Measuring Robustness of Complex Networks under MVC Attack

Rong-Hua Li, Jeffrey Xu Yu, Xin Huang, Hong Cheng, Zechao Shang
The Chinese University of Hong Kong, Hong Kong, China
{rhli,yu,xhuang,hcheng,zcshang}@se.cuhk.edu.hk

ABSTRACT

Measuring robustness of complex networks is a fundamental task for analyzing the structure and function of complex networks. In this paper, we study the network robustness under the maximal vertex coverage (MVC) attack, where the attacker aims to delete as many edges of the network as possible by attacking a small fraction of nodes. First, we present two robustness metrics of complex networks based on MVC attack. We then propose an efficient randomized greedy algorithm with near-optimal performance guarantee for computing the proposed metrics. Finally, we conduct extensive experiments on 20 real datasets. The results show that P2P and co-authorship networks are extremely robust under the MVC attack while both the online social networks and the Email communication networks exhibit vulnerability under the MVC attack. In addition, the results demonstrate the efficiency and effectiveness of our proposed algorithms for computing the corresponding robustness metrics.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; G.2.2 [**Discrete mathematics**]: Graph theory—*Graph algorithms*

General Terms

Algorithm, Theory, Experimentation

Keywords

Network robustness, FM sketch, Submodular function, MVC attack,

1. INTRODUCTION

Networks are ubiquitous. Many practical systems in nature and society can be characterized by the network. Examples include (online) social networks, computer networks, Internet, biological networks, transportation networks, and so on. After the seminal work by Watts and Strogatz [20] and Barabási and Albert [2], complex networks have attracted increasing attention in both industry and research communities in the last decade. The studies on complex network theory mainly focus on investigating the underlying organizing principles, the function, and the dynamics of the network.

In general, the function and performance of a network depend on its robustness [1, 21], i.e., the ability of a network to tolerate the nodes or links error. For example, in an airline network, the robustness reflects its operational ability given certain airports are closed. In a computer network, the robustness denotes its communication capacity provided some computers in the network crash. In P2P networks, the robustness represents the ability of a network to still work well when some peers depart from the network. In online social networks, the robustness signifies the ability of a network to connect well when some users withdraw from the network. In co-authorship networks, the robustness stands for the ability of a network that the co-authorship does not significantly reduce when some scholars leave the research community.

When we measure a network, a fundamental problem is how to assess its robustness. Due to a large number of applications, measuring robustness of a network receives growing attention. Early works on robustness measurement are based on the connectivity of a network. In the literature, there are a considerable number of connectivity metrics. Examples include the algebraic connectivity [9], super connectivity [3], conditional connectivity [13], and isoperimetric number [16]. However, the connectivity-based robustness measures only consider the topological structure of the network, but they ignore the concrete node or link error process. This may result in some networks with lage connectivity but they are easily attacked by intended node or link attack. To address this problem, Albert et al. [1] study the robustness of a network by considering some statistical properties of the network after the deletion of a small fraction of nodes. Many subsequent studies [5, 4, 6] follow this framework to study the robustness of the network. However, most of them focus on the analytical solution of the robustness metric on the basis of some specific random graph models. More recently, Schneider et al. [19] present a robustness metric based on the size of the giant connected component. However, they do not provide a detailed algorithm for computing their metric, and the complexity for calculating their metric is unknown. To summarize, the potential challenges for measuring the robustness of complex network include: (1) how to define a

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

reasonable and intuitive robustness metric, and (2) how to develop an efficient and effective algorithm for calculating the robustness metric.

To address these challenges, in this paper, we study the robustness of the complex network from an attacker's point of view. Specifically, we measure the robustness of a network based on the minimal number of residual edges after removing a small fixed budget of k nodes of the network. The number of residual edges is a very natural and intuitive metric for measuring the function and performance of the network. Intuitively, after the removal of k nodes, the network with a large number of residual edges implies that the function and performance of the network are not extensively damaged. As a consequence, the larger number of residual edges suggest the better robustness of a network. On the other hand, from the attacker's point of view, the attacker wants to maximize the number of edges that are deleted after attacking a budget of k nodes. This problem is equivalent to the maximal vertex coverage (MVC) problem on networks [12]. We refer to such type of attack as the MVC attack. There are many practical applications that can suffer from such MVC attack. For instance, in computer networks, the hacker may want to attack k workstations so as to minimize the number of surviving links in the network. In online social networks, the attacker may want to target k users by providing some incentives to persuade them to leave the social network so as to minimize the number of residual social ties of the network. Consequently, it is very important to measure the robustness of a network under the MVC attack.

