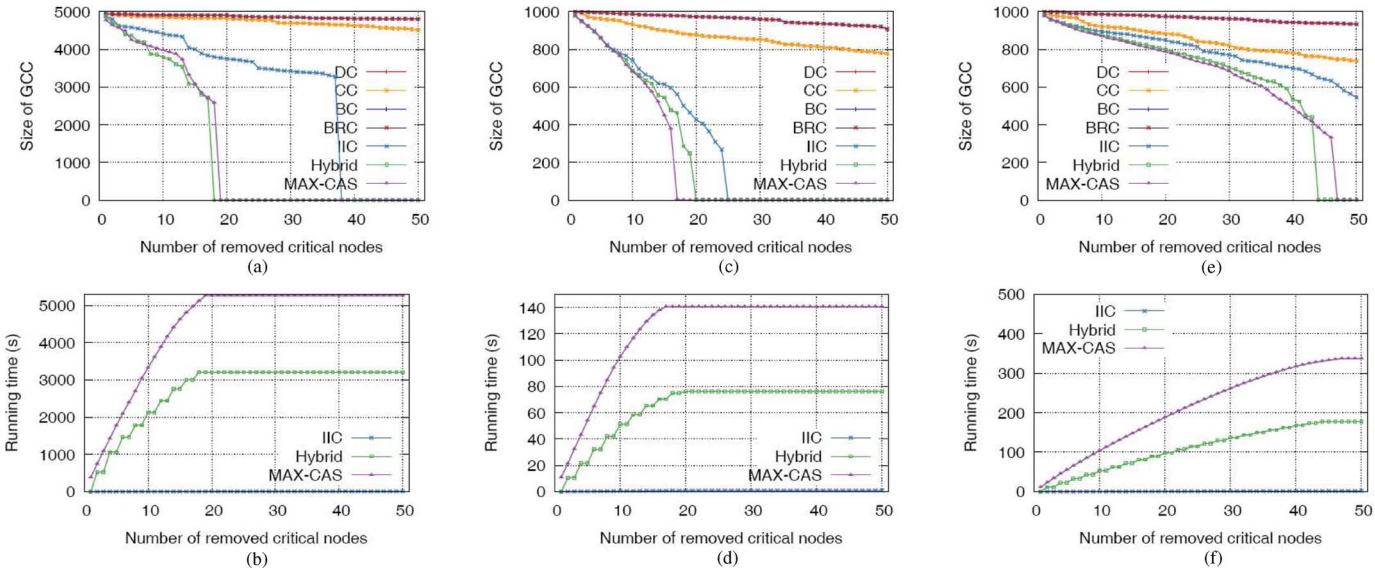


Fig. 3. Performance Comparison on Different Interdependent Systems: WS System (a, b), SS System (c, d), and Eq-SS System (e, f).

As illustrated in Fig. 3(c), the performance of Hybrid is lower than Max-Cas initially, but higher than Max-Cas when the networks get weak enough. Additionally, in all of these systems, when the number of removed nodes reach to a certain value, the whole system is failed. These numbers are about 20 for WS and SS system and 40 for Eq-SS system. This shows that interdependent networks are vulnerable, especially when loosely connected.

C. *Vulnerability Assessment of Interdependent Systems*

With the effectiveness of Hybrid algorithm observed through the above experiments, we confidently use it to further assess the vulnerability of interdependent systems and explore some insight properties.

1) Different Coupled Communication Networks: We are interested in investigating the vulnerability of a certain power network when it is coupled with different communication networks. First, we fix one synthetic power network by generating a scale-free network with $\beta = 3$ according to [4]. The coupled communication networks are also generated as scale-free networks, with their exponential factors β between 2.5 and 2.7, as mentioned above. All generated networks have 1000 nodes.

As illustrated in Fig. 4, the power networks tend to be more vulnerable when their coupled communication networks are more sparse, i.e., with larger exponential factor β. That is, it gives us an intuition that the power networks will become more vulnerable when their coupled networks are easy to be attacked. In particular, in order to destroy the power networks, the numbers of critical nodes in them are 23, 17, and 11 when their coupled communication networks have $\beta = 2.5$, $\beta = 2.6$ and $\beta = 2.7$, respectively, which indicates some key thresholds to protect the function of power networks with the knowledge of their interdependent networks.

