International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 9, Number 2, pages 201–216, July 2019

# Critical Node Identification based on Articulation Point Detection for Uncertain Network

Kouzou Ohara
College of Science and Engineering, Aoyama Gakuin University
Sagamihara, Kanagawa, 252-5258, Japan

Kazumi Saito
Faculty of Science, Kanagawa University
Hiratsuka, Kanagawa, 259-1923, Japan
Center for Advanced Intelligence Project, RIKEN
Tokyo, 103-0027, Japan

Masahiro Kimura
Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga, 520-2194, Japan

Hiroshi Motoda
Institute of Scientific and Industrial Research, Osaka University
Ibaraki, Osaka, 567-0047, Japan

Received: February 15, 2019
Revised: May 6, 2019
Accepted: May 29, 2019
Communicated by Hiroyuki Sato

**Abstract**

The problem of efficiently identifying critical nodes that substantially degrade network performance if they do not function is crucial and essential in analyzing a large complex network such as social networks on the Web and road network in the real world, and it is still challenging. In this paper, we tackle this problem under a realistic situation where each link is probabilistically disconnected as assumed in studies in uncertain graphs. This reflects that in case of a social network an information path between two persons is not always open and may not pass on any information from one to the other and in case of a road network a road between two intersections is not always travelable and may be blocked by a traffic accident, a road repair, a nearby construction, etc. To solve this problem, we focus on the articulation point and utilize the bridge detection technique in graph theory to efficiently identify critical nodes when the node reachability is taken as the performance measure. In case of a social network disfunction of a node causes loss of the total number of people receiving information and in case of a road network it causes loss of the total number of people movable to other places. Using two real-world social networks and one road network, we empirically show that the proposed method has a good scalability with respect to the network size and the nodes our method identified possesses unique properties and they are difficult to be identified by using conventional centrality measures.

*Keywords:* critical node, articulation point detection, uncertain graphs, network analysis

[0]This paper is an extended version of our conference paper for CANDAR 2018 [13].

# 1 Introduction

Identifying critical nodes and links that play an important role in a large complex network is one of the essential and crucial issues in various fields including communication network analysis, urban design, etc. For instance, in the field of social network mining [2, 7], detecting a limited number of influential nodes that are effective for widely spreading information is known as the influence maximization problem, which is motivated by viral marketing. Conventional centrality metrics such as degree centrality and betweenness centrality are often used to quantify the importance of individual nodes. However, since these metrics are based only on network topology, other factors that are more realistic, such as probabilistic information diffusion model and geodesic distance have also been used to assess a certain network performance metric [11, 12, 14, 17–19, 21]. The objective of these studies is identifying the most critical nodes/links in maintaining or maximizing a desired network performance, that is, identifying those nodes/links that degrade the performance substantially if they fail to function. For example, suppose that the desired performance is maintaining an ability of information spread of a given social network. Then, the critical nodes to be identified are those which maximally reduce the total number of people who can receive information issued by other persons when those nodes do not distribute information they received. This problem can be mathematically formulated as an optimization problem once the performance measure is quantitatively defined when a network structure is given.

In reality, every link in a given network is not necessarily always available. In a social network, a user may not always pass on information she received to her friends. In a road network, some roads may be damaged and blocked in a natural disaster. Thus, in this paper, considering the probability that each link is disconnected, we introduce a probabilistic link disconnection model into our problem setting as another kind of realistic factor. This is a standard approach taken in most studies on uncertain graphs. In other words, in this paper, we consider a problem of identifying critical nodes in a given uncertain network. Under this problem setting, we adopt, as the performance measure, the contribution value which is defined as the reduction of expected total number of nodes that are reachable from each node in a given network when a specific node stops functioning. For example, assuming a social network, this measure quantifies how a person affects the overall communication over the network if she does not distribute information she received, and assuming a road network, this measure quantifies how a particular intersection of roads affects overall reachability over the network if it stops allowing people to pass. We refer to the contribution value as the *latent* criticalness centrality because the contribution value does not make sense unless the node fail to function.

To solve the problem of ranking the nodes according to the contribution values and identifying the most critical nodes, we focus on the articulation point in graph theory, and propose an algorithm to efficiently compute the contribution value of each node in a given network by incorporating the standard bridge detection algorithm that helps us find the articulation points. An articulation point is a node in a connected component such that removing the node and links incident to it divides the component into two or more disjoint ones. Obviously, dividing a single component into more than one disjoint one reduces the number of nodes reachable from each node in each component. Thus, the articulation point has a positive contribution value and can be a candidate of critical nodes. A bridge is a link in a connected component such that removing it divides the component into two disjoint ones. Therefore, one of the end points of a bridge can be an articulation point unless it becomes an isolated node after removing the bridge. Thus, we identify all bridges in a network to find all articulation points and compute their contribution values based on the standard bridge detection algorithm whose computational complexity is $O(|\mathcal{E}|)$, where $\mathcal{E}$ is a set of links in the network. To the best of our knowledge, our approach is the first to identify critical nodes based on the bridge detection algorithm under a probabilistic link disconnection model.

Below we summarize our contributions in this paper. 1) We present a realistic problem of detecting critical nodes under a probabilistic link disconnection model. 2) We propose an efficient method to compute the contribution values of each node based on bridge detection, whose computational complexity is $O(H \times |\mathcal{E}|)$, where $H$ is the number of networks generated from an original one based on the probabilistic link disconnection model. 3) We experimentally demonstrate the effectiveness of our proposed method for two real-world social networks and one real-world road network. The results show that our method is much faster than computing the traditional centrality measures, and the nodes detected by our method are substantially more critical than those by the other methods in terms of the contribution values for the same setting we used for the probabilistic link disconnection model.

The paper is organized as follows. Section 2 briefly explains the related work of this paper. Section 3

formulates the critical node detection problem based on a probabilistic link disconnection model. Section 4 presents the proposed method based on bridge detection. Section 5 shows experimental results on three real-world networks. Section 6 summarizes the main achievements and future directions.

## 2 Related Work

Conventional node-centrality measures such as the degree centrality and the betweenness centrality [4] have been used to quantify the importance (criticalness in view of this paper) of each node in a network. These centrality measures are based only on network topology and have been used to analyze spatial networks embedded in the real world from structural viewpoints [3,11,12,15]. Some studies have used a more problem specific performance measure to quantify the criticalness of nodes. For example, we proposed a performance measure based on the node reachability [17, 19], similarly to the one we adopt in this paper. This work is different from these in that in this paper we focus on the critical node detection problem in a situation where links are probabilistically blocked.

The problem setting in this paper is closely related to the information maximization problem in social network analysis [2, 7], as mentioned previously. It aims at detecting a limited number of influential nodes that are effective for widely spreading information. But, the influential nodes are not always critical to maintain the network performance such as the node reachability we consider in this work because alternative information paths that do not pass through those nodes might exist. Rather, more closely related to this work is the contamination minimization problem [9] in which a small number of links are blocked to minimize the spread of contamination over a network. Indeed, we adopt the same idea in [9] and sample graphs from a given original network by deciding connectivity of each link according to the disconnection probability to estimate the expected number of nodes reachable from each node in the network. However, unlike the work [9], we compute the difference of reachability sizes before and after a node is removed directly, given a sampled graph. The method in [9] requires much larger number of samples for each link as it computes the expected value of both when it is connected and disconnected separately.

Our work can be regarded as a method for identifying critical nodes over uncertain graphs [6, 8] with independent disconnection probabilities for each link. Several techniques have been proposed for analyzing uncertain graphs for tasks including $k$-nearest neighbors [16] and clustering [5]. However, unlike these studies on uncertain graphs, our proposed method is based on the bridge detection algorithm [22] in graph theory. The bridge itself is critical in a network since removing it breaks the connectivity of the network. Thus, bridge detection is embedded into various problems such as critical link detection in wireless sensor networks [1] and improvement of computational efficiency of conventional centrality measures [20]. However, these studies on bridge detection assume that the network structure is stable. To the best of our knowledge, there is no work that uses the bridge detection technique to identify critical nodes over uncertain graphs produced by a probabilistic link disconnection model.