Based on the MVC attack, we present two new robustness metrics of the network, namely k-robustness and cumulative k-robustness. More specifically, the k-robustness is defined by the minimal fraction of the residual edges after removing k nodes, and the cumulative k-robustness is the average k-robustness from $k = 1$ to k. To compute the two proposed robustness metrics, we propose a randomized greedy algorithm with near-optimal approximation guarantee for calculating our robustness metrics efficiently. To the best of our knowledge, our work is the first work for measuring robustness of complex networks under the MVC attack. We conduct extensive experimental studies on 20 real datasets. The results show that the P2P and co-authorship networks are extremely robust under the MVC attack, whereas the online social networks, the Email communication networks, as well as the web graph are shown to be very vulnerable under the MVC attack. Also, the results confirm the effectiveness and efficiency of the proposed algorithms for computing the corresponding robustness measures.

2. PROBLEM FORMULATION

Consider an undirected and unweighted network $G = (V, E)$, where V denotes a set of nodes and E denotes a set of undirected edges between the nodes. Let $n = |V|$ and $m = |E|$ be the number of nodes and the number of edges in G, respectively. The problem that we address in this paper is to measure the robustness of the network from an attacker's perspective.

We consider the following setting. Assume that there is an attacker who wants to attack a network, and the attacker has a budget of k nodes to attack. If a node is attacked by the attacker, then the node and its incident edges will be removed from the network. The attacker aims to maximize some utility functions by attacking k nodes. From the robustness point of view, our goal is to evaluate the robustness of the network under such attack.

In this paper, we introduce a utility function for the attacker. That is, the number of edges that are removed by attacking k nodes. In other words, the goal of the attacker is to maximize the number of edges that are removed after deleting k nodes. Note that this problem is equivalent to the maximal vertex coverage (MVC) problem [12] which aims to select k nodes that cover as many edges as possible. Therefore, we refer to such an attack as the MVC attack. Formally, let S $(S \subseteq V)$ be a set of nodes, and $F(S)$ be the number of edges that are removed after deleting the nodes in S. Then, the MVC attack problem can be formulated as

$$\begin{aligned} &\max_{S\subseteq V} \ F(S) \\ &s.t. \quad |S| \le k. \end{aligned} \tag{1}$$

Let $F^*(S)$ be the optimal solution for Eq. (1). Then, we define the k-robustness of a network G as follows.

Definition 2.1: Given a network $G = (V, E)$, the k-robustness of G is $\sigma_k = 1 - F^*(S)/m$.

By Definition 2.1, the k-robustness (σ_k) denotes the fraction of residual edges after removing k nodes. Intuitively, after removing k nodes, the larger the fraction of residual edges is, the more robust the network is. In addition, it can be seen that σ_k falls into the interval [0, 1]. If $\sigma_k = 0$, we say a network is completely collapsed. We refer to the minimal k that causes $\sigma_k = 0$ as the collapsed point denoted by $\tilde{k}$. Clearly, $\tilde{k}$ equals to the minimum vertex cover number of a network. Note that σ_k measures the *point-wise* robustness of a network. Naturally, we define the cumulative k-robustness as follows.

Definition 2.2: Given a network $G = (V, E)$, the cumulative k-robustness of G is $\bar{\sigma}_k = (\sum_{i=1}^{k} \sigma_i)/k$.
Unlike the k-robustness, the cumulative k-robustness evaluates the average point-wise robustness. According to Definitions 2.1 and 2.2, large σ_k and $\bar{\sigma}_k$ indicate a high robustness of the network. Note that the key subroutine to compute the cumulative k-robustness ($\bar{\sigma}_k$) is to calculate the k-robustness (σ_k). In the following section, we focus on how to compute σ_k efficiently.

