

# Containing Viral Spread on Sparse Random Graphs: Bounds, Algorithms, and Experiments

Milan Bradonjić, Michael Molloy, and Guanhua Yan

**Abstract.** Viral spread on large graphs has many real-life applications such as malware propagation in computer networks and rumor (or misinformation) spread in Twitter-like online social networks. Although viral spread on large graphs has been intensively analyzed on classical models such as Susceptible–Infectious–Recovered, there still exists a deficit of effective methods in practice to contain epidemic spread once it passes a critical threshold. Against this backdrop, we explore methods of containing viral spread in large networks with the focus on sparse random networks. The viral containment strategy is to partition a large network into small components and then to ensure that all messages delivered across different components are free of infection. With such a defense mechanism in place, an epidemic spread starting from any node is limited to only those nodes belonging to the same component as the initial infection node. We establish both lower and upper bounds on the costs of inspecting intercomponent messages. We further propose heuristic-based approaches to partitioning large input graphs into small components. Finally, we study the performance of our proposed algorithms under different network topologies and different edge-weight models.

## I. Introduction

Although computer worms came into existence more than two decades ago, they remain a severe threat to Internet security. Modern computer malware has commonly applied social engineering tricks for their propagation, due to the fact that social trust among computer users has made them less vigilant against potential malware threats. For instance, the ILOVEYOU worm managed to infect tens of millions of Windows computers worldwide in 2000 through email attachments.<sup>1</sup> Moreover, the emerging popularity of online social network sites such as Facebook and Twitter has provided a new playground for computer malware, as evidenced by a few recently spotted worms, such as Koobface and W32/KutWormer,<sup>2</sup> that specifically targeted these networks. In contrast to traditional Internet worms such as Code Red [Zou et al. 02] and Slammer [Moore et al. 03], which use port scanning to discover vulnerable machines, computer worms based on social engineering pose an even more severe cyberthreat to many enterprise networks, since they can easily penetrate through enterprise firewalls or intrusion detection systems (IDSs).

Like malware propagation in computer networks, rumor or misinformation spread in social networks has a destructive nature. The growing popularity of social networking sites such as Twitter and Facebook has made them one of the major news sources for many people. Unfortunately, misinformation can also be spread on these social media networks, in some cases causing undesirable consequences, such as public panic. One such example is the spread of rumors regarding swine flu on Twitter in 2009.<sup>3</sup>

The challenge of containing viral spread on large graphs is common to both malware propagation in computer networks and misinformation spread in social networks. A folklore fact from epidemiology is that when the spread of an epidemic passes a critical threshold, or takeoff point, it may become a pandemic [Ben-Naim and Krapivsky 04]. One well-studied strategy in containing epidemic spread is *immunization* [Cohen et al. 03], which ensures that a set of nodes are immune to infection. Applying immunization schemes to fight against malware propagation in computer networks or misinformation spread in social networks, however, has its limitations. Due to the distributed nature of computer networks, computers are often administered by a variety of domains or organizations,

---

<sup>1</sup>See <http://news.cnet.com/2100-1001-240112.html>.

<sup>2</sup>See <http://www.kaspersky.com/news?id=207575670> and <http://www.statemaster.com/encyclopedia/Orkut>.

<sup>3</sup>See [http://neteffect.foreignpolicy.com/posts/2009/04/25/swine\\_flu\\_tweeters\\_power\\_to\\_misinform](http://neteffect.foreignpolicy.com/posts/2009/04/25/swine_flu_tweeters_power_to_misinform).

making it a daunting task to immunize a specific computer. Also, in the context of social networks, it may be difficult to convince a specific user to stop spreading misinformation.

Realizing the challenges of immunization in containing viral spread on large graphs, we instead focus on a different strategy, which is to partition a large graph into a number of small *islands* and then deploy message sanitization techniques to ensure that all messages that are delivered across islands are free of infection. Hence, when a node is infected and starts spreading viral information (either malicious messages for spreading computer malware or misinformation in social networks), epidemic spreading can take place only within the nodes in the same island as the initial infection point. From a practical standpoint, the strategy considered is applicable only to scenarios in which viral messages across islands can be inspected and stopped. For our problems of interest, social relationships exploited by computer malware or misinformation spread are usually maintained at centralized servers or social networking sites, such as Gmail, Twitter, and Facebook, where communication messages can be inspected for viral information.

To be effective in containing viral spread on a large social graph, the strategy under consideration must ensure that none of the islands after graph partitioning is too large. Note that the problem is different from the balanced graph partitioning problem [Andreev and Räcke 04], which aims at balancing the sizes of different components. Although an ideal approach is to sanitize every message in the network, so that every component contains exactly one node, the solution would be computationally prohibitive in reality due to the enormous number of communications to inspect in a large network like Twitter.<sup>4</sup> Hence, in a practical setting, it is crucial to strike a balance between efficiency and effectiveness.

Motivated by such a tradeoff, we study how to contain viral spread on large social graphs under limited operational resources. More specifically, we focus on the following problem. Consider an undirected graph  $G(V, E)$ , where the set of nodes  $V$  denotes the set of users, and the set of edges  $E$  contains all friendships among the users in the network. The weight  $w(u, v)$  of an edge  $(u, v)$  is the normalized number of messages sent between users  $u$  and  $v$  in the past.

The goal is to find a subset of edges  $E' \subseteq E$  that minimizes the overall cost  $B = \sum_{e \in E'} w(e)$ , given the constraint that the size of the largest connected component after removing all edges in  $E'$  from  $G$  must not be greater than a certain threshold. Social network graphs are typically sparse [Faloutsos et al. 04], so in this work we consider only sparse random graphs in which the numbers of nodes

---

<sup>4</sup>There are 50 000 000 messages transmitted within the Twitter network every day. See <http://blog.twitter.com/2010/02/measuring-tweets.html>.

and edges are of the same asymptotic order. Extending this work beyond sparse random graph models will be a future project for us.

In a nutshell, our main contributions are as follows. First, we consider minimizing the overall cost of disintegrating sparse Bernoulli (Erdős–Rényi) random graphs under constrained edge deletion such that the size of each connected component is no greater than a certain threshold. Under this random graph model, we provide both (i) a threshold on the number of edges to be deleted such that every connected component in the remaining graph has size at most a given constant, and (ii) a threshold on the number of edges to be deleted below which the remaining graph always has a connected component of size linear in the number of nodes.

We extend our results to a more generic type of random graph models, that is, random graphs with a given degree sequence. In particular, we provide lower and upper bounds on the cost of disintegrating sparse random graphs with a given degree distribution under different edge-weight distributions, including uniform edge weights, bounded edge weights, and unbounded edge weights with finite mean. These thresholds on the costs are related to the expected value and concentration of the maximum spanning forest of an input graph.

Finally, based on insights gained from our theoretical analysis, we propose heuristic-based algorithms to disintegrate sparse random graphs into small connected components. Our method first computes the maximum spanning forest of the original graph, and then uses one of two different heuristics to disintegrate the forest. We further generate synthetic graph topologies using sparse random graph models and study experimentally the performance of our proposed algorithm under different edge-weight models, including uniform, exponential, and power-law distributions. The experimental results confirm that our theoretical analysis guides us toward better heuristic-based approaches to containing viral spread on sparse random graphs.

Our results establish theoretical bounds on the performance of containing viral spread on large sparse random graphs based on graph partitioning and thus shed light on its limitation when it is deployed on real-world networks. Although put in the context of containing malware or misinformation spread in large social networks, the conclusions drawn from this work have independent interest in other application domains as well, since they essentially deal with a fundamentally theoretical problem of how to disintegrate networks under edge-deletion constraints.