## 3 Problem Formulation

Let $G = (\mathcal{V}, \mathcal{E})$ be a given simple undirected graph without self-loops, where $\mathcal{V} = \{u, v, w, \cdots\}$ and $\mathcal{E} = \{e, \cdots\}$ are sets of nodes and undirected links, respectively. We also express each link $e$ as a pair of nodes, *i.e.*, $e = \{u, v\}$. Let $\mathcal{R}(v; G)$ be the set of reachable nodes by following links from a node $v$ over $G$, where note that $v \in \mathcal{R}(v; G)$. For each link $e \in \mathcal{E}$, let $x_e$ be a random variable expressing the link connectivity, *i.e.*, $x_e = 1$ if the link $e$ is disconnected; otherwise $x_e = 0$, where we denote its disconnection probability by $p(x_e = 1) = p_e$. As performed in studies on uncertain graphs, we also introduce such probabilities that each link is accidentally missed and disfunctions in order to reflect the realistic situation and evaluate the robustness of networks. By using a set of random variables defined by $\mathcal{X} = \{x_e \mid e \in \mathcal{E}\}$, we can define a graph $G_{\mathcal{X}} = (\mathcal{V}, \mathcal{E}_{\mathcal{X}})$, where $\mathcal{E}_{\mathcal{X}} = \{e \in \mathcal{E} \mid x_e = 0\}$. Now, by assuming independent Bernoulli trials for all the links, we can compute the occurrence probability of each graph $G_{\mathcal{X}}$ by

$$p(G_{\mathcal{X}}) = \prod_{x_e \in \mathcal{X}} p_e^{x_e}(1 - p_e)^{1 - x_e} = \prod_{e \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{X}}} p_e \prod_{e \in \mathcal{E}_{\mathcal{X}}} (1 - p_e). \tag{1}$$

where $\cdot \setminus \cdot$ stands for a set difference operator.

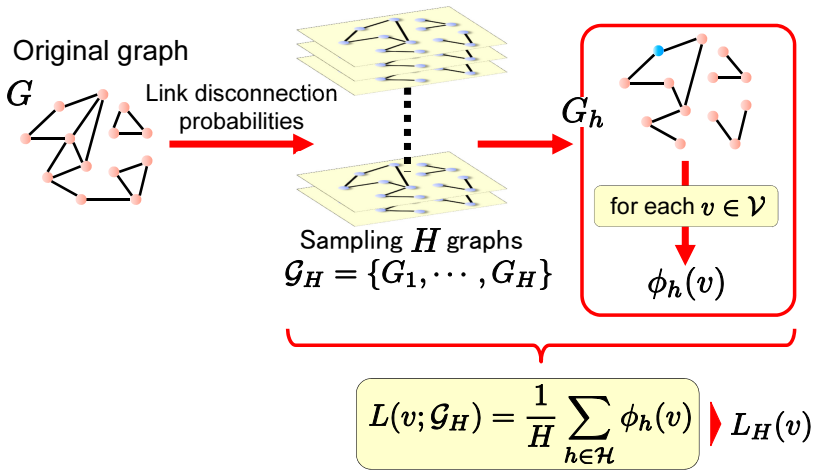



Figure 1: Intuitive illustration of the problem setting of computing the expectation of the degradation value of each node in an uncertain network.

For each graph $G_X$, let $G_X(v) = (\mathcal{V} \setminus \{v\}, \mathcal{E}_X(v))$ be a graph constructed by removing a node $v \in \mathcal{V}$ and its incident links, where $\mathcal{E}_X(v) = \{e \in \mathcal{E}_X \mid e \cap \{v\} = \emptyset\}$. Here, assume that a graph $G_X$ is decomposed into $K$ connected components, and let $\mathcal{V}_X^{(k)}$ be the set of nodes belonging to the $k$-th connected component. Then, we define the total reachability value of $G_X$ for all nodes by the following formula:

$$\sum_{w\in\mathcal{V}} |\mathcal{R}(w;G_X)| = \sum_{k=1}^{K} |\mathcal{V}_X^{(k)}|^2.$$

Thus, in case that $v \in \mathcal{V}_X^{(k)}$, *i.e.*, $\mathcal{V}_X^{(k)} = \mathcal{R}(v;G_X)$, by focusing only on the $k$-th connected component, we define the expected reachability degradation value of $G_X(v)$ from $G_X$. Here, we note the following inequality.

$$|\mathcal{R}(v;G_X)| - 1 \leq \sum_{w\in\mathcal{R}(v;G_X)\setminus\{v\}} |\mathcal{R}(w;G_X(v))| \leq (|\mathcal{R}(v;G_X)| - 1)^2.$$

In this paper, we propose to define and employ the following reachability degradation value.

$$\phi_X(v) = (|\mathcal{R}(v;G_X)| - 1)^2 - \sum_{w\in\mathcal{R}(v;G_X)\setminus\{v\}} |\mathcal{R}(w;G_X(v))|. \tag{2}$$

Evidently, in order to obtain the expected reachability degradation value, we need to compute $\phi(v) = \langle\phi_X(v)\rangle_X$, where $\langle\cdot\rangle_X$ means an expected value taken for all the possible assignments to random variables. However, it is difficult to exactly compute $\phi(v)$ due to a large number of possible network configurations, which amounts to $2^{|\mathcal{E}|}$. Thus, we employ an approach based on Monte Carlo simulation. Let $\mathcal{H}$ be a set of integers defined by $\mathcal{H} = \{1, \cdots, H\}$. Now, we repeat simulations $H$ times based on the probabilistic model defined in Eq. (1), and sample a set $\mathcal{G}_H$ of $H$ graphs *i.e.*, $\mathcal{G}_H = \{G_h = (\mathcal{V}, \mathcal{E}_h) \mid h \in \mathcal{H}\}$, where $\mathcal{E}_h \subset \mathcal{E}$ is the set of non-disconnected links at the $h$-th simulation. Then, by defining $\phi_h(v)$ as follows:

$$\phi_h(v) = (|\mathcal{R}(v;G_h)| - 1)^2 - \sum_{w\in\mathcal{R}(v;G_h)\setminus\{v\}} |\mathcal{R}(w;G_h(v))|, \tag{3}$$

we can define the reachability degradation value $L(v;\mathcal{G}_H)$ of a node $v \in \mathcal{V}$ for $\mathcal{G}_H$ as follows:

$$L(v;\mathcal{G}_H) = \frac{1}{H}\sum_{h\in\mathcal{H}} \phi_h(v). \tag{4}$$

Hereafter, we denote $L(v;\mathcal{G}_H)$ simply as $L_H(v)$ because the set of generated graphs $\mathcal{G}_H$ is fixed in our experiments. Figure 1 illustrates the overall process to compute $L_H(v)$. Evidently, $L_H(v)$ is an unbiased estimator,

*i.e.*,

$$\langle L_H(v)\rangle_{\mathcal{H}} = \frac{1}{H}\sum_{h\in\mathcal{H}}\langle \phi_{\mathcal{X}}(v)\rangle_{\mathcal{X}} = \phi(v). \tag{5}$$

Thus, when $H$ is a sufficiently large number, $L_H(v)$ can be a close approximation to the expected reachability degradation value $\phi(v)$. In this paper, we focus on the problem of accurately and efficiently calculating $L_H(v)$ for every $v \in \mathcal{V}$.