3. ALGORITHMS

Given a network G, the key issue to evaluate the k-robustness of G is to solve the MVC attack problem (Eq. (1)). Unfortunately, finding the MVC on general networks has been known to be NP-complete [7]. Hence, there is no hope to exactly compute σ_k in polynomial time. In this section, we present a randomized greedy algorithm with near-optimal performance guarantee for computing σ_k efficiently. First, we briefly describe the concept of nondecreasing submodular set function. Let A be a finite set. A set function F defined on the subsets of A is a nondecreasing submodular function if the following condition holds. For any subsets B and C such that $B \subseteq C \subseteq A$, and for any element $j \notin C$, we have $\rho_j(B) \ge \rho_j(C) \ge 0$, where $\rho_j(B)$ represents the marginal gain and it is defined as $\rho_j(B) = F(B \cup \{j\}) - F(B)$.

It is easy to check that $F(S)$ is a nondecreasing monotone submodular set function. Based on this, there exists a greedy algorithm for solving the MVC problem (Eq. (1)) efficiently. In particular, the greedy algorithm works in k rounds. At each round, the algorithm finds the node with the maximal marginal gain ($\rho_j(S)$) and adds it into the optimal node set S, where S is initialized to be an empty set. By a celebrated result [17], this greedy algorithm can achieve a $1-1/e$ approximate ratio. The time complexity of the greedy algorithm is $O(km)$, because the algorithm needs to visit all the edges to find the node with the maximal marginal gain in the worst case. Below, we will propose a more efficient randomized greedy algorithm using the well-known Flajolet-Martin (FM) sketch [10].

The FM sketch is a probabilistic counting data structure and it can be utilized to estimate the cardinality of a multi-set [10]. Let N be the cardinality of a multi-set A. Then, the FM sketch only uses $\log N + c$ bits for estimating N accurately, where c is a small constant. In particular, the FM sketch is a bitmap with size $l = \log N + c$. There exists a hash function $h : A \to \{1, \cdots, l\}$, mapping an element a ($a \in A$) to a bit i ($i \in \{1, \cdots, l\}$) in the bitmap with probability $\Pr(h(a) = i) = 1/(2^{i+1})$. At the beginning, all the bits in the bitmap are set to 0. Then, for processing an element a ($a \in A$), we set the corresponding $h(a)$-th bit of the bitmap to 1. Finally, an asymptotically unbiased estimator for the cardinality N can be obtained by $2^z/0.77351$, where z denotes the position of the least-significant zero bit in the bitmap. Another important property of the FM sketch is that it can be easily used to estimate the cardinality of the union of two multi-sets if these two multi-sets come from the same domain. Specifically, we construct two FM sketches with the same size for two multi-sets respectively. To estimate the cardinality of the union of two multi-sets, we only need to do a bitwise-OR between the two FM sketches, and then estimate the cardinality based on the resulting FM sketch. To enhance the estimation accuracy, we can make use of multiple hash functions. For convenience, we only consider one hash function to describe our algorithm.

The key idea of our algorithm is described as follows. For each node u, we create an FM sketch to sketch the incident edges of u and use it to estimate $F(\{u\})$. Then, for any set S, $F(S)$ can be calculated by