The rest of the paper is organized as follows. Section 2 presents related work. In Section 3, we establish bounds on the cost for disintegrating sparse Erdős–Rényi random graphs. Section 3.3 provides bounds on the cost for disintegrating sparse random graphs with a given degree sequence, where edge weights are

either constant, bounded, or unbounded i.i.d. random variables. In Section 4, we provide a heuristic-based algorithm to partition sparse random graphs and study its performance on synthetically generated graph topologies.

## 2. Related Work

One motivating application behind this work is to contain propagation of malware based on online social contacts. Zou et al. developed a model that characterizes propagation of computer worms on email networks that typically follow a heavy-tailed distribution [Zou et al. 04]. They found that computer worms spread fast in scale-free networks, but by selectively immunizing those highly connected nodes, it is possible to slow down malware spread significantly in such networks. As we shall discuss later, targeted immunization for containing spread of social-based malware has its limitations. Xu et al. proposed a correlation scheme to monitor a small set of nodes for detecting malware spread in online social networks, but their algorithm works only for malware detection rather than containment. Similarly, techniques proposed in [Yan et al. 08] focus on detection, instead of containment, of malware spread in IM (instant messaging) social networks. Using a dataset collected from a real-life online social network, Yan et al. analyzed its social graph and user activity patterns and found that both play a critical role in malware spreading in online social networks [Yan et al. 11]; they further tried a community structure detection algorithm to partition the social graph into small connection components, and found that a significant fraction of edges have to be removed in order to disintegrate the graph effectively. The study in [Yan et al. 11] was done empirically on a specific network topology, and thus does not have a strong theoretical foundation.

Another motivating application of this work is containment of rumor or misinformation diffusion in large social networks. Research on this topic is still in its infancy. Budak et al. considered the problem of limiting the spread of misinformation in social networks [Budak et al. 11]. Their approach was to convince a small set of users in the online social network to spread “good” rumors that cancel out the influence of “bad” ones. Convincing people to spread “good” rumors in social networks, albeit an interesting idea, may not be feasible in practice. The strategy we consider in this work, however, does not require involvement of individual users.

Percolation theory has established a critical threshold for wide-scale epidemic spreading and has been widely applied to studying epidemic spreading in diverse network structures, such as small-world networks [Newman and Watts 99], heterogeneous networks [Sander et al. 02], and sensor networks [Sun et al. 09]. In the context of scale-free networks, selectively immunizing those highly connected

nodes is an effective approach to slowing down epidemic spread in such networks [Zou et al. 04, Pastor-Satorras and Vespignani 02].

In problems of containing social-based malware and misinformation spread in social networks, however, the key challenge that faces node immunization is the difficulty of interacting with individual users due to the distributed nature of social networks. An alternative approach would be to achieve node immunization by sanitizing all messages that come to or from those nodes to be immunized. In our problem, however, this may not be the most cost-effective approach, because highly connected nodes could generate a large number of communication messages.

### 3. Bounds on the Cost of Disintegrating Erdős–Rényi Random Graphs

To tackle the problem of disintegrating sparse random graphs effectively, we first consider the simple Erdős–Rényi random graphs  $G_{n,M}$  [Erdős and Rényi 59]. We are interested in establishing theoretical bounds on the cost  $B$  that is necessary to disintegrate a sparse Erdős–Rényi random graph. In this work, we will use the abbreviation a.a.s. for “asymptotically almost surely” to denote a probability tending to one as the number of nodes tends to infinity.

We fix  $c > 1/2$  and consider a random graph  $G$  from  $G_{n,M=cn}$ . Let  $L = L(c)$  be the unique positive solution to  $L = 1 - e^{-2cL}$ , and let

$$R = R(c) = cL(1 + e^{-2cL}).$$

The giant component of  $G$  has  $Ln + o(n)$  vertices [Erdős and Rényi 60] and  $Rn + o(n)$  edges. Then, we can establish the following theorem.

**Theorem 3.1.** *For any constants  $\gamma, c > 0$ , there exist constants  $\alpha, t > 0$  such that a.a.s.  $G_{n,M=cn}$  is such that:*

- (a) *If we remove fewer than  $(R - L - \gamma)n$  edges, then there must exist a component of size at least  $\alpha n$ .*
- (b) *We can remove  $(R - L + \gamma)n$  edges so that every component has size at most  $t$ .*

That is, the critical threshold occurs at  $(R - L)n$  edges. In Section 3.3, we will describe how to extend this result to other random graph models, including random graphs with a fixed power-law degree sequence.

In order to prove Theorem 3.1, we first bound the number of edges that must be removed from a tree with bounded degree so that each of the resulting components has size  $\leq t$  for some given positive  $t$ .

**Lemma 3.2.** *Let  $H$  be a tree with maximum degree at most  $d$ . For any  $\beta \in (0, 1)$ , we can remove at most  $\beta|H|$  edges from  $H$  so that each resulting tree has size less than or equal to  $t = t(\beta, d) = \max\{4d, r\}$ , where  $r$  is the largest root of  $x - (3.5/\beta)\log x = 0$ .*

**Proof.** Denote the number of nodes in the tree by  $n = |H|$ . If  $n \leq t$ , then no edge needs to be removed. Hence, consider the case  $n \geq t + 1$ . We prove the assertion by induction on the size of a given tree  $H$ .

Suppose that the assertion is true for every tree of size  $\leq k$ , where  $k \geq t$ . Pick a node  $v$  that is not a leaf of the tree, and denote its degree by  $d_v = \deg(v)$ . Since  $v$  is not a leaf,  $d_v \geq 2$ , and moreover,  $d_v \leq d$  by the conditions of the lemma. Denote by  $K_1, K_2, \dots, K_{d_v}$  the corresponding tree components incident to  $v$ . From  $|K_1| + \dots + |K_{d_v}| = n - 1 \geq t$ , by the pigeonhole principle, it follows that at least one of those components, denoted by  $H'$ , has size  $|H'| \geq t/d_v \geq t/d$ . Let  $e$  be the edge connecting  $v$  and  $H'$ . Consider the two newly obtained trees  $H'$  and  $H'' = V \setminus (H' \cup \{e\})$  of sizes  $a = |H'|$  and  $b = |H''|$ . Then  $a \geq t/d$  and  $b \geq 2$ , since  $v$  is not a leaf. Both of  $H', H''$  have maximum degree at most  $d$ , and so we can apply the inductive hypothesis to those components.

We now prove that the number of edges that must be removed in order to achieve the assertion of the lemma is at most  $\phi_t(k)$ , defined as follows. For a given integer  $t > 0$ , define  $\phi_t$  on  $\mathbb{N}_0 = \{0, 1, 2, \dots\}$  such that

$$\phi_t(k) = \begin{cases} \beta k - \alpha \log k, & \text{if } k > t, \\ 0, & \text{if } k \leq t, \end{cases}$$

where  $\beta$  is given in the statement of the lemma and  $\alpha$  is to be chosen later. Notice that  $\phi_t(k) \geq 0$  for every  $k$ , given that  $t$  is at least as large as the larger root of the equation  $\beta x - \alpha \log x = 0$ . This simple fact follows by examining the function  $\beta x - \alpha \log x$  on  $\mathbb{R}^+$ .