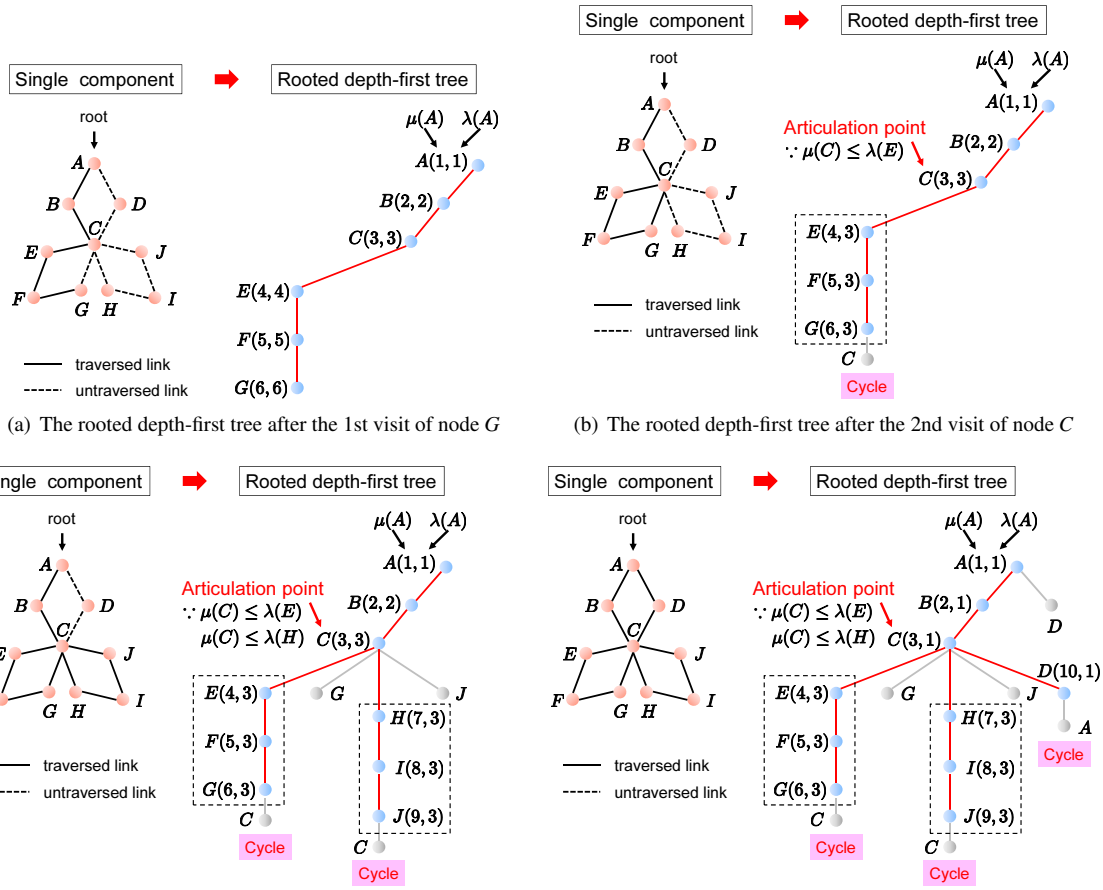
# 4 Proposed Method

To develop an effective algorithm for computing degradation value $L_H(v)$ for every $v \in \mathcal{V}$, we focus on the following condition. For a generated graph $G_h = (\mathcal{V}, \mathcal{E}_h)$, each node $v \in \mathcal{V}_h$ has a positive degradation value, *i.e.*, $\phi_h(v) > 0$, if and only if $v$ is an articulation point in a connected component in $G_h$. Here, a vertex is said to be an articulation point (or cut vertex) if and only if removing it (and edges through it) disconnects the graph. In order to compute all articulation points of $G_h$, we employ the idea of standard bridge detection algorithm originally proposed by Tarjan [22], which constructs a directed, rooted depth-first tree for each connected component from an arbitrary selected node $u \in \mathcal{V}$.

More specifically, for a given graph $G_h$, let $\mu(v)$ be the order number assigned to each node $v \in \mathcal{V}$ during the depth-first search and $\lambda(v)$ be the minimum order number in a cycle that is detected during the search and contains node $v$, just as used in the standard bridge detection algorithm, which yields $\mu(w) = \lambda(w)$ if a link $e = \{v, w\}$ is a bridge. This means that any descendant node of $w$ cannot go to any ancestor nodes of $w$ without passing through the link $e$. Here, we define the disconnection condition by $\mu(v) \leq \lambda(w)$. Based on this algorithm, we can detect each articulation point by examining the disconnection condition. This means that any descendant node of $w$ cannot go to any ancestor nodes of $v$ without passing through the node $v$.

For example, let us consider constructing a rooted depth-first tree from a single component shown in the left in Fig. 2(a) by traversing it from node $A$ in the leftmost manner, which results in the tree shown in the right in Fig. 2(d). As shown in Fig. 2(a), each node of the rooted depth-first tree is basically given its node name and a pair of values $\mu(\cdot)$ and $\lambda(\cdot)$. For example, $(1, 1)$ for node $A$ stands for $\mu(A) = 1$ and $\lambda(A) = 1$. For every node $v$, $\lambda(v)$ is initialized to $\mu(v)$, and, during the construction of the tree, it is updated to $\lambda(w)$ if $\lambda(w) < \lambda(v)$ holds for its child node $w$. For example, when visiting node $G$ first time, $\lambda(G)$ is set to $\mu(G) = 6$ as shown in Fig. 2(a). At the next step in the traversal, we reach node $C$ and can detect a cycle because $\mu(C) = 3$ is smaller than $\mu(G) = 6$, which means node $C$ has already been visited. Therefore, as node $C$ becomes a child node of $G$, $\lambda(G)$ is updated to $\lambda(C) = 3$ as shown in Fig. 2(b). Then, this value of 3 is propagated to $\lambda(F)$ and $\lambda(E)$ during backtracking to node $C$ from node $G$. After the backtrack, the values of $\lambda(G)$, $\lambda(F)$, and $\lambda(E)$ are fixed and never changed. Here, note that $\mu(C) = \lambda(E)$ holds for the link $\{C, E\}$, which satisfies the disconnection condition aforementioned, that is, $\mu(C) \leq \lambda(E)$. Obviously, nodes $E$, $F$, and $G$ cannot reach the ancestor nodes $A$ and $B$ in the tree without passing node $C$. Similarly, we can detect the same condition holds for the link $\{C, H\}$, that is, $\mu(C) = \lambda(H)$ as shown in Fig. 2(c), which implies that nodes $H$, $I$, and $J$ cannot reach the ancestor nodes $A$ and $B$ without passing node $C$, too. Thus, node $C$ can be detected as an articulation point that divides the given component into three sub-graphs by removing it. At the same time, nodes $E$ and $H$ that are child nodes of node $C$ satisfying the disconnection condition become root nodes of sub-trees that are now separated from the ancestor sub-tree. Then, after traversing node $D$ from node $C$, we reach node $A$ again and detect a cycle, which updates $\lambda(D)$ from its initial value of 10 to 1 ($= \lambda(A)$) as node $A$ becomes a child node of node $D$. This value of 1 propagates to $\lambda(C)$ and $\lambda(B)$ during backtracking to node $A$ as shown in Fig. 2(d). Note that the values of $\lambda(E), \lambda(F), \lambda(G), \lambda(H), \lambda(I)$, and $\lambda(J)$ that were updated based on the value of $\lambda(C)$ do not change any more even if $\lambda(C)$ changed from 3 to 1. As aforementioned, their values are fixed and never changed after the backtracking.

Now, let $\{w_1, \cdots, w_J\}$ be a set of such child nodes of $v$ in the depth-first tree, where these child nodes satisfy the disconnection condition, *i.e.*, $\mu(v) \leq \lambda(w_j)$ for $j \in \{1, \cdots, J\}$, while the other ones do not satisfy the disconnection condition. Here note that for arbitrary node $w \in \mathcal{V}$ over the constructed depth-first tree, we can define the number of descendant nodes of $w$ adding $w$ itself as $\theta(w)$. Then, we can express the numbers




(a) The rooted depth-first tree after the 1st visit of node $G$

(b) The rooted depth-first tree after the 2nd visit of node $C$

(c) The rooted depth-first tree after the 3rd visit of node $C$

(d) The rooted depth-first tree after traversing all the nodes

Figure 2: An example of constructing a rooted depth-first tree from a given single component.

of nodes in the $J+1$ connected components after removing the node $v$ as

$$\theta(w_1), \quad \cdots, \quad \theta(w_J), \quad \theta(u) - 1 - \sum_{j=1}^{J} \theta(w_j),$$

where $u$ means the root node of depth-first tree. Figure 3 illustrates the relation among them. Thus, by setting $\eta(v)$ and $\zeta(v)$ as follow:

$$\eta(v) = \sum_{j=1}^{J} \theta(w_j), \quad \zeta(v) = \sum_{j=1}^{J} \theta(w_j)^2,$$

we can compute the degradation value by the following formula.

$$\phi_h(v) = (\theta(u) - 1)^2 - (\theta(u) - 1 - \eta(v))^2 - \zeta(v), \tag{6}$$

Figure 3 also shows how Eq. (6) corresponds to Eq. (3).

Now, we can summarize our proposed method as Algorithm 1. Evidently, for a given $h \in \mathcal{H}$, the computational complexity of the steps 3 to 9 is $O(|\mathcal{E}|)$ which is the same as the standard bridge detection algorithm. Thus, the total computational complexity of our algorithm becomes $O(H \times |\mathcal{E}|)$. Hereafter, we also refer to each degradation value $L_H(v)$ as latent criticalness centrality for a node $v \in \mathcal{V}$ (*LCC* for short), and our proposed algorithm described above as the *LCC* method.