$$F(S) = |\bigcup_{u \in S} E(\{u\})|, \tag{2}$$

where $E(\{u\})$ denotes the set of incident edges of node u. Note that $E(\{u\})$ can be represented by an FM sketch. As a result, for any set S, we can estimate $F(S)$ by performing $|S|$ times bitwise-OR operation. Our algorithm is described in Algorithm 1. Firstly, Algorithm 1 creates an FM sketch for each node $v_i \in V$ (line 2-5). In particular, for each node v_i, we initialize a bitmap FM[i], i.e., set all the bits of FM[i] to 0 (line 3). For all the incident edges of node v_i, we insert them into the bitmap FM[i] by setting the corresponding bits to 1 (line 4-5). Secondly, Algorithm 1 greedily chooses k nodes based on their approximate marginal gain (line 6-22). Specifically, we create two FM sketches CFM and OFM and use them to estimate the current optimal solution and the current marginal gain, respectively. Algorithm 1 works in k rounds. At each round, it selects the node with the maximal approximate marginal gain (line 12-19). To compute the approximate marginal gain of node v_i (denoted by $\hat{\rho}_i$), we only need to do a bitwise-OR between the FM sketches CFM and FM[i] (line 13), which results in the FM sketch OFM. Then, we can use the standard unbiased estimator to estimate $\hat{\rho}_i$ for node v_i (line 14-15). After finding the node with the maximal approximate marginal gain, we need to update the answer set S and the FM sketch CFM. Note that we only need to do a bitwise-OR between the FM sketches CFM and FM[Idx] to update the CFM (line 19-22). Here FM[Idx] denotes the FM sketch of the node v_{Idx} which achieves the maximal approximate marginal gain. Finally, Algorithm 1 outputs the answer set S and the approximate σ_k (line 23). Notice that to calculate the cumulative k-robustness $\bar{\sigma}_k$ we do not need to invoke Algorithm 1 k times, but invoke Algorithm 1 with parameter k only once. Because we can record all the F (line 22) obtained in each round and compute the cumulative k-robustness. Additionally, we can use the so-called CELF framework [15] to accelerate both the original greedy algorithm and our randomized greedy algorithm.

Algorithm 1 The Randomized Greedy Algorithm

```
Input:   Network $G = (V, E)$ and $k$.
Output:  A set $S$ with $k$ nodes, $\sigma_k$

1: Let $h : \{e_1, \cdots, e_m\} \to \{1, \cdots, l\}$ be the hash function that
   maps the edges to a position of the BITMAP, here $l = \log m +$
   $c$ denotes the size of the BITMAP ;
2: for each node $v_i \in V$ do
3:    Initialize a BITMAP FM[i] $\leftarrow 0$;
4:    for each incident edge $e$ of $v_i$ do
5:       Set the $h(e)$-bit of FM[i] to 1;
6: $S \leftarrow \emptyset$;
7: Create two FM sketches CFM $\leftarrow 0$, OFM $\leftarrow 0$;
8: $F \leftarrow 0$;
9: for iter = 1 to $k$ do
10:    max $\leftarrow -1$;
11:    $Idx \leftarrow 0$;
12:    for each node $v_i \in (V \backslash S)$ do
13:       OFM $\leftarrow$ (CFM) bitwise-OR (FM[i]);
14:       Let $z$ be the position of the least-significant 0 bit in
          OFM;
15:       $\hat{\rho}_i \leftarrow 2^z/0.77351 - F$;
16:       if $\hat{\rho}_i >$ max then
17:          max $\leftarrow \hat{\rho}_i$;
18:          $Idx \leftarrow i$;
19:    $S \leftarrow S \cup \{v_{Idx}\}$;
20:    CFM $\leftarrow$ (CFM) bitwise-OR (FM[$Idx$]);
21:    Let $z$ be the position of the least-significant 0 bit in CFM;
22:    $F \leftarrow 2^z/0.77351$;
23: return $S$ and $1 - F/m$;
```

Theoretically, by a similar analysis as in [11], Algorithm 1 can achieve a $1-1/e-\epsilon$ approximate ratio with high probability for computing the σ_k on general networks. The reason is because the FM sketch estimates the marginal gain $\rho_i(S)$ of any set S within an ϵ error bound with high probability [10]. The time complexity of Algorithm 1 is $O(kn + m)$. First, Algorithm 1 takes $O(m)$ time to initialize the FM sketches for every node (line 2-5). Second, Algorithm 1 uses $O(kn)$ time to compute the σ_k. The rationale is that the bitwise-OR (line 13) and the estimation step (line 14-15) can be done in constant time [18]. We emphasize that $O(kn+m)$ is more efficient than $O(km)$ when k cannot be ignored. Another advantage of Algorithm 1 is that the FM sketches for every node can be built offline. Assume that we have built the FM sketches for every node of a given network G. Then, for any given k, Algorithm 1 can compute the corresponding σ_k in $O(kn)$ time. However, the original greedy algorithm still needs $O(km)$ time complexity for computing σ_k. For the space complexity, Algorithm 1 needs to store the network G which takes $O(m + n)$ space complexity. In addition, Algorithm 1 maintains $O(n)$ FM sketches which take $O(n \log m)$ bits, because each FM sketch only takes $O(\log m)$ bits. The