2) Disruptor Threshold: In this part, we evaluate an important indicator of the vulnerability, the disruptor threshold which is the number of nodes whose removal totally destroys the whole system. The smaller it is, the more vulnerable the system is. We would like to observe the dependence of the disruptor threshold

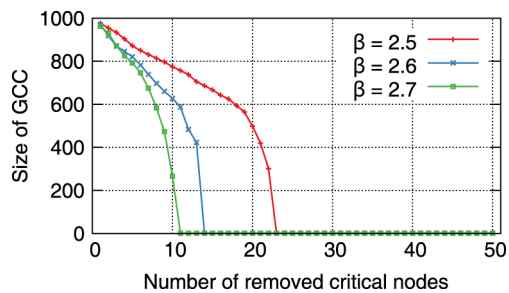


Fig. 4. The Vulnerability Of A Fixed Power Network.

on the network size. Particularly, we generate two scale-free networks with the same size and exponential factors β of 3.0 and 2.2, corresponding to power and communication networks, then couple them using RPDCC scheme.

As shown in Fig. 5, the disruptor threshold provided by all proposed algorithms is small and increases slowly with respect to the growth of the network size. When the network size is raised by 5 times, from 1000 to 5000 nodes, the disruptor threshold only increases roughly 3 times. When the size of network is 5000, the disruptor thresholds of Max-Cas and Hybrid algorithms are roughly 51 and 57. This implies that the removal of 1% number of nodes is enough to destroy the whole system. Even the IIC algorithm needs to destroy only 1.5% fraction of nodes to break the system down. Large interdependent systems seem to be extremely vulnerable under different attack strategies due to the following reason. When the network size grows up, the possibility that a high degree node is dependent on a low degree node also runs up. As a result, it is easier to disable the functionality of high degree nodes which often play an important role in the network connectivity. Therefore, the vulnerability of the interdependent system needs to be reevaluated regularly, especially fast growing up systems.

3) Different Coupling Schemes: Another interesting observation is to investigate the impacts of the way nodes are coupled to the vulnerability of interdependent system. Apart from

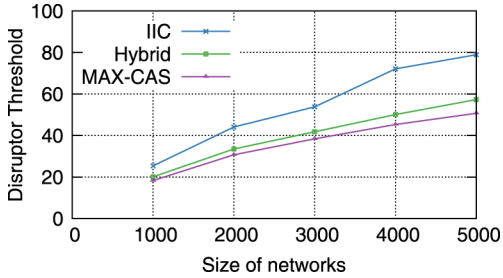


Fig. 5. The Disruptor Threshold with Different Network Sizes.

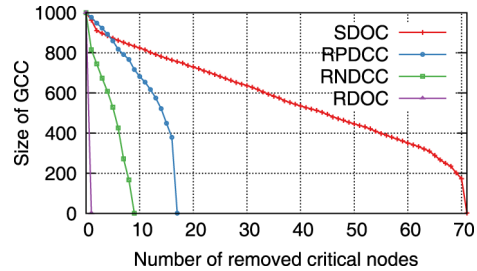


Fig. 6. Vulnerability Comparison using Different Coupling Schemes.

the RPDCC scheme, we evaluate the robustness with other three coupling strategies, as follows:

1) *Same Degree Order Coupling (SDOC)*: The nodes of i^{th} highest degree in two networks are coupled together.
2) *Reversed Degree Order Coupling (RDOC)*: The node of i^{th} highest degree in one network is coupled with the node of i^{th} lowest degree in the other network.
3) *Random Negative Degree Correlation Coupling (RNDCC)*: A node of higher degree in one network are randomly, with lower probability, to couple with another node of higher degree nodes in the other network.

Note that the RNDCC scheme is the opposite strategy to the RPDCC scheme (in the Appendix). We test on the interdependent systems, consisting of a power network with $\beta = 3$ and a communication network with $\beta = 2.2$ using the four different coupling schemes. All networks have 1000 nodes.