By the inductive hypothesis, the number of edges that must be removed, including  $e$ , is at most

$$\begin{aligned} 1 + \phi_t(a) + \phi_t(b) &= 1 + (\beta a - \alpha \log a) + (\beta b - \alpha \log b) \\ &= 1 + \beta(a + b) - \alpha \log ab = 1 + \beta n - \alpha \log ab. \end{aligned}$$

It suffices to show that

$$1 + \beta n - \alpha \log ab \leq \phi_t(n) = \beta n - \alpha \log(a + b),$$

which is equivalent to

$$\frac{1}{a} + \frac{1}{b} \leq e^{-1/\alpha}. \quad (3.1)$$

Recall that  $a \geq t/d$  and  $b \geq 2$ . With  $\alpha = 3.5$ , the conditions of the lemma yield (3.1):

$$\frac{1}{a} + \frac{1}{b} \leq \frac{d}{t} + \frac{1}{2} \leq \frac{d}{4d} + \frac{1}{2} = \frac{3}{4} < e^{-1/3.5} \approx 0.751477,$$

which concludes the proof.  $\square$

Now we can prove Theorem 3.1.

**Proof.** (a): Let  $G'$  be formed by removing edges from  $G = G_{n,M}$ . A standard lemma (see Lemma 6.1) implies that there exists  $\alpha = \alpha(\gamma, c) > 0$  such that every subgraph  $H \subset G$  with fewer than  $\alpha n$  vertices has fewer than  $(1 + \frac{1}{2}\gamma)|H|$  edges. Therefore, if every component of  $G'$  has size less than  $\alpha n$ , then the total number of edges remaining from the giant component of  $G$  must be less than the sum of  $(1 + \frac{1}{2}\gamma)|H|$  taken over every component  $H$  of  $G'$  such that  $H$  is a subgraph of the giant component of  $G$ . This sums to

$$(Ln + o(n)) \left(1 + \frac{1}{2}\gamma\right) < Ln + \frac{1}{2}\gamma n,$$

since  $L < 1$ . Therefore, we must have removed more than  $(R - L - \gamma)n$  edges. This is a contradiction, so there must be a component of size at least  $\alpha n$ .

(b): Choose any spanning tree of the giant component. Remove all edges not in that tree. Then every other component that is not a tree has a.a.s. exactly one cycle. For each such component, remove an edge from that cycle. We expect to remove  $O(1)$  such edges [Bollobás 01]. We now have a forest, and a.a.s. we have removed a total of  $Rn - Ln + o(n)$  edges, since the spanning tree of the giant component has  $Ln + o(n)$  vertices.

Standard results on the tail of the degree sequence of  $G_{n,p}$  (see, e.g., [Bollobás 01]) show that the total number of edges incident with vertices of degree at least  $d$  is at most  $\epsilon n$ , where  $\epsilon \rightarrow 0$  as  $d \rightarrow \infty$ . So we can choose  $d$  large enough in terms of  $\gamma, L, c$  that the total number of edges touching vertices of degree greater than  $d$  will be less than  $\frac{1}{2}\gamma n$ . Remove all such edges. What remains is a forest of maximum degree at most  $d$ . By Lemma 3.2 with  $\beta = \gamma/4$ , we can remove  $\frac{\gamma}{2}|H|$  edges from each component  $H$  of that forest, so that each resulting tree has size at most  $t = t(d, \gamma) = O(1)$ . The total number of edges removed in this step is at most

$$\frac{\gamma}{2} \sum |H| < \frac{\gamma}{2} n.$$

So in all, we have removed at most  $(R - L + \gamma)n$  edges.  $\square$



### 3.1. Bounded Edge Weights

Now we extend Theorem 3.1 to the setting in which the edges have weights. We begin with the case that the weights are bounded. Let  $W$  be the maximum edge weight. In this case, we allow an adversary to first examine the random graph, and then add edge weights.

Let  $L'n$  be the total weight of a maximum-weight spanning tree of the giant component. Let  $R'n$  be the total weight of the edges in the giant component.

**Theorem 3.3.** *For any constants  $\gamma, c > 0$ , there exist constants  $\alpha, t > 0$  such that a.s.  $G_{n,M=cn}$  is such that if an adversary places positive weights of up to  $W$  on each edge, then:*

- (a) *If we remove edges of total weight less than  $(R' - L' - \gamma)n$ , then there must be a component of size at least  $\alpha n$ .*
- (b) *We can remove edges of total weight at most  $(R' - L' + \gamma)n$  so that every component has size at most  $t$ .*

**Proof.** (a): Remove any set of edges. From Lemma 6.1, it follows that for every  $\gamma, W > 0$ , there exists  $\alpha > 0$  such that every subgraph  $H \subset G$  with fewer than  $\alpha n$  vertices has at most

$$\left(1 + \frac{\gamma}{4W}\right) |H|$$

edges. Therefore, if every component has size less than  $\alpha n$ , then each such component  $H$  consists of a maximum spanning tree of  $H$  plus at most

$$\frac{\gamma}{4W} |H| + 1$$

additional edges (since the maximum spanning tree has  $|H| - 1$  edges).

If  $|H| > 12W/\gamma$ , then

$$\frac{\gamma}{4W} |H| + 1 < \frac{\gamma}{3W} |H|.$$

The expected number of cycles of length at most  $12W/\gamma$  in  $G_{n,M=cn}$  is  $O(1)$ , for constant  $c$ . Therefore, a.s. there are fewer than  $\sqrt{n}$  components  $H$  with  $|H| \leq 12W/\gamma$  containing any additional edges besides the maximum spanning tree.

Therefore, the total weight of the remaining edges that are in the giant component of  $G$  must be at most  $W \times \sqrt{n}$  plus the sum over every component  $H_i$  that is a subgraph of that giant component of the weight of a maximum spanning tree of  $H_i$  plus  $W \times \frac{\gamma}{3W} |H_i|$ . The total weight of these maximum

spanning trees is at most  $L'n$ , the weight of a maximum spanning tree of the giant component. To see this, note that we can form a spanning tree of the giant component by adding edges to join together maximum spanning trees of all these  $H$ 's. So the total weight of all remaining giant component edges is at most

$$W\sqrt{n} + L'n + \frac{\gamma}{3} \sum |H_i| \leq L'n + \frac{\gamma}{2}n.$$

(b): We proceed as in the proof of Theorem 3.1(b). This time, we choose a maximal spanning tree of the giant component. The total weight of the giant component's edges not in that tree is  $(R' - L')n$ . We choose  $d$  such that the total number of edges touching vertices of degree greater than  $d$  is at most  $\frac{1}{4W}\gamma n$ . We remove at most  $\frac{\gamma}{4W}|H|$  edges from each component  $H$  of that forest, so that each resulting tree has size at most  $t = t(d, \gamma, W) = O(1)$ . Since each edge has weight at most  $W$ , the total weight of the edges removed is less than  $(R' - L' + \gamma/2)n$ .  $\square$

### 3.2. Unbounded Edge Weights

We now consider random graphs with unbounded edge weights. This time, we do not permit an adversary to weight the graph; instead, the weights of the edges are chosen at random. More specifically, the weights are i.i.d. random variables, chosen from any probability distribution  $f(w)$  that has a sufficiently small tail  $\mathbb{P}(W > n) = o(n^{-2})$ .

Let  $\mu = \int_0^\infty wf(w)dw$  denote the expected weight and  $F(x) = \int_0^x f(w)dw$  the cumulative distribution function of an edge. Note that  $\mathbb{P}(W > n) = o(n^{-2})$  implies finite mean  $\mu < \infty$ .

Again, we let  $L'n$  be the total weight of a maximum-weight spanning tree of the giant component. Let  $R'n$  be the total weight of the edges in the giant component.