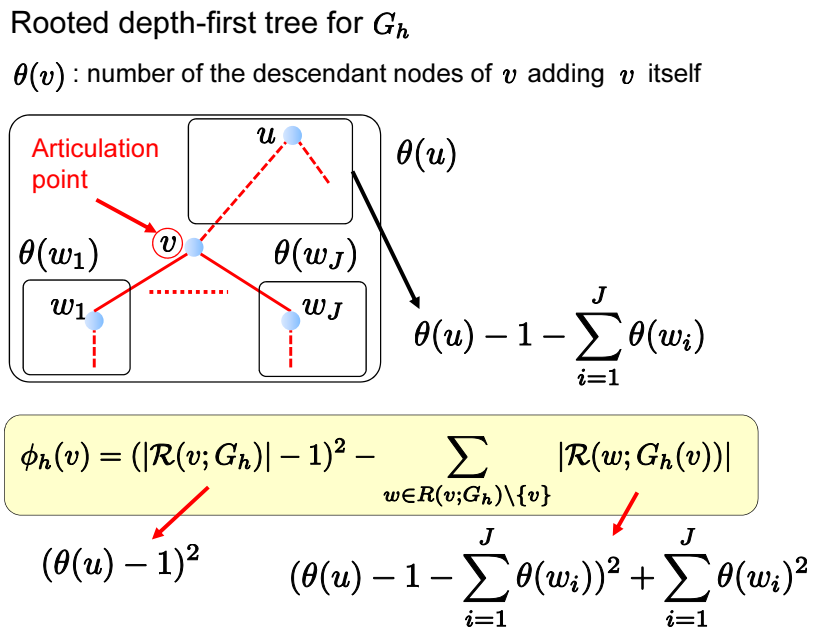



Figure 3: An illustration to explain how to compute the degradation value $\phi_h(v)$ by means of the rooted depth-first tree constructed from a single connected component by focusing on an articulation point.

# 5 Experiments

Using real data of social network $G = (\mathcal{V}, \mathcal{E})$, we evaluated the effectiveness of the proposed method.

## 5.1 Experimental Settings

In this work, we used two social and one road networks. The first one is the Blog network, which is a trackback network of Japanese blogs collected in May, 2005 at the site "goo"[1] and has 12,047 nodes and 39,960 links. The second one is the Enron network derived from the Enron Email Dataset [10]. We constructed this network by regarding each email address as a node and linking two nodes if and only if they have bidirectional communications. The resulting network has 4,254 nodes and 22,157 links. The third one is the Road network of Washington D.C., which is constructed from OSM (OpenStreetMap) data obtained from Metro Extracts [2] in August, 2015 [11]. The resulting network has 114,758 nodes and 128,746 links. The average degree of a node, *i.e.*, the average number of links incident to a node, is 6.63 for the Blog network, 10.42 for the Enron network, and 2.2 for the Road network, while the maximum degree is 222 for the Blog network, 384 for the Enron network, and 8 for the Road network. As the aim of this work is to evaluate the fundamental performance of the proposed method, we set $p_e = p$ for every $e \in \mathcal{E}$, which implies that $p$ is the unique parameter for controlling the disconnection probability. Here we should emphasize that this uniform setting has been widely employed in many previous studies in the filed of information diffusion over social networks [2, 7].

As mentioned earlier, to find critical nodes in a network, several kinds of node-centrality measures have been proposed so far. From among them, we adopted three measures, *i.e.*, degree, eigenvector, and betweenness centralities, as the ones to be compared with the proposed *LCC*. Degree centrality is the most fundamental centrality measure and computable most easily. The degree centrality *DGC* of node $v$ is defined as the number of links incident to $v$ in the network, *i.e.*,

$$DGC(v) = |\{e \in \mathcal{E} | v \in e\}|.$$

[1] http://blog.goo.ne.jp/usertheme/ (currently not available)
[2] https://mapzen.com/data/metro-extracts

**Algorithm 1** Proposed algorithm

**Require:** A set of $H$ graph samples, $\mathcal{G}_H = \{G_h = (\mathcal{V}, \mathcal{E}_h) \mid h \in \mathcal{H}\}$.
**Ensure:** The set of reachability degradation values for $\mathcal{G}_H$, $\{L_H(v) \mid v \in \mathcal{V}\}$.
1: Initialize $L_H(v)$ as $L_H(v) \leftarrow 0$ for every $v \in \mathcal{V}$.
2: **for all** $h \in \mathcal{H}$ **do**
3: Initialize $\theta(v)$, $\eta(v)$ and $\zeta(v)$ as $\theta(v) \leftarrow 0$, $\eta(v) \leftarrow 0$ and $\zeta(v) \leftarrow 0$ for every $v \in \mathcal{V}$.
4: Compute connected components by repeatedly doing the depth-first search from a node $u \in \mathcal{V}$ after setting $\mu(u) \leftarrow 1$.
5: **for all** link $e = \{v, w\}$ obtained during the depth-first search **do**
6: Compute $\mu(w)$, $\lambda(w)$ and $\theta(w)$ by the depth-first search.
7: Declare $v$ as an articulation point if the disconnection condition meets, *i.e.*, $\mu(v) \leq \lambda(w)$, and then update $\eta(v)$ and $\zeta(v)$ as $\eta(v) \leftarrow \eta(v) + \theta(w)$ and $\zeta(v) \leftarrow \zeta(v) + \theta(w)^2$, respectively.
8: **end for**
9: Update $L_H(v)$ as $L_H(v) \leftarrow L_H(v) + (\theta(u) - 1)^2 - (\theta(u) - 1 - \eta(v))^2 - \zeta(v)$ for every $v \in \mathcal{V}$ if the node $v$ is an articulation point.
10: **end for**
11: **return** $\{L_H(v) \mid v \in \mathcal{V}\}$ after setting $L_H(v) \leftarrow L_H(v)/H$ for every $v \in \mathcal{V}$.

Eigenvector centrality is thought as a natural extension of degree centrality because when computing the centrality value of each node, not only the number of its neighboring nodes as in degree centrality, but also their centrality values in a given network are considered. More specifically, the eigenvector centrality $EVC$ of node $v$ is defined as

$$EVC(v) = \lambda^{-1} \sum_{u \in n(v)} EVC(u),$$

where $\lambda$ is the first eigenvalue of the adjacency matrix of a given network $G$ and $n(v)$ is the set of neighboring nodes of $v$, *i.e.*, $n(v) = \{u \in \mathcal{V} | \{v, u\} \in \mathcal{E}\}$. Generally, $EVC$ is computed through solving the equation $A_G \mathbf{x} = \lambda \mathbf{x}$, where $A_G$ is the adjacency matrix of $G$ and $\mathbf{x}$ is the vector consisting of values of $EVC$ for every node. Betweenness centrality $BWC$ takes into account the global topology of a given network unlike degree centrality that considers only the local topology surrounding the target node, and is defined as follows:

$$BWC(v) = \sum_{u \in \mathcal{V}} \sum_{w \in \mathcal{V}} \{N^{sp}(u, w; v)/N^{sp}(u, w)\},$$