Table 1: Summary of the datasets

Name	#nodes	#edges	Ref.	Description
GrQc	5242	28968	[14]	Co-authorship networks
Astroph	18772	396100	[14]	
HepTh	9877	51946	[14]	
HepPh	12008	236978	[14]	
CondMat	23133	186878	[14]	
DBLP	78649	382294	website	
Delicious	537392	1459778	[22]	Online social networks
Douban	154908	654324	[22]	
Epinions	75872	396026	[14]	
Slashdot1	77360	826544	[14]	
Slashdot2	82168	867372	[14]	
Brightkite	58228	428156	[14]	Location based social networks
Gowalla	196591	1900654	[14]	
EmailEnron	36692	367662	[14]	Communication networks
EmailEuAll	265182	224372	[14]	
Gnutella04	10876	36308	[14]	P2P networks
Gnutella05	8846	27572	[14]	
Gnutella06	8717	27790	[14]	
Gnutella08	6301	18284	[14]	
NotreDame	325729	1522178	[14]	Web

size of $O(n \log m)$ bits can be dominated by the $O(m + n)$ graph size. So putting it all together, the space complexity of Algorithm 1 is $O(n+m)$, which is the same as the original greedy algorithm.

4. EXPERIMENTS

In this section, we conduct extensive experiments on 20 real datasets to evaluate the effectiveness and efficiency of our approaches. In the following, we first describe our experimental setup and then report our findings.

4.1 Experimental setup

Datasets: The network datasets used in our experiments are given in Table 1. These networks can be classified into six categories. (1) The co-authorship networks: we collect 5 physics co-authorship networks which are GrQc, Astroph, HepTh, HepPh, and CondMat from Stanford network data collections [14]. These 5 physics co-authorship networks represent the co-authorship over 5 different areas in physics respectively. DBLP (`http://www.informatik.uni-trier.de/~ley/db/`) is a computer science bibliographic dataset. We built a co-authorship graph from a subset of the DBLP data with 78,649 authors. (2) Online social networks (OSNs): we download the Delicious (`http://delicious.com/`) and Douban (`http://www.douban.com/`) from ASU social computing data repository [22] and download the Epinions (`http://www.epinions.com`) and two Slashdot datasets (`http://slashdot.org/`) from Stanford network data collections [14]. (3) Location-based social networks (LBSNs): the Brightkite and Gowalla are two notable LBSNs. We download these two datasets from Stanford network data collections [14]. (4) Communication networks: we download two Email communication networks (EmailEnron and EmailEuAll) from Stanford network data collections [14]. (5) P2P networks: we employ four P2P networks (Gnutella04, Gnutella05, Gnutella06, and Gnutella08) which are originally collected from Gnutella [14]. (6) Web graphs: we download a web graph dataset from Stanford network data collections [14], which is originally collected from University of Notre Dame.

Parameter settings and experimental environment: There are two parameters in Algorithm 1: the number of hash functions and the size of the bitmap. In all of our experiments, we set the number of hash functions and the size of the bitmap to be 100 and 30, respectively. We conduct our experiments on a Windows Server 2007 with 4xDual-Core Intel Xeon 2.66 GHz CPU, and 128G memory. All the algorithms are implemented by Visual C++ 6.0.

4.2 Experimental results

Here we report our experimental results on 20 general network datasets. For the directed networks, we consider them as the undirected networks by ignoring the direction of the edges. We use our k-robustness and cumulative k-robustness as two metrics. Notice that the budget of an attacker, i.e., k, is typically very small in practice. Hence, we mainly focus on measuring the robustness of a network under a small budget k. Table 2 reports our results when $k = 0.1\% \cdot n$ and $0.2\% \cdot n$, where $n = |V|$. In the following, we concentrate on analyzing the result on $k = 0.1\% \cdot n$ and similar results can be obtained when $k = 0.2\% \cdot n$.