**Theorem 3.4.** *Consider any probability distribution  $f(w)$  such that  $\mathbb{P}(W > n) = o(n^{-2})$ . For any constants  $\gamma, c > 0$ , there exist constants  $\alpha, t > 0$  such that a.a.s.  $G_{n, M=c\gamma n}$  is such that:*

- (a) *If we remove edges of total weight less than  $(R - L - \gamma)n$ , then there must be a component of size at least  $\alpha n$ .*
- (b) *We can remove edges of total weight at most  $(R - L + \gamma)n$  so that every component has size at most  $t$ .*

**Proof.** We first argue that there exists  $W$  sufficiently large that a.a.s. the total weight of all edges of weight greater than  $W$  is at most  $\gamma/(4n)$ . Let us consider a random variable

$$\varphi(x) = \sum_{i=1}^M w_i \mathbb{I}\{w_i \geq x\}$$

that depends on  $x \geq 0$ . Moreover,

$$\mathbb{E}(\varphi(x)) = M \int_x^\infty w dF(w).$$

By Lebesgue's dominated convergence theorem (see [Billingsley 79]), given a finite mean  $\int_0^\infty w dF(w) < \infty$ , it follows that for any constants  $\gamma, c > 0$ , there exists  $W$  sufficiently large that

$$M^{-1} \varphi(W) = \int_W^\infty w dF(w) \leq \frac{\gamma}{5c}.$$

(This statement would not necessarily hold for infinite mean  $\int_0^\infty w dF(w) = \infty$ .) Therefore,

$$\mathbb{E}(\varphi(W)) \leq M \frac{\gamma}{5c} = \frac{\gamma n}{5}.$$

A random variable  $w_i \mathbb{I}\{w_i \geq W\}$  is absolutely integrable, since  $\mathbb{E}(w_i) < \infty$ . Hence, by the strong law of large numbers,  $n^{-1} \varphi(W)$  tends to  $n^{-1} \mathbb{E}(\varphi(W)) < \gamma/5$  a.a.s.

(a): We proceed as in the proof of Theorem 3.3(a). As in that proof, the total weight of the remaining giant component's edges of weight at most  $W$  is at most  $(L + \gamma/3)n$ . The total weight of the remaining edges of weight greater than  $W$  is at most  $\frac{\gamma}{4}n$ . So the total weight of the remaining giant component's edges is less than  $(L + \gamma)n$ , and so we must have removed weight at least  $(R - L - \gamma)n$ .

(b): We proceed as in the proof of Theorem 3.3(b). As in that proof, the removed edges of weight at most  $W$  total less than  $(R - L + \gamma/2)n$ . The removed edges of weight greater than  $W$  total at most  $\frac{\gamma}{4}n$ .  $\square$

We close this section by noting that for the model considered here, we can determine  $R', L'$ .

Determining  $R'$  is straightforward. The number of edges in the giant component is a.a.s.  $R = R(c)$ , as given at the beginning of this section. Since the edges are weighted independently, standard concentration arguments, as in the proof above, yield that a.a.s. the total weight of those edges is  $R'n + o(n)$ , where  $R' = R\mu$ .

The techniques of [Frieze 85] yield the expected size of the maximum spanning forest of the entire graph. First determine the weights of the edges, without

yet exposing their endpoints. Then sort these weights such that  $w_1 \geq \dots \geq w_n$ . For  $x > \frac{1}{2}$ , define  $g = g(x)$  to be the unique positive solution to  $g = 1 - e^{-2xg}$ . Thus, the giant component of  $G_{n,M=xn}$  a.a.s. has size  $g(x)n + o(n)$  [Erdős and Rényi 60]. Analyzing Kruskal's algorithm as in [Frieze 85] yields that the probability that the edge with weight  $w_i$  belongs to the maximum spanning forest is  $\pi(i/n) + o(1)$ , where

$$\pi(x) = \begin{cases} 1, & x \leq 1/2, \\ 1 - g(x)^2, & x > 1/2. \end{cases} \quad (3.2)$$

**Remark 3.5.** This is simply the probability, on selecting the endpoints of the edges in order of their weights, that the  $i$ th edge will have endpoints that are in different components of the graph formed by the first  $i - 1$  edges.

Thus, the expected total weight of the maximum spanning forest of the entire  $G_{n,M}$ , denoted by  $\text{MaxSF}(G_{n,M})$ , is given by

$$\mathbb{E}(\text{MaxSF}(G_{n,M})) = \sum_{i=1}^M \mathbb{E}(w_i) \pi\left(\frac{i}{n}\right) + o(n). \quad (3.3)$$

To obtain the weight of the maximum spanning tree of just the giant component, we need to subtract the total weight of the rest of the forest. The graph outside the giant component is a.a.s. a forest plus  $O(1)$  edges. The total weight of those  $O(1)$  edges will be  $o(n)$ , and so we can simply subtract the total weight of all edges outside the giant component. There are  $M - Rn$  such edges, and each has expected weight  $\mu$ . This yields

$$\mathbb{E}(L') = \sum_{i=1}^M \mathbb{E}(w_i) \pi\left(\frac{i}{n}\right) - (c - R)\mu n. \quad (3.4)$$

Straightforward concentration arguments (using, for example, Azuma's inequality plus an a.a.s. bound on the maximum edge weight) yield that  $L'$  is concentrated around its mean. We omit the details.

We now concentrate on (3.4). For the probability density function  $f(w)$  and the cumulative density function  $F(w)$ , the expected value  $\mathbb{E}(w_i)$  of the order statistics  $w_i$  is given by

$$\mathbb{E}(w_i) = M \binom{M-1}{i-1} \int_0^\infty w f(w) F(w)^{M-i} (1 - F(w))^{i-1} dw. \quad (3.5)$$

In general, (3.5) can be evaluated numerically. The following two cases in which weights are drawn from the uniform or exponential distribution demonstrate the possibility of expressing  $\mathbb{E}(w_i)$  analytically.

**Example 3.6. (Uniform weight distribution.)** The edge weights follow  $f(x) = 1$  on  $[0, 1]$ . Then  $\mu = 1/2$  and  $w_i$  follows the beta distribution  $B(M + 1 - i, i)$ . Hence,  $\mathbb{E}(w_i) = 1 - i/(M + 1)$ .

**Example 3.7. (Exponential weight distribution.)** The edge weights follow  $f(x) = \lambda e^{-\lambda x}$  for  $x \geq 0$ . Then  $\mu = 1/\lambda$  and

$$\mathbb{E}(w_i) = \lambda^{-1} M \binom{M-1}{i-1} \sum_{k=0}^{M-1} \binom{M-1}{k} \frac{(-1)^k}{(k+i)^2}.$$

### 3.3. Other Random Graph Models

One of the largest and most comprehensive recent studies on the structure of networks was done in [Leskovec et al. 09]. The results have been compared across more than one hundred large social and information networks, as well as small social networks, expanders, and networks with mesh-like or manifold-like geometry. That paper examined the number of nodes, the number of edges, the fraction of nodes in the largest biconnected component, the fraction of edges in the largest biconnected component, the average degree, the empirical second-order average degree, average clustering coefficient, the estimated diameter, and the estimated average path length. These examined networks contained roughly from 5000 to 14 000 000 nodes, and from 6000 to 100 000 000 edges. A key conclusion relevant to our work is that all of the studied networks were very sparse, with average degree  $\approx 2.5$  for the networks of blog posts, 400 for the network of movie ratings from Netflix, 10 (with median 6) for the social networks as well as most of the other studied networks; see [Leskovec et al. 09]. Those arguments lead us to study sparse random graphs with given degree distribution as a model for large networks, which is the main subject of our work. However, we are aware of imperfections of this model. As an example, a study on online social networks in [Kumar et al. 06] shows that more than half the nodes in online social networks do not belong to the largest component and mainly form stars.

Given the previous, we now list some properties that will suffice for our theorems to extend to other models of large real-world networks.