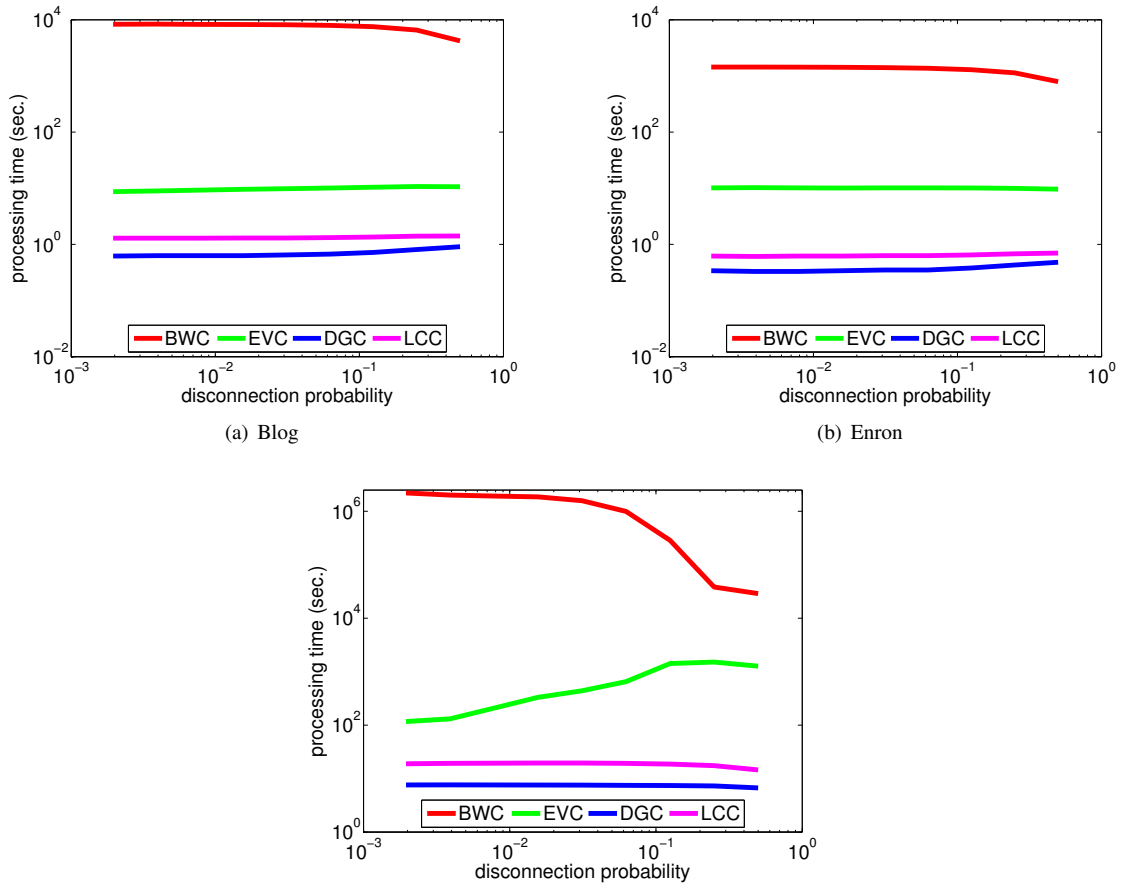
where $N^{sp}(u, w)$ stands for the number of the shortest paths from node $u$ to node $w$, and $N^{sp}(u, w; v)$ denotes the number of those paths passing through a node $v$.

Note that we actually used $DGC(v; \mathcal{G}_H)$, $EVC(v; \mathcal{G}_H)$, and $BWC(v; \mathcal{G}_H)$ as conventional node-centrality values for each node $v \in \mathcal{V}$ that are respectively defined as the average over $H$ graphs just like $L_H(v)$ defined by Eq. (4).

## 5.2 Experimental results

First, we evaluated the computational efficiency of the proposed $LCC$ method by comparing it with $BWC$, $EVC$, and $DGC$ in processing time to compute the average centralities for $\mathcal{G}_H$ with a setting $H = 10^3$. In this work, we set the disconnection probability $p_e = p$ to $2^{-k}$ for every link $e \in \mathcal{E}$ and varied the integer $k$ from 1 to 9, leading to the range of $p$, $(0.0019, 0.5]$. Figure 4 shows the experimental results [3]. Figures 4(a), 4(b), and 4(c) are the results for the Blog, Enron and Road networks, respectively. From Fig. 4, it can be observed that these three networks having different sizes exhibit quite similar tendency. The processing times for computing $LCC$ were substantially smaller than those for $BWC$ and $EVC$ although they were roughly twice as large as those for $DGC$, most efficiently computable one. The reason why $BWC$ is most expensive is that its computational complexity approximately becomes a square order of the network size. $EVC$ needs power-iterations for solving the eigenvector equation. It is noted that the processing times for $BWC$ are likely

[3] Our programs were written in C, and run on a computer with Xeon X5690 3.47GHz CPUs using a single thread within a 192GB main memory capacity.




(a) Blog (b) Enron

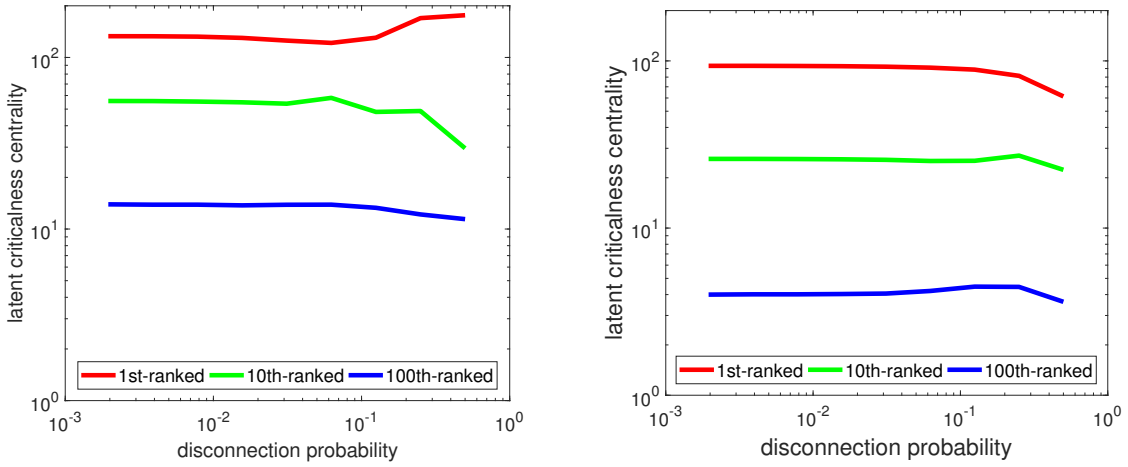
(c) Road

Figure 4: Evaluation of computational efficiency as a function of the link disconnection probability.

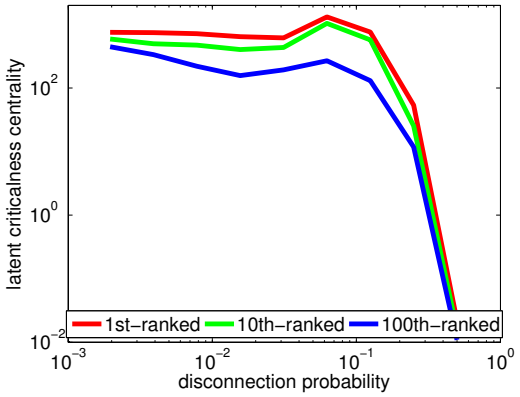
to decrease when the disconnection probability $p$ becomes large, especially for the Road network. This is because the number of disconnected node pairs increases for a large $p$. From these results, we can claim that our proposed centrality *LCC* has a desirable scalability with respect to the size of networks.

Next, we investigated, in terms of latent criticalness centrality, how the performance of the 1st-ranked, 10th-ranked, and 100th-ranked nodes in $L_H(e)$ changes according to the link disconnection probability $p \in \{2^{-1}, \cdots, 2^{-9}\}$. Figure 5 shows our experimental results, where Figs 5(a), 5(b), and 5(c) are also those for the Blog, Enron and Road networks, respectively. From Figs. 5(a) and 5(b) for the Blog and Enron networks, respectively, we can observe quite similar tendency for these two social networks, *i.e.*, the latent criticalness centralities are almost stable, but slightly decrease for large disconnection probabilities. This is attributed to the fact that the original networks are separated to a large number of connected components of smaller sizes when $p$ is large, and thus the number of node pairs that are disconnected by removing a specific node becomes smaller accordingly. The only exception is the 1st-ranked node of the Blog network. Its latent criticalness centrality increased even for large values of $p$. This would be because the 1st-ranked node has an exceptionally large criticalness centrality value as shown in the experimental result shown below. In general, we expect that these 1st-ranked nodes have relatively large betweenness and degree centrality values. Removing such a node from the network could substantially degrade the network performance in node reachability. This expectation will also be justified by the experimental results shown below.