As can be seen in Table 2, the P2P networks are more robust than other types of networks. For example, in the Gnutella05 dataset, after removing 0.1% of nodes, the k-robustness and cumulative k-robustness by the Greedy algorithm are 0.9838 and 0.9898, respectively. That is to say, there are only 1.62% of edges being deleted after removing 0.1% of nodes in the worst case. This observation indicates that removing a small fraction of peers from the P2P network does not significantly affect the number of links between the peers. Similarly, the co-authorship networks (first 6 rows in Table 2) are shown to be very robust. For instance, in the DBLP network, the k-robustness and cumulative k-robustness by the Greedy algorithm are 0.9618 and 0.9783 after removing 0.1% of nodes, respectively. These results suggest that a small number of "important researchers" leaving the research community will not significantly affect the co-authorship between the scholars. In general, the online social networks (rows 7-11 in Table 2) and the location based social networks (rows 12-13) show poor robustness. Taking the Gowalla dataset as an example, the k-robustness and the cumulative k-robustness by the Greedy algorithm are 0.8329 and 0.8788 when $k = 0.1\% \cdot n$, respectively. In other words, after removing 0.1% of nodes, 16.7% of social ties in the Gowalla network will be deleted. Also, the robustness of the Email communication networks, especially the EmailEuAll network, is very poor. In the EmailEuAll network, the k-robustness and cumulative k-robustness (for $k = 0.1\% \cdot n$) by the Greedy algorithm are 0.4404 and 0.6408, respectively. In other words, after deleing 0.1% of nodes, the number of residual edges in the EmailEuAll network are only 44.04% of the original edges. This observation suggests that the Email communication networks may be very vulnerable under the MVC attack. In addition, we find that the NotreDame web graph is not very robust w.r.t. the MVC attack, as the k-robustness and cumulative k-robustness (for $k = 0.1\% \cdot n$) by the Greedy algorithm are 0.8337 and 0.8835, respectively. This result is consistent with the previous results on the "robust yet fragile" nature of the Internet [8], which means that the Internet is robust to random errors but it is vulnerable w.r.t. the intended node attacks. Over all the datasets, we find that the k-robustness and cumulative k-robustness by the RGreedy algorithm are very close to the k-robustness and

Table 2: Robustness of general networks

Datasets	$k = 0.1\% \cdot n$				$k = 0.2\% \cdot n$			
	k-robustness		Cumulative k-robustness		k-robustness		Cumulative k-robustness	
	Greedy	RGreedy	Greedy	RGreedy	Greedy	RGreedy	Greedy	RGreedy
GrQc	0.9743	0.9747	0.9841	0.9841	0.9536	0.9551	0.9729	0.9732
Astroph	0.9660	0.9675	0.9807	0.9811	0.9430	0.9449	0.9670	0.9680
HepTh	0.9791	0.9794	0.9879	0.9881	0.9620	0.9635	0.9787	0.9791
HepPh	0.9554	0.9561	0.9751	0.9754	0.9176	0.9203	0.9547	0.9559
CondMat	0.9641	0.9652	0.9785	0.9787	0.9414	0.9442	0.9652	0.9662
DBLP	0.9618	0.9643	0.9783	0.9795	0.9369	0.9427	0.9636	0.9664
Delicious	0.7390	0.7842	0.8220	0.8422	0.6512	0.7011	0.7567	0.8064
Douban	0.9345	0.9403	0.9625	0.9651	0.8924	0.9021	0.9375	0.9433
Epinions	0.8515	0.8589	0.9056	0.9088	0.7789	0.7948	0.8592	0.8667
Slashdot1	0.8825	0.8903	0.9246	0.9286	0.8223	0.8389	0.8873	0.8949
Slashdot2	0.8800	0.8862	0.9230	0.9259	0.8203	0.8336	0.8856	0.8919
Brightkite	0.8982	0.9025	0.9353	0.9368	0.8479	0.8568	0.9034	0.9074
Gowalla	0.8329	0.8401	0.8788	0.8863	0.7835	0.8002	0.8426	0.8566
EmailEnron	0.8474	0.8529	0.9061	0.9083	0.7741	0.7836	0.8575	0.8621
EmailEuAll	0.4404	0.4809	0.6408	0.6711	0.2755	0.3078	0.4924	0.5272
Gnutella04	0.9826	0.9829	0.9888	0.9889	0.9720	0.9731	0.9827	0.9831
Gnutella05	0.9838	0.9839	0.9898	0.9898	0.9733	0.9743	0.9838	0.9841
Gnutella06	0.9829	0.9833	0.9893	0.9894	0.9739	0.9751	0.9839	0.9843
Gnutella08	0.9835	0.9835	0.9894	0.9894	0.9701	0.9706	0.9820	0.9821
NotreDame	0.8337	0.8492	0.8835	0.8920	0.7761	0.8055	0.8437	0.8574