1. For any constants  $\epsilon, \gamma > 0$ , we can choose  $d, \alpha$  such that
  - (a) a.a.s. the sum of the degrees of all vertices of degree at least  $d$  is less than  $\epsilon n$ ;
  - (b) a.a.s. every subgraph  $S$  of size at most  $\alpha n$  contains at most  $(1 + \gamma)n$  *small edges*, where a small edge is one whose endpoints both have degree less than  $d$  in the original graph.

2. For any constant  $c_1$ , the expected number of cycles of length at most  $c_1$  is  $o(n)$ .

The proofs of Theorems 3.1, 3.3, 3.4 extend easily to any random graph with those properties. The only nontrivial modification is that in part (a), we must account for the at most  $\epsilon n$  remaining edges that are not small; by taking  $\epsilon < \frac{1}{2}\gamma$  (or  $\epsilon < \frac{\gamma}{10W}$ ), this number is negligible.

For example, random graphs with a power-law degree sequence (when the fraction of nodes having degree  $k$  is  $\Theta(k^{-\gamma})$  for some  $\gamma > 2$ ) are easily seen to satisfy these properties. Properties 1(a) and 2 are trivial; see Section 6 for a proof that if our degree sequence satisfies Property 1(a), then Property 1(b) also holds.

## 4. Algorithms and Experiments

In Section 3.2 we analyzed the expectation and concentration of the total weight of the maximum spanning forest. Based on the insights gained from our theoretical analysis, we propose a heuristic-based approach to disintegrating a sparse random graph, and then study its performance using synthetically generated graph topologies and edge weights.

### 4.1. Heuristic-Based Algorithm

We now propose a heuristic-based algorithm that disintegrates a sparse random graph  $G$  and demonstrate its experimental results. The main goal of the proposed Algorithm 1 is to break a forest of the input graph by using one of the two heuristics *susceptibility* and *edge betweenness centrality*. The steps of Algorithm 1 are as follows.

For an input graph  $G$ , find the maximum spanning forest  $F = \text{MaxSF}(G)$ . Assign  $B := \sum_{e \in G \setminus F} w(e)$  to be the initial value of the overall cost of disintegrating  $G$ .

Let  $F_1 := F$ , and let  $k$  be the number of edges in  $F$ . At every time step  $\tau = 1, \dots, k$ , find an edge  $e_\tau^*$  that maximizes  $\Upsilon(F_\tau, e)$  of the current graph  $F_\tau$ , defined by (4.1) or (4.2). If multiple edges have the highest score, pick one among them uniformly at random. Add the weight of  $w(e_\tau^*)$  to the overall cost  $B$ , remove  $e^*$  from the current graph  $F_{\tau+1} := F_\tau - e_\tau^*$ , and repeat the process.

We now define the score function  $\Upsilon(H, e)$  for the two heuristics: susceptibility and edge betweenness centrality.

**Susceptibility Heuristic.** For every  $e \in H$ , calculate the susceptibility [Grimmett 99] of the graph  $H - e$ . That is, let connected components  $C_1, C_2, \dots$  partition  $H - e$ .

**Algorithm 1.** Heuristic-Based Algorithm

- 
- 1: Find the maximum spanning forest  $F := \text{MaxSF}(G)$  of an input graph  $G$ .
  - 2: Assign  $B := \sum_{e \in G \setminus F} w(e)$  to be the initial value of the overall cost of disintegrating  $G$ .
  - 3: Let  $F_1 := F$ , and let  $k$  be the number of edges in  $F$ .
  - 4: **for**  $\tau \leftarrow 1, k$  **do**
  - 5:   Pick uniformly at random  $e_\tau^* \in \arg \max_{e \in F_\tau} \Upsilon(F_\tau, e)$  (apply (4.1) or (4.2), respectively).
  - 6:    $B := B + w(e_\tau^*)$ .
  - 7:    $F_{\tau+1} := F_\tau - e_\tau^*$ .
  - 8: **end for**
- 

Then define the score to be

$$\Upsilon_S(H, e) = \frac{1}{|V(H)|} \sum_i |C_i|^2. \quad (4.1)$$

The concept of susceptibility is imported from theoretical physics; see [Grimmett 99]. Moreover, notice that (4.1) equals the expected component size of the given graph. We stress that susceptibility in this work should not be confused with susceptibility in epidemic models [Kermack and McKendrick 33]. For example, susceptibility for a random graph with a given degree sequence is given in [Janson 10]. Note that at every step the susceptibility heuristic chooses an edge that breaks the current graph into a larger number of small components. The norm  $L_2$  has been used for purposes of our experiments, but any other norm  $L_p$  with  $p \geq 1$  could be used.

**Edge Betweenness Centrality Heuristic.** Betweenness centrality of an edge  $e$  represents the sum of the fraction of all pairs of shortest paths that traverse  $e$ ,

$$\Upsilon_{BC}(F_\tau, e) = \sum_{u, v \in F_\tau} \frac{\sigma(u, v \mid e)}{\sigma(u, v)}, \quad (4.2)$$

where  $\sigma(u, v)$  is the number of shortest paths between  $u$  and  $v$ , and  $\sigma(u, v \mid e)$  is the number of shortest paths between  $u$  and  $v$  that traverse  $e$ .

## 4.2. Experiments

To study the performance of the heuristic-based algorithm, we run it on synthetically generated sparse random graphs under different edge-weight models. In our experiments, we consider the following two random graph models:

- (a) Random graphs  $G_{50,100}$  with  $n = 50$  nodes and  $M = 100$  edges.
- (b) Random graphs on  $n = 50$  nodes with a given power-law degree distribution, Powerlaw(3, 1/4), that is,  $f(w) = w^{-3}/8$  for  $w \in [1/4, \infty)$ .

For the edge weights, we consider the following three distributions in our experiments:

- (i) Unif[0, 1], that is,  $f(w) = 1$  for  $w \in [0, 1]$ ,
- (ii) Exp(2), that is,  $f(w) = 2e^{-2w}$  for  $w \in [0, \infty)$ ,
- (iii) Powerlaw(3, 1/4), that is,  $f(w) = w^{-3}/8$  for  $w \in [1/4, \infty)$ .

The parameters of each edge-weight distribution are set in such a way that the average edge weights are the same for different distributions.