Fig. 6 reports the vulnerability of power networks when coupling them with communication networks in different manners. As one can see, SDOC provides the most robust interdependent system, although it is not practical. The size of the remained giant connected component decreases slowly when the number of removed nodes increases. On the other hand, RDOC makes the system very vulnerable, which can be destroyed by only removing one node from the power network. This is because the nodes of lower degree in communication networks are very easy to be failed, which, immediately, cause the failures to their coupled nodes of higher degree in power networks. When many high degree nodes are removed, the network is easy to be fragmented which leads to the destruction of the whole system shortly. The interdependent systems with the other two schemes, RPDCC and RNDCC, illustrate their robustness between those using SDOC and RDOC, due to the random factors in RPDCC and RNDCC. Compared with RNDCC, systems coupled by RPDCC are almost twice more robust because of their positive correlations. These results point out an important principle that the higher correlation between the degrees of coupled nodes, the stronger the interdependent system is. In other words, a node of high degree in one network should not be coupled with a node of low degree in the other network; otherwise, this node will be a weak point to attack.

VI. Conclusion

In this paper, we studied the optimization problem of detecting critical nodes to assess the vulnerability of interdependent power networks based on the well-accepted cascading failure model and metric, the size of largest connected component. We showed its NP-hardness, along with its inapproximability. Due to its intractability, we proposed a greedy framework with various novel centralities, which measures the importance of each node more accurately on interdependent networks. The extensive experiment not only illustrates the effectiveness of our approaches in networks with different topologies and interdependencies, but also reveals several important observations on interdependent power networks.

Appendix
RPDCC/RNDCC Coupling Scheme

In the appendix, we present the RPDCC scheme to randomly couple two networks with positive degree correlation. Given two network G_s and G_c, we form two weighted sets that contain vertices of G_s and G_c as elements and their degrees as weights. Then we generate two random weighted permutations $\{v_1^{s'}, v_2^{s'}, \ldots, v_n^{s'}\}$ and $\{v_1^{c'}, v_2^{c'}, \ldots, v_n^{c'}\}$ of nodes in G_s and G_c as described in in Algorithm 5, then $v_i^{s'}$ is coupled with $v_i^{c'}$, $1 \leq i \leq n$. In the following theorem, we show that a node of larger weight has smaller expected index in each permutation, that is, nodes of high degrees in two permutations tend to have low indices. In other words, this results in the positive correlation between degrees of coupled nodes. (For RNDCC, we couple $v_i^{s'}$ with $v_{n-i}^{c'}$.)

Algorithm 5 Random weighted permutation

Input: A weighted set of n elements $X = \{x_1, x_2, \ldots, x_n\}$ with weights $w(\cdot)$
Output: Weighted random permutation Y of X.
$total \leftarrow \sum_{i=1}^{n} w(x_i)$
for $i = 1$ to n **do**
  $e \leftarrow$ a random selected element in X with probability $w(e)/total$
  $Y[i] \leftarrow e$; $X \leftarrow X \setminus \{e\}$; $total \leftarrow total - w(e)$
end for
Return Y

Theorem 4: In the random weighted permutation, an element with larger weight has lower expected index than an element with smaller weight.

Sketch of Proof: Let $\mathsf{E}(X, e)$ be the expected index of an element e in the random weighted permutation. Then, we have:

$$\mathsf{E}(X, e) = \frac{w(e)}{\sum\limits_{x \in X} w(x)} + \sum_{z \in X \setminus \{e\}} \frac{w(z)}{\sum\limits_{x \in X} w(x)} (1 + \mathsf{E}(X \setminus \{z\}, e).$$

Therefore, $\mathsf{E}(X, e_1) \leq \mathsf{E}(X, e_2)$ if $w(e_1) \geq w(e_2)$. ■

REFERENCES

[1] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the north american power grid," *Phys. Rev. E*, vol. 69, no. 2, Feb. 2004.

[2] R. Albert, H. Jeong, and A.-L. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[3] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Comput. Oper. Res.*, vol. 36, pp. 2193–2200, Jul. 2009.

[4] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[5] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466–484, 2006.