On the other hands, from Fig. 5(c) for the Road network, we observe that the differences between latent criticalness centralities from the 1st- to 100th-ranked nodes are relatively quite small, and these values rapidly decreased for large values of $p$. This observation can be naturally explained by the fact that typical hub nodes with high degrees do not exist in the road network, *i.e.*, the average and maximum degrees for the Road




(a) Blog (b) Enron



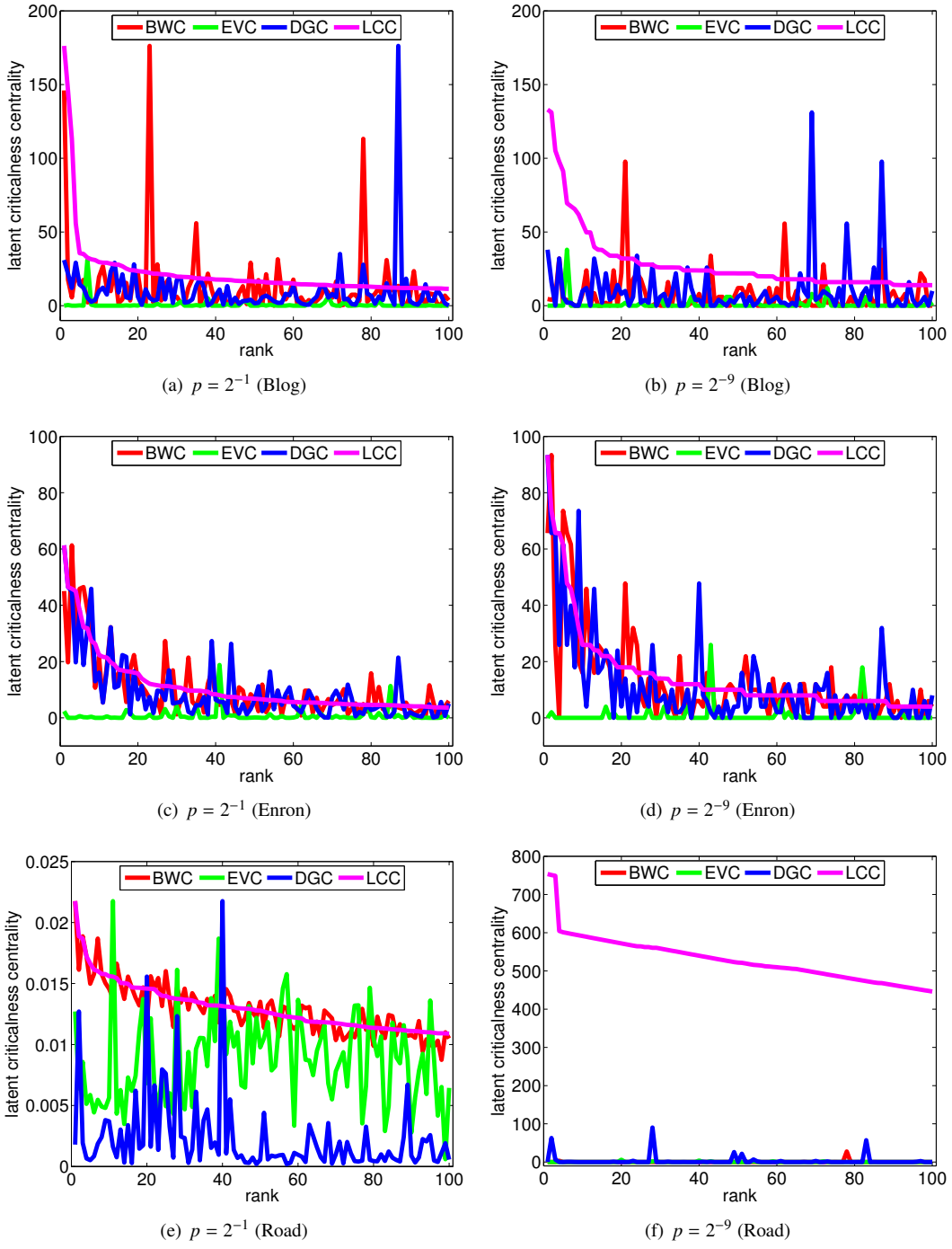

(c) Road

Figure 5: Evaluation of criticalness centrality of the 1st-ranked, 10th-ranked and 100th-ranked nodes as a function of the link disconnection probability.

network are sufficiently smaller than those for the Blog and Enron networks. In other words, this is the reason that the Road network is likely to be separated to a quite large number of connected components of smaller sizes when $p$ is large.

From semantic viewpoint, considering the vertical axes of these figures are logarithmic, the top-ranked node has much larger latent criticalness centrality than the 10th-ranked and 100th-ranked nodes for both the social networks, which means the top-ranked node is substantially more critical than the others for maintaining the connectivity of each network under the assumption of the probabilistic link disconnection model. However, the difference from the top is not as large for the road network, implying that it is not the case that there exist a single critical node. In the context of information diffusion such as the ones we consider here, such a node corresponds to an influencer in a network, and many people would become unable to get information if he/she stops sending the information. On the other hands, in the context of disaster evacuation over a road network, such a node corresponds to a critical intersection, and many people would fail to reach an evacuation facility within a reasonable time if they cannot pass the intersection. Our experimental results suggest that there exist not a small number of critical intersections even when $p$ is small.

We further examined the performance of highly ranked nodes by the conventional centralities, *i.e.*, $BWC$, $EVC$, and $DGC$ in terms of latent criticalness centrality $L_H(v) = LCC(v)$. More specifically, we evaluated the performance defined by $L_H(v_i^{(C)})$, where $C$ denotes one of $\{BWC, EVC, DGC, LCC\}$ and $v_i^{(C)}$ stands for the node with the $i$-th rank in each centrality measure $C$. Figure 6 shows the experimental results, where the horizontal and vertical axes denote the ranking of nodes up to top-100 and the performance evaluated in




(a) $p = 2^{-1}$ (Blog)

(b) $p = 2^{-9}$ (Blog)

(c) $p = 2^{-1}$ (Enron)

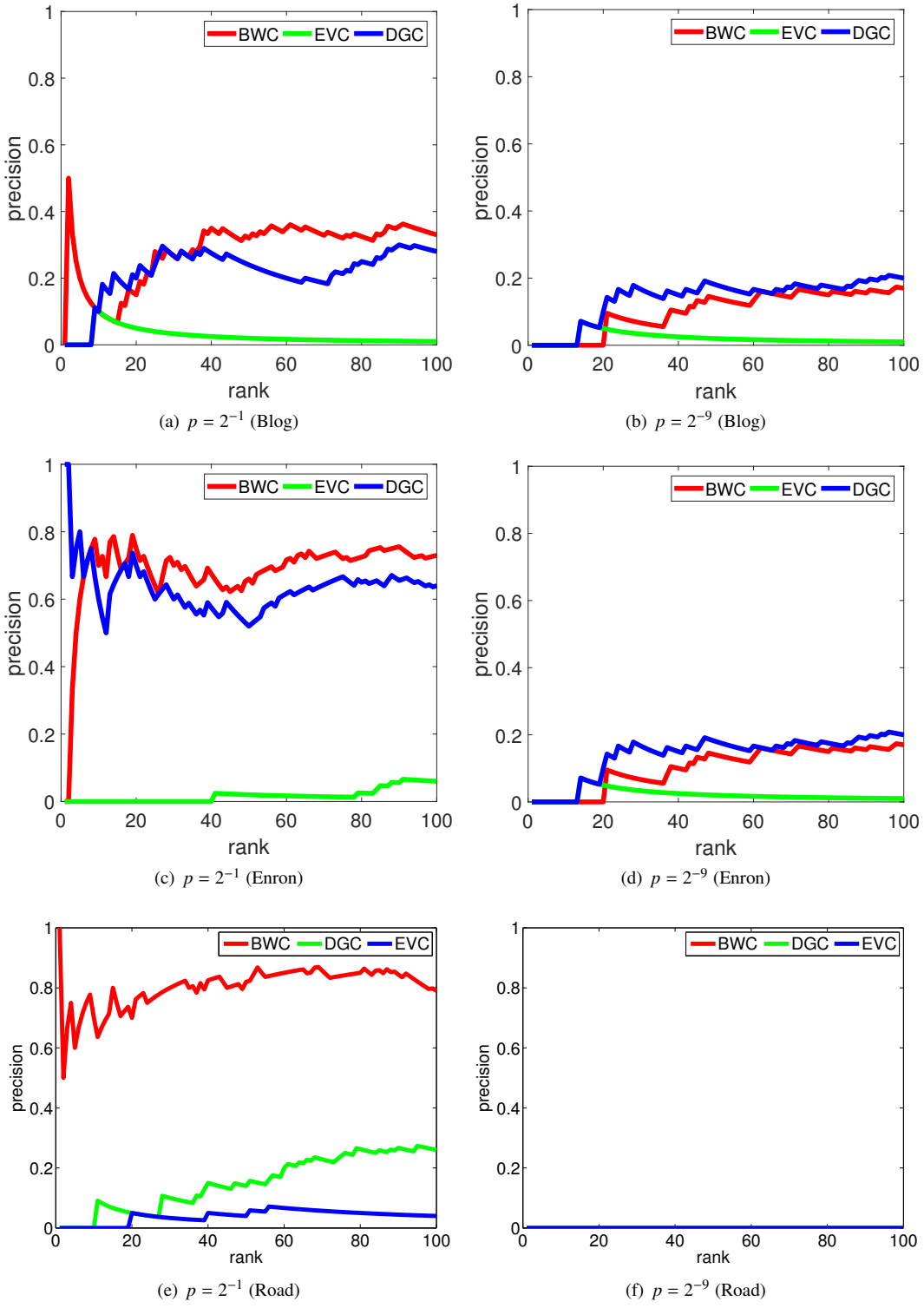
(d) $p = 2^{-9}$ (Enron)