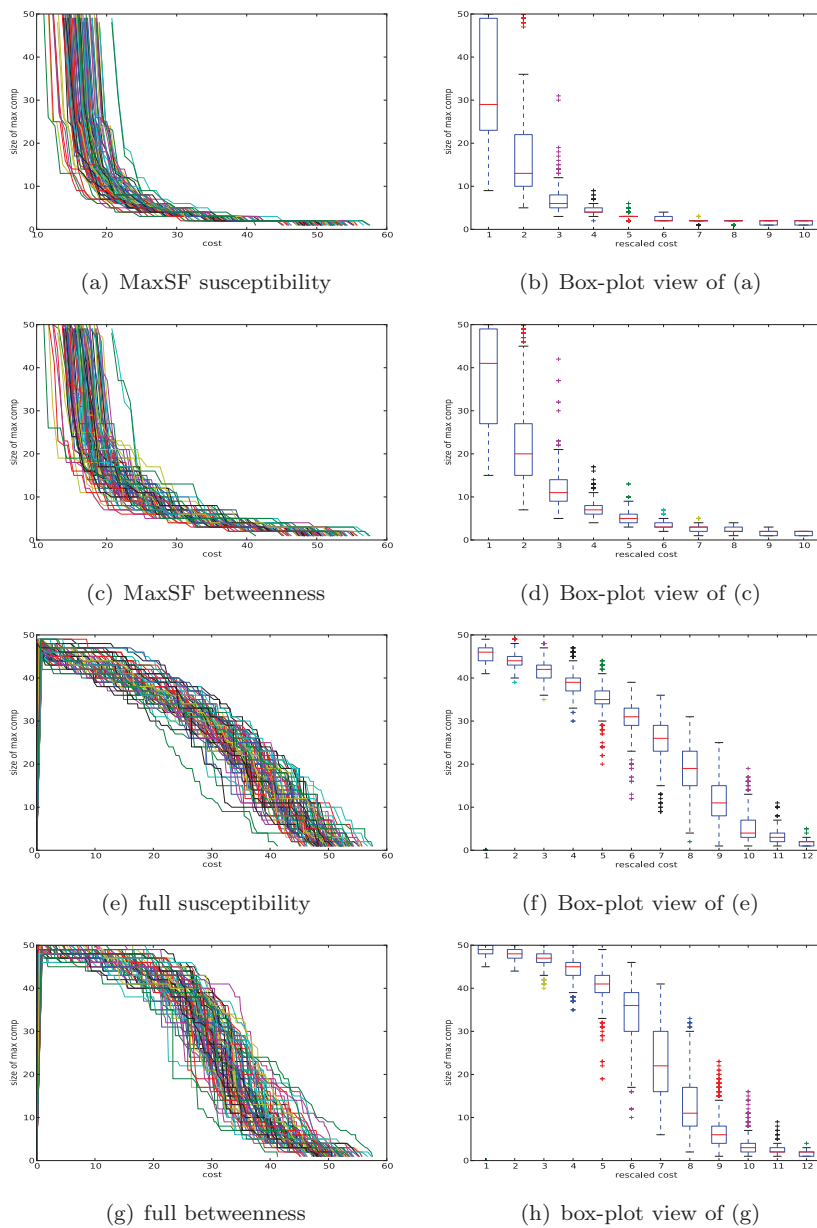
For each combination of graph topology and edge-weight distribution, we randomly generate 100 instances. For each instance, we use four different heuristic-based methods to compute the relationship between the size of the maximum component and the total cost of removed edges:

- (1) The algorithm described in Section 4.1 with greedy heuristics (MaxSF susceptibility).
- (2) The algorithm described in Section 4.1 with the betweenness-centrality heuristics (MaxSF betweenness).
- (3) In a different algorithm from the one described in Section 4.1, we use greedy heuristics as described in Section 4.1 directly on the entire input graph rather than on its maximum spanning forest (full susceptibility).
- (4) Similarly, we use the betweenness-centrality heuristics as described in Section 4.1 directly on the entire input graph rather than on its maximum spanning forest (full betweenness).

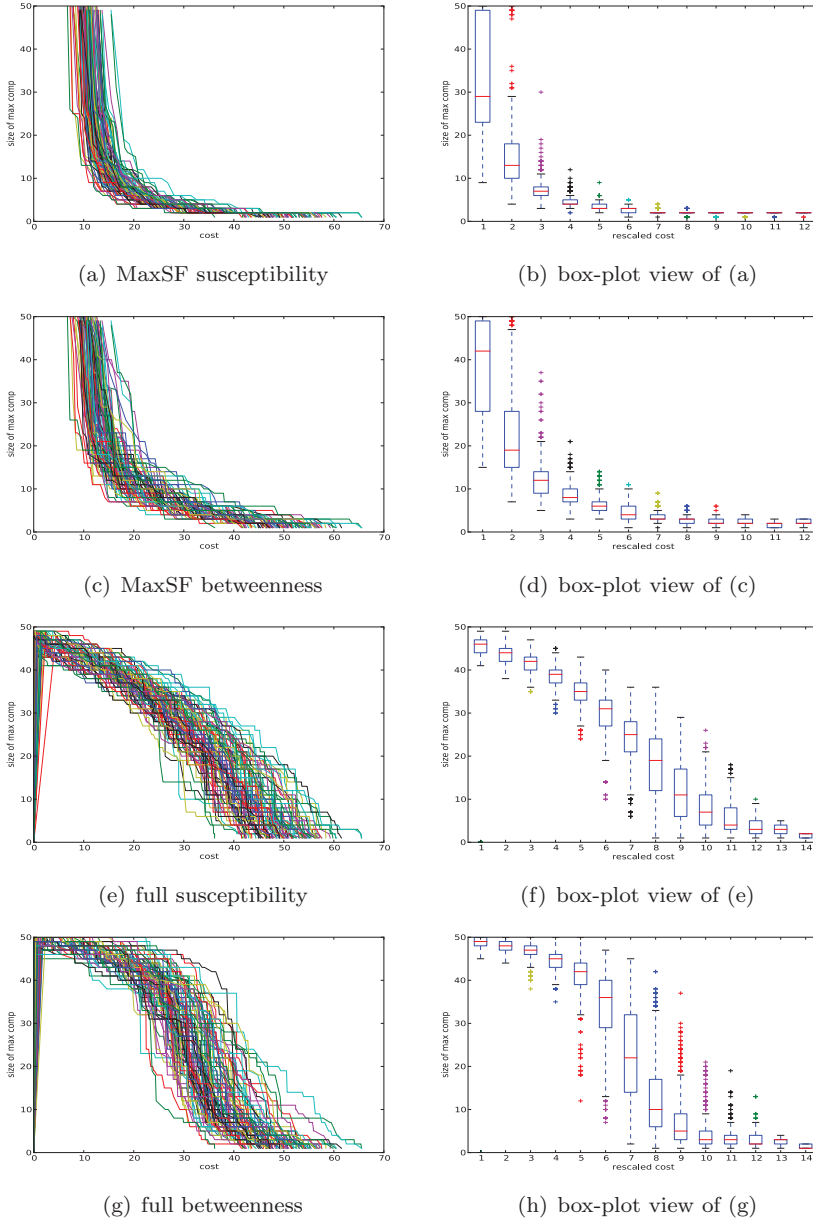
By comparing performance of the algorithms on maximum spanning forests against those directly on the entire topologies, we are interested in whether our theoretical analysis indeed guides us to find better heuristic-based approaches. Figures 1–6 provide evaluation results under different scenarios. In each of these figures, the left column shows the size of the maximum component against the total cost of removed edges, and the right column shows its box-plot view. The box-plot diagrams are standard box diagrams in statistics, representing the median and the 25th and 75th percentiles (the lower and upper quartiles). We heuristically chose the width of the boxes to be 5.

From the results, we make the following observations. For the two heuristics used, the susceptibility heuristic performs consistently better than the