cumulative k-robustness by the Greedy algorithm, respectively. More specifically, for the k-robustness and cumulative k-robustness when $k = 0.1\% \cdot n$, the maximal absolute differences between the RGreedy algorithm and the Greedy algorithm are only 0.0452 (appearing in the Delicious dataset) and 0.0303 (appearing in the EmailEuAll dataset) over all the datasets, respectively. When $k = 0.2\% \cdot n$, the maximal absolute differences for the k-robustness and cumulative k-robustness are 0.0499 and 0.0497 (both appearing in the Delicious dataset), respectively. These results imply that our RGreedy algorithm is as effective as the Greedy algorithm. The detailed performance analysis of the RGreedy algorithm is deferred to the full version of this paper.

Acknowledgments

The work was supported by grant of the Research Grants Council of the Hong Kong SAR, China No. 419109 and 411211.

5. REFERENCES

[1] R. Albert, H. Jeong, and A.-L. BarabÍćsi. Error and attack tolerance of complex networks. *Nature*, 406, 2000.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 1999.

[3] D. Bauer, F. Boesch, C. Suffel, and R. Tindell. Connectivity extremal problems and the design of reliable probabilistic networks. *in Theory and application of graphs*, 1981.

[4] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25), 2000.

[5] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85(21), 2000.

[6] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Breakdown of the internet under intentional attack. *Physical Review Letters*, 86(16), 2001.

[7] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. Worst-case and probabilistic analysis of algorithms for a location problem. *Operations Research*, 28(4):847–858, 1980.

[8] J. C. Doyle, D. L. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger. The ąŕrobust yet fragileąś nature of the internet. *PNAS*, 102(41):14497–14502, 2005.

[9] M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. J.*, 23(98):298–305, 1973.

[10] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.

[11] P. R. Goundan and A. S. Schulz. Revisiting the greedy approach to submodular set function maximization. *Technical report, MIT*, 2008.

[12] Q. Han, Y. Ye, H. Zhang, and J. Zhang. On approximation of max-vertex-cover. *European Journal of Operational Research*, 143(2):342–355, 2002.

[13] F. Harary. Conditional connectivity. *Networks*, 13:346–357, 1983.

[14] J. Leskovec. Standford network analysis project. 2010.

[15] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[16] B. Mohar. Isoperimetric number of graphs. *J. Combin. Theory, Ser. B*, 47(3):274–291, 1989.

[17] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 14:265–294, 1978.

[18] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. Anf: a fast and scalable tool for data mining in massive graphs. In *KDD*, pages 81–90, 2002.

[19] C. M. Schneider, A. A. Moreira, J. Joslę́ S. Andrade, S. Havlin, and H. J. Herrmann. Mitigation of malicious attacks on networks. *PNAS*, 108(10):427–486, 2011.

[20] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *nature*, 1998.

[21] J. Wu, M. Barahon, Y.-J. Tan, and H.-Z. Deng. Spectral measure of structural robustness in complex networks. *IEEE Transactions on Systems, man and cybernetics-part A*, 41(6), 2011.

[22] R. Zafarani and H. Liu. Social computing data repository at ASU, 2009.