(e) $p = 2^{-1}$ (Road)

(f) $p = 2^{-9}$ (Road)

Figure 6: Evaluation of criticalness centrality of the top-100 nodes obtained by each centrality measure for different link disconnection probabilities ($p = 2^{-1}$ and $2^{-9}$).

the latent criticalness centrality measure $L_H(v)$, respectively. The pairs of Figs. 6(a) and 6(b), Figs. 6(c) and 6(d), and Figs. 6(e) and 6(f) are the results for the Blog, Enron and Road networks, respectively. Each pair is the results performed with $p = 2^{-1}$ and $2^{-9}$. We chose these two extreme cases: $p = 2^{-9}$ that provides the closest ones to the original networks and $p = 2^{-1}$ that results in the latent criticalness centrality values that are clearly different from the ones observed in the other settings. From these results, except for the result of the Road network with $p = 2^{-9}$ shown in Fig. 6(f), we can see that some of the high ranked nodes in *BWC*




(a) $p = 2^{-1}$ (Blog)

(b) $p = 2^{-9}$ (Blog)

(c) $p = 2^{-1}$ (Enron)

(d) $p = 2^{-9}$ (Enron)

(e) $p = 2^{-1}$ (Road)

(f) $p = 2^{-9}$ (Road)

Figure 7: Precision comparison in criticalness centrality of the top-100 nodes obtained by each centrality measure for different link disconnection probabilities ($p = 2^{-1}$ and $2^{-9}$).

and *DGC* tend to have a large value in *LCC*, too. But, since the results for *BWC* and *DGC* fluctuate wildly, a node does not always have a large latent criticalness centrality even if it has a large value either in *BWC* or

in *DGC*. On the other hand, except for the result of the Road network with $p = 2^{-1}$ shown in Fig. 6(e), we can see that the performance for most of the top-100 nodes by *EVC* are almost zeros despite the value of $p$ and the network used. In short, these experimental results suggest that our latent criticalness centrality has unique properties and it is difficult to identify nodes having a large latent criticalness centrality by means of conventional centrality measures.

Here note that, in Fig. 6(a) ($p = 2^{-1}$), we can confirm that the 1st-ranked node has an exceptionally large criticalness centrality value and forms a peak in comparison to the other 1st-ranked nodes shown in from Figs. 6(b) to 6(f). We consider that this exception caused the difference between the nodes' behavior in Fig. 5. Moreover, in Fig. 6(a), as the highest peaks for *LCC*, *BWC*, and *DGC* stand for the identical node having a value of 176.2 in *LCC*, we can find that the top-1 node of the Blog network in *LCC* has a large value in *BWC* (the 23rd largest value) and a relatively large value in *DGC* (the 87th largest value). Thus, its removal from the network could increase the number of disconnected node pairs, which results in the increase of its latent criticalness centrality. We can also observe this tendency in the other figures in Figs. 6(b) to 6(e) except 6(f). For instance, in Figs. 6(c), 6(d) and 6(e), the top-1 nodes of the Enron and Road networks in *LCC* have large values also in *BWC* (the 3rd, 2nd and 1st largest values, respectively), and in Figs. 6(b) and 6(e), the top-1 nodes of the Blog and Road networks in *LCC* have large values also in *DGC* (the 70th and 40th largest values, respectively),

Next, we further investigated the similarity between the ranking based on *LCC* and the ranking based on the other centralities, *BWC*, *DGC*, and *EVC* in the cases of $p = 2^{-1}$ and $p = 2^{-9}$ for the three networks. More specifically, we measured the similarity between the top $k$ nodes for the ranking based on *LCC*, denoted by $\mathcal{A}_k^{(LCC)} = \{v_i^{(LCC)}\}_{i=1}^k$, and those for the ranking based on the other centralities, *BWC*, *DGC*, and *EVC*, denoted by $\mathcal{A}_k^{(C')} = \{v_i^{(C')}\}_{i=1}^k$, where $C'$ stands for one of $\{BWC, DGC, EVC\}$, by using the precision $Prec(k)$ defined as follows:

$$Prec(k) = \frac{|\mathcal{A}_k^{(LCC)} \cap \mathcal{A}_k^{(C')}|}{k}.$$

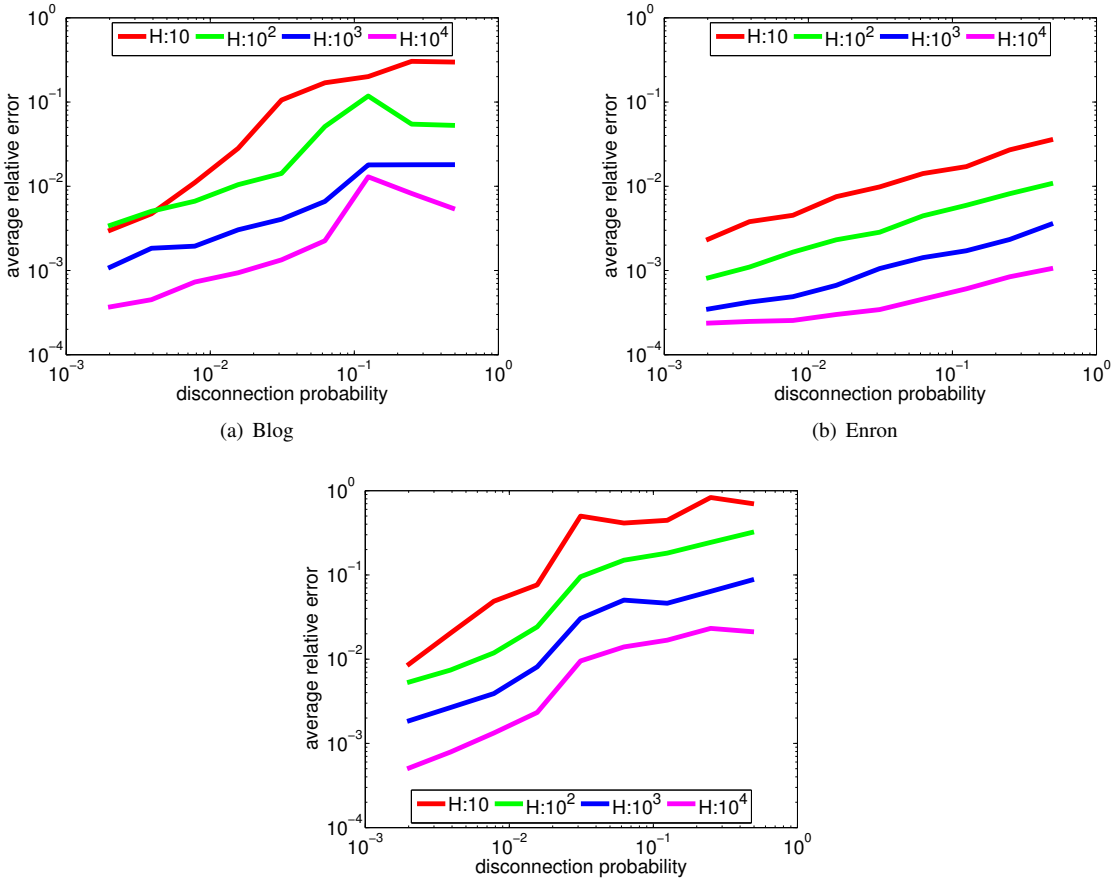
Figure 7 shows the results, where the horizontal and vertical axes denote the rank $k$ up to top-100 and the precision $Prec(k)$, respectively. Again, the pairs of Figs. 7(a) and 7(b), Figs. 7(c) and 7(d), and Figs. 7(e) and 7(f) are the results for the Blog, Enron and Road networks, respectively, where each pair is the results performed with $p = 2^{-1}$ and $2^{-9}$. From these results, we can also see that there does not exist any similarity between *LCC* and *EVC* for all the three networks. On the other hand, it is found that *BWC* for the Enron and Road networks, and *DGC* for the Enron network succeed in identifying around 50 to 80% of the top-$k$ nodes for the ranking based on *LCC* in the case of $p = 2^{-1}$, while they do not work as well for the Blog network.