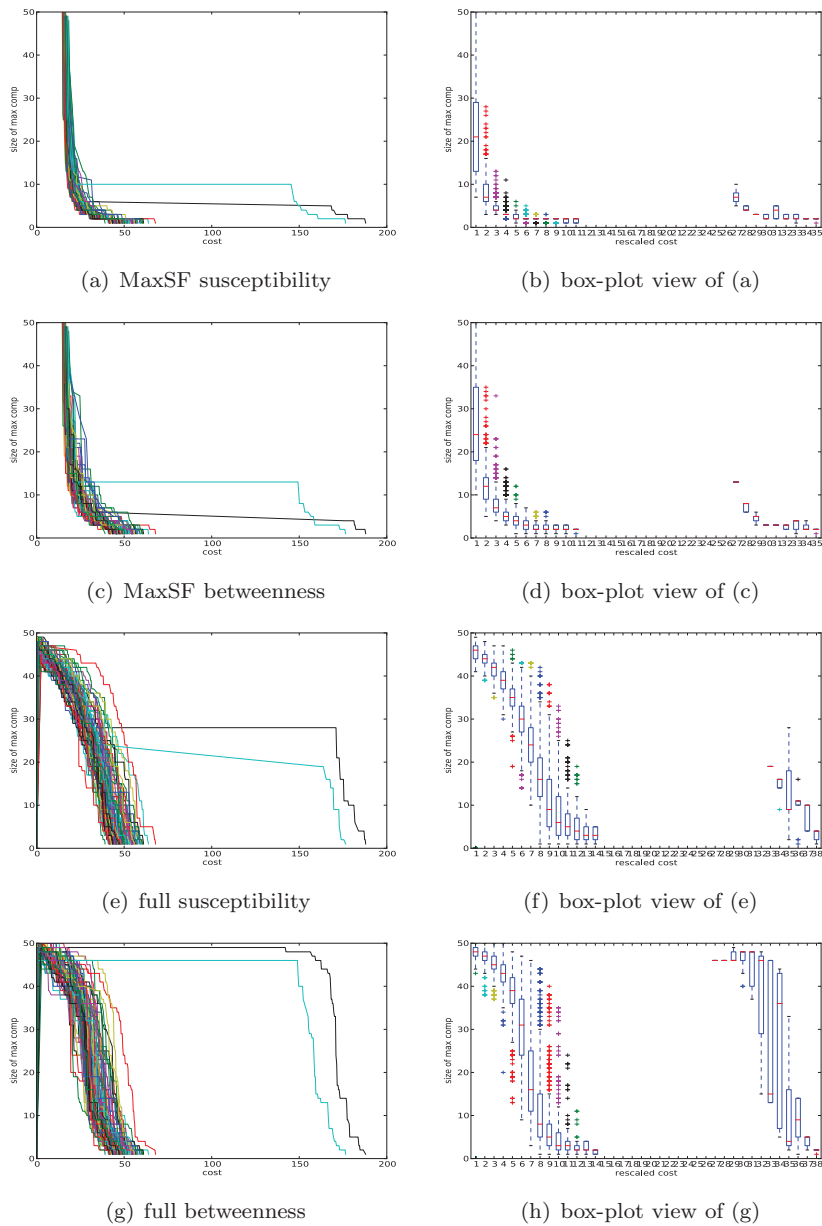




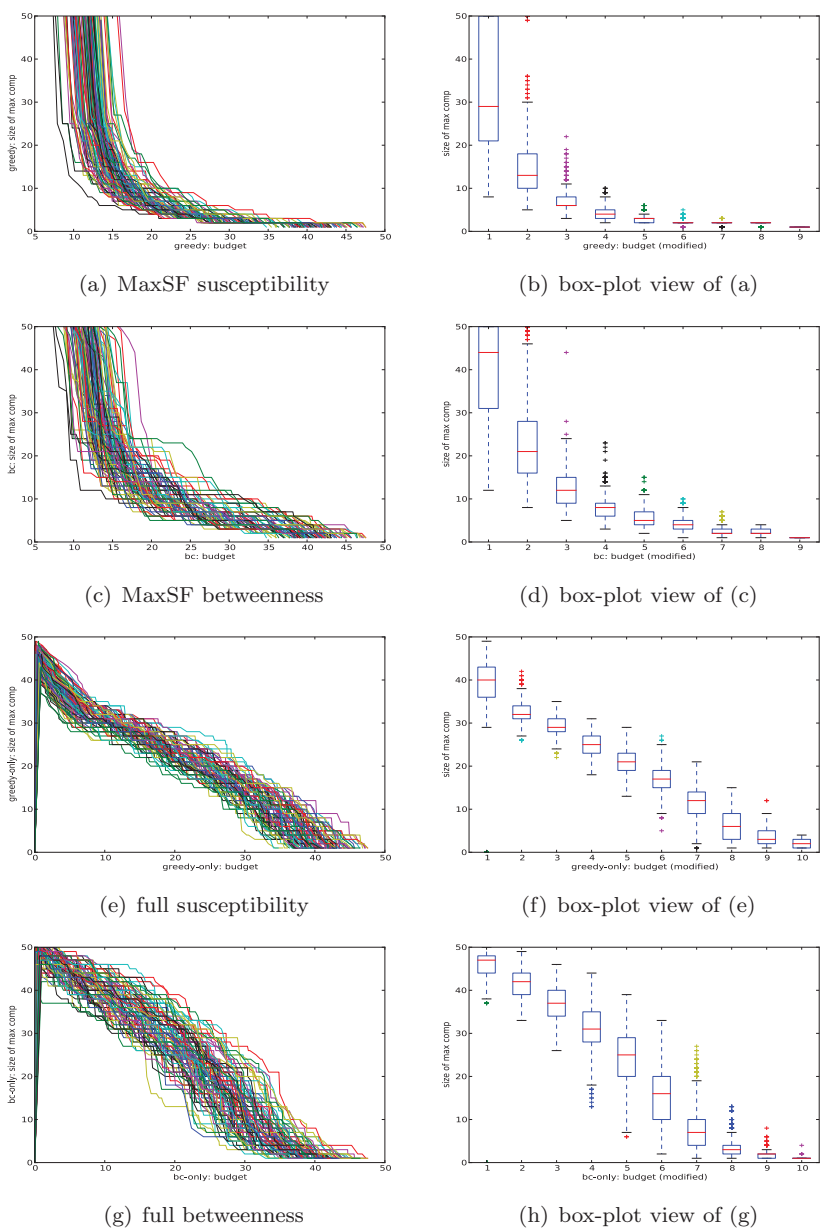
**Figure 1.** Size of maximum component vs. edge removing cost for  $G_{n,M}$  model with uniform edge weights (color figure available online).



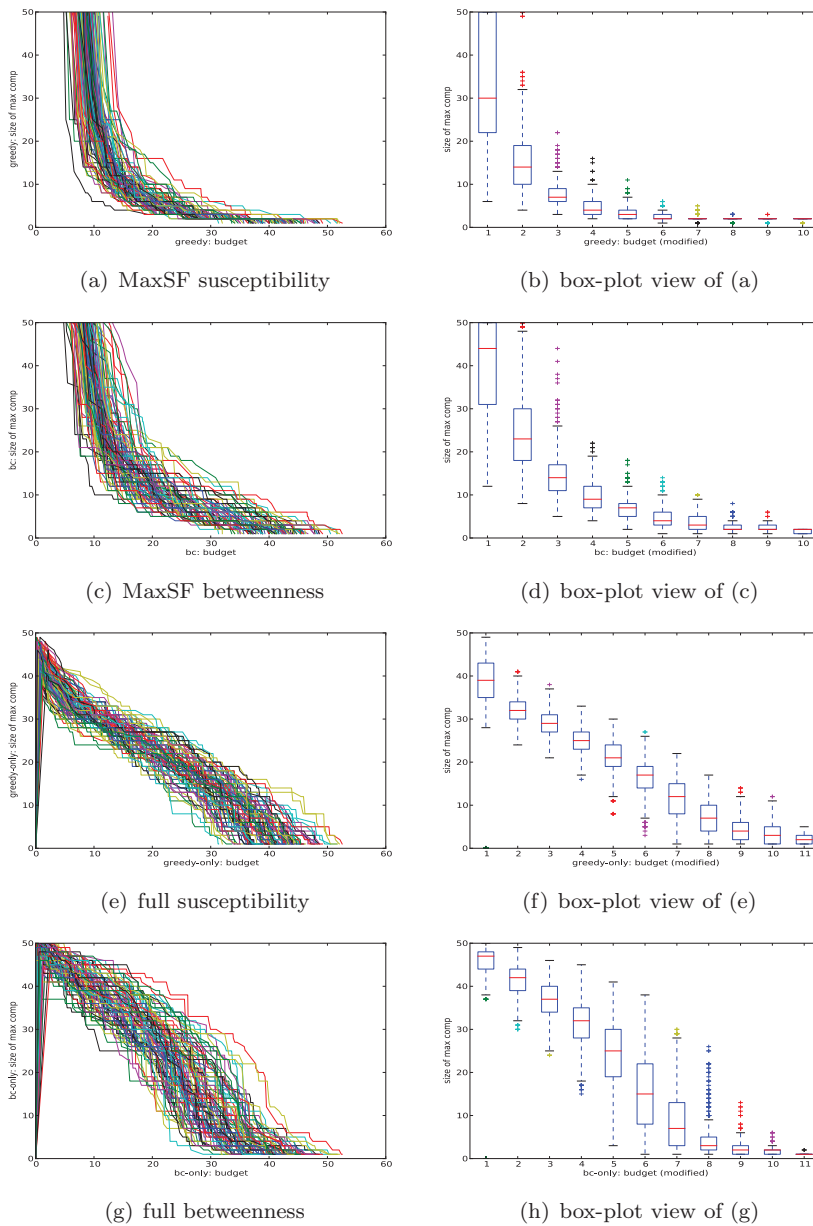
**Figure 2.** Size of maximum component vs. edge removing cost for  $G_{n,M}$  model with exponential edge weights (color figure available online).