[6] , S. Bornholdt and H. G. Schuster, Eds., *Handbook of Graphs and Networks: From the Genome to the Internet*. New York, NY: Wiley, 2003.

[7] U. Brandes and T. Erlebach, *Network Analysis: Methodological Foundations*. New York: Springer, 2005.

[8] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[9] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal, Complex Syst.*, p. 1695, 2006.

[10] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Networks formed from interdependent networks," *Nature Phys.*, vol. 8, no. 1, pp. 40–48, 2011.

[11] J. Hopcroft and R. Tarjan, "Algorithm 447: Efficient algorithms for graph manipulation," *Commun. ACM*, vol. 16, no. 6, pp. 372–378, Jun. 1973.

[12] X. Huang, J. Gao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, "Robustness of interdependent networks under targeted attack," *Phys. Rev. E*, vol. 83, p. 065101, Jun. 2011.

[13] W. Hwang, Y.-R. Cho, and M. Ramanathan, Bridging Centrality: Identifying Bridging Nodes in Scale-free Networks Dept., CSE, Univ. Buffalo, 2006.

[14] F. Luciano, F. Rodrigues, G. Travieso, and V. P. R. Boas, "Characterization of complex networks: A survey of measurements," *Adv. Phys.*, vol. 56, no. 1, pp. 167–242, 2007.

[15] R. Parshani, S. V. Buldyrev, and S. Havlin, "Interdependent networks: Reducing the coupling strength leads to a change from a first to second order percolation transition," *Phys. Rev. Lett.*, vol. 105, no. 4, p. 048701, 2010.

[16] R. Parshani, C. Rozenblat, D. Ietri, C. Ducruet, and S. Havlin, "Inter-similarity between coupled networks," *EPL (Europhysics Lett.)*, vol. 92, no. 6, p. 68002, 2010.

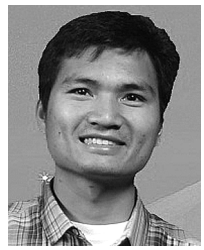
[17] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. D. Porcellinis, and R. Setola, "Modelling interdependent infrastructures using interacting dynamical models," *Int. J. Critical Infrastructures*, vol. 4, no. 1/2, p. 63, 2008.

[18] C. M. Schneider, N. A. M. Araujo, S. Havlin, and H. J. Herrmann, "Towards designing robust coupled networks," *Minerva*, pp. 1–7, 2011.

[19] G. Strang, *Linear Algebra and its Applications*, 3rd ed. Independence, KY: Brooks Cole, 1988, p. 505.

[20] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[21] Y. Xia, C. K. Tse, W. M. Tam, F. C. M. Lau, and M. Small, "Scale-free user-network approach to telephone network traffic analysis," *Phys. Rev. E*, vol. 72, p. 026116, Aug. 2005.



Dung T. Nguyen received the B.S. degree in information technology from Hanoi University of Science and Technology, Hanoi, Vietnam (2008). Currently he is pursuing the Ph.D. degree in the Department of Computer and Information Science and Engineering, University of Florida, Gainesville.

His areas of interest are viral marketing on online social networks, vulnerability and cascading failures on coupled networks, and approximation algorithms for network optimization problems.

Yilin Shen (S'12) received the M.S. degree in computer engineering from the University of Florida, Gainesville, in 2012, where he is currently pursing the Ph.D. degree at the Department of Computer and Information Science and Engineering, under the supervision of Dr. My T. Thai.

His research interests include the vulnerability assessment and security of complex networks, and the design of approximation algorithms for network optimization problems.

My T. Thai (M'06) received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, in 2005.

She is an Associate Professor at the Computer and Information Science and Engineering Department, University of Florida. Her current research interests include algorithms and optimization on network science and engineering.

Dr. Thai has engaged in many professional activities, serving many conferences, such as being the PC chair of IEEE IWCMC 12, IEEE ISSPIT 12, and COCOON 2010. She is an Associate Editor of JOCO, IEEE TPDS, and a series editor of Springer Briefs in Optimization. She has received many research awards including a Provost's Excellence Award for Assistant Professors at the University of Florida, a DoD YIP, and an NSF CAREER Award.