Indeed, for the Blog network, their precisions are less than 0.4 in most cases in the case of $p = 2^{-1}$, and less than about 0.2 in the case of $p = 2^{-9}$, respectively. These tendencies coincide with the ones shown in Fig. 6, in which *BWC* and *DGC* fluctuate more closely around the change of *LCC* for the Enron network than they do for the Blog network. On the other hand, although some of the high ranked nodes in *EVC* and *DGC* tend to have a large value in *LCC* as shown in Fig. 6(e), we can see from Fig. 7(e) of the Road network with $p = 2^{-1}$ that their precisions are less than 0.3. This can be naturally explained by the fact that there exist many nodes with a large value in *LCC* as shown in Fig. 5(c), These results suggest that *EVC* is not useful at all for finding nodes having a high *LCC* value, and that *BWC* and *DGC* may be helpful to some degree for some kinds of uncertain networks, but their performance highly depends on the topological characteristics of the given network.

Finally, we evaluated the approximation accuracy by changing $H$. More specifically, we computed the latent criticalness centrality $L_H(v)$ for each node $v$ by setting $H = 1,000,000$, and regarded them as the ground truth by setting $L^*(v) \leftarrow L_H(v)$. Then, we computed each set of 100 estimation results, $\{L_H^{(i)}(v) \mid 1 \le i \le 100\}$, which are obtained by a series of independent Bernoulli trials for $H = 10, 10^2, 10^3$, and $10^4$, and evaluated them in terms of the relative error $RE_H(v)$ defined by

$$RE_H(v) = \frac{1}{100} \sum_{i=1}^{100} \left| \frac{L_H^{(i)}(v) - L^*(v)}{L^*(v)} \right|. \tag{7}$$

Figure 8 shows the experimental results of the top-1 node according to $L^*(v)$ for each probability setting of $p \in \{2^{-1}, \cdots, 2^{-9}\}$. Figs. 8(a), 8(b) and 8(c) are the results for the Blog, Enron and Road networks, respectively. Here we should note that we obtained similar experimental results for the other nodes of lower




(a) Blog

(b) Enron

(c) Road

Figure 8: Fluctuation of the relative error of the criticalness centrality as a function of the link disconnection probability for different numbers of simulations.

ranks. From Fig. 8, we can see for these three networks that the relative errors were less than 10% for any disconnection probability $p$ even by setting $H = 10^2$ for the Blog network, $H = 10$ for the Enron network and $H = 10^3$ for the Road network. In short, these experimental results suggest that we can stably compute our latent criticalness centrality without sampling very many graphs.

# 6 Conclusion

We addressed the problem of efficiently identifying critical nodes under a realistic situation where all the links in a network are probabilistically blocked. A node is critical if it has a large contribution value, that is a measure of performance degradation when the node and links incident to it are removed. We formalized this situation as the probabilistic link disconnection model, similar to studies on uncertain graphs, and proposed a novel algorithm that can efficiently identify critical nodes by incorporating the bridge detection technique to the algorithm to search for articulation points in case the node reachability is taken as the performance measure. Using two real-world social networks and one real-world road network, we experimentally showed that the proposed method can identify nodes that are more critical than those detected by the other methods based on the traditional centrality measures with much less computation time. Especially, through the comparison of the top-100 nodes based on our proposed latent criticalness centrality and the traditional node centralities (betweenness, degree, and eigenvector), it is empirically shown that it is difficult for the traditional centralities to identify critical nodes detected by the latent criticalness centrality. The betweenness and degree centralities may be helpful for identifying some of critical nodes having a high value in the latent criticalness centrality,

but their performance highly depends on the topological characteristics of a given network. The eigenvector centrality does not exhibit any similarity to the latent criticalness centrality when comparing those top-ranked nodes.

Our immediate future work is to extend the proposed method to be able to deal with directed networks. Besides, we are planning to evaluate the critical nodes detected by our method from a more realistic viewpoint. For example, although we assumed that the link disconnection probability for each link is uniform in our experiments, we can define the probability to be proportional to the inverse of communication frequency of each link in the case of communication networks such as the Enron network we used, which could be a more reasonable assumption. On the other hand, investigating the relationships between critical nodes identified based on our proposed centrality and those based on other centralities more in depth by constructing synthetic networks having various kinds of topological features and conducting experiments on them is also one of the possible directions of this work.

## Acknowledgment

This material is based upon work supported by JSPS Grant-in-Aid for Scientific Research (C) (No. 17K00314).

## References

[1] V. K. Akram and O. Dagdeviren. Breadth-first search-based single-phase algorithms for bridge detection in wireless sensor networks. *Sensors*, 13(7):8786 – 8813, 2013.

[2] W. Chen, L.V.S. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[3] P. Crucitti, V. Latora, and S. Porta. Centrality measures in spatial networks of urban streets. *Physical Review E*, 73(3):036125, 2006.

[4] L. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[5] R. Jin, L. Liu, C. Aggarwal, and Y. Shen. Reliable clustering on uncertain graphs. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM 2012)*, pages 459 – 468, 2012.

[6] V. Kassiano, A. Gounaris, A. N. Papadopoulos, and K. Tsichlas. Mining uncertain graphs: An overview. In *Proceedings of the International Workshop on Algorithmic Aspects of Cloud Computing (ALGOCLOUD 2016)*, pages 87 – 116, 2017.

[7] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. *Theory of Computation*, 11:105–147, 2015.

[8] A. Khan, Y. Ye, and Chen L. *On Uncertain Graphs*. Morgan & Claypool, 2018.

[9] M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data*, 3:9:1–9:23, 2009.

[10] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the 2004 European Conference on Machine Learning (ECML'04)*, pages 217–226, 2004.

[11] K. Ohara, K. Saito, M. Kimura, and H. Motoda. Accelerating computation of distance based centrality measures for spatial networks. In *Proceedings of the 19th International Conference on Discovery Science (DS'16)*, pages 376–391. LNCS 9956, 2016.

[12] K. Ohara, K. Saito, M. Kimura, and H. Motoda. Maximizing network performance based on group centrality by creating most effective k-links. In *Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA'17)*, pages 561 – 570, 2017.

[13] K. Ohara, K. Saito, M. Kimura, and H. Motoda. Critical node identification based on articulation point detection for network with uncertain connectivity. In *Proceedings of the Sixth International Symposium on Computing and Networking (CANDAR 2018)*, pages 146 – 152, 2018.

[14] E. L. Oliveira, L. S. Portugal, and W. Porto Junior. Determining critical links in a road network: vulnerability and congestion indicators. *Procedia - Social and Behavioral Sciences*, 162:158 – 167, 2014.

[15] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, 2010.

[16] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. In *Proceedings of the VLDB Endowment (PVLDB)*, volume 3, pages 997 – 1008, 2010.

[17] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Detecting critical links in complex network to maintain information flow/reachability. In *Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence (PRICAI2016)*, pages 419–432, 2016.

[18] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Super mediator - a new centrality measure of node importance for information diffusion over social network. *Information Sciences*, 329:985–1000, 2016.

[19] K. Saito, M. Kimura, K. Ohara, and H. Motoda. An accurate and efficient method to detect critical links to maintain information flow in network. In *Proceedings of the 23th International Symposium on Methodologies for Intelligent Systems (ISMIS2017)*, pages 116–126, 2017.

[20] A. E. Sariyüce, K. Kaya, E. Saule, and Ü. V. Çatalyürek. Graph manipulations for fast centrality computation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(3), 2017.

[21] Y. Shen, N. P. Nguyen, Y. Xuan, and M. T. Thai. On the discovery of critical links and nodes for assessing network vulnerability. *IEEE/ACM Transaction on Networking*, 21(3):963 – 973, 2013.

[22] R. E. Tarjan. A note on finding the bridges of a graph. *Information Processing Letters*, 2(6):160 – 161, 1974.