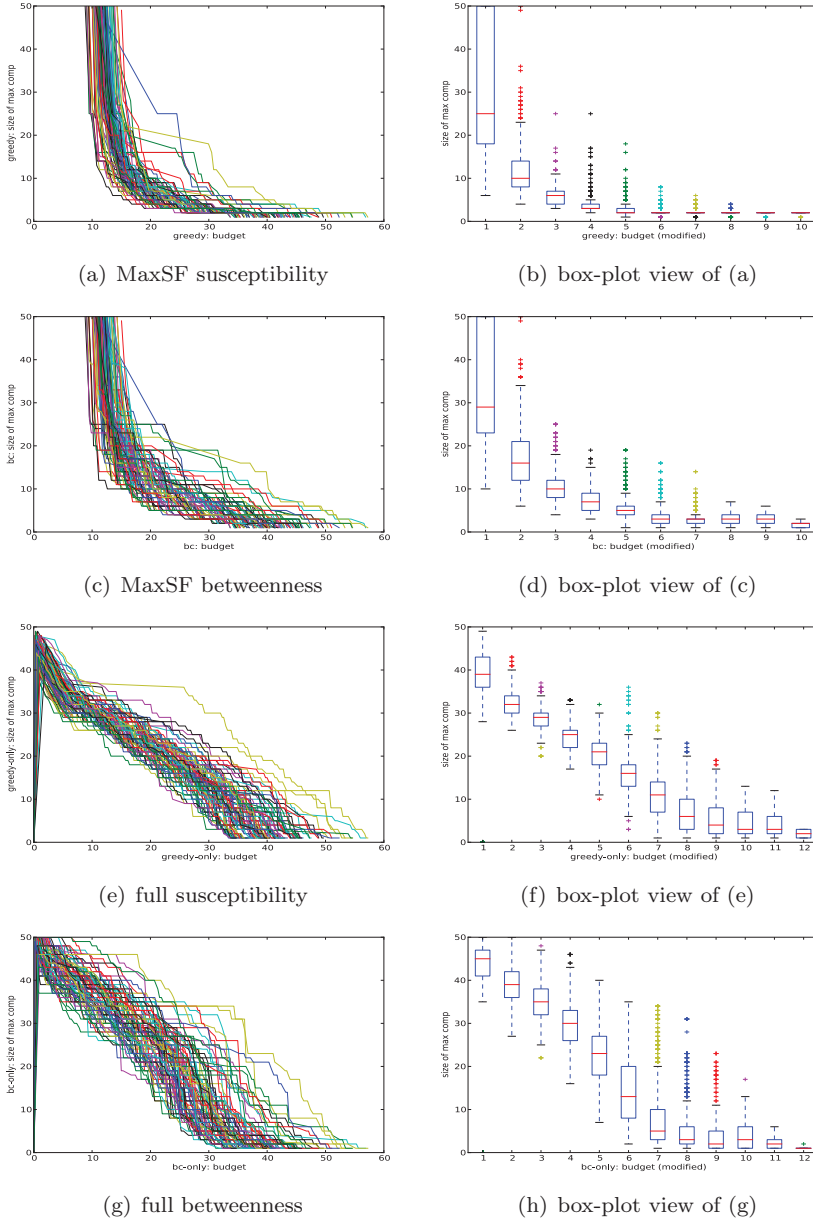
**Figure 3.** Size of maximum component vs. edge removing cost for  $G_{n,M}$  model with power-law edge weights (color figure available online).



**Figure 4.** Size of maximum component vs. edge removing cost for random graph model with power-law degree sequence, with uniform edge weights (color figure available online).



**Figure 5.** Size of maximum component vs. edge removing cost for random graph model with power-law degree sequence, with exponential edge weights (color figure available online).



**Figure 6.** Size of maximum component vs. edge removing cost for random graph model with power-law degree sequence, with power-law edge weights (color figure available online).

betweenness centrality heuristic in different combinations of network graphs and edge-weight distributions. We note, however, that the algorithm based on the susceptibility heuristic has a higher computational overhead, given its greedy nature. For every edge removed, the algorithm has to recompute the score of each edge in the remaining graph. For a large graph with billions of edges, this may not be a practical solution.

Comparing the results from MaxSF susceptibility and MaxSF betweenness against those from full susceptibility and full betweenness, we first observe that there are initial costs associated with the former to remove edges outside of the maximum spanning forests. Also, we find that the curves from the former drop off much more sharply. This suggests that for sparse random graphs, the strategy of removing edges outside the maximum spanning forest and then disintegrating the forest obtained really pays off, as long as the budget allows us to do that. The results confirm that our theoretical analysis indeed leads us to a better heuristic-based approach in disintegrating sparse random graphs.

In comparing results from different edge-weight distributions, we find that power-law distributions tend to have some samples with long tails. These phenomena occur when some expensive edges have to be removed in disintegrating the graphs. This is in accordance with the fact that power-law distributions are highly skewed, and thus more likely to produce heavy-weight edges than the other two. In some cases, when a heavy-weight edge bridges two connected components, this edge has a high betweenness centrality measure, and removing it can reduce the maximum component size significantly.

As evidenced from our experimental results, finding the maximum spanning forest plays an instrumental role in disintegrating sparse random graphs. Since our theoretical analysis concerns only sparse random graphs, the proposed algorithm may not perform well for other types of graphs. Pursuing efficient algorithms for those graphs remains as a project for future work.

## 5. Conclusions

This work is motivated by the challenges that arise in containing the spread of malware based on social engineering tricks and rumor propagation in large social networks. We consider a containment strategy that partitions the original graph into a number of small islands and sanitizes every message delivered across islands. We establish theoretical bounds on the cost necessary to disintegrate a sparse random graph such that none of the connected components has size greater than a given threshold. We also derive the expectation on the total weight of the

maximum spanning forest of the original graph, including its lower and upper bounds, as well as its concentration. Based on the insights gained from our theoretical analysis, we propose a heuristic-based approach to disintegrating a sparse random graph, and we study its performance with synthetically generated topologies. Our results from this work not only shed light on how to contain viral spread in real-life networks effectively, but also have independent interest in other application domains, since our work essentially deals with a fundamental theoretical problem.

We hope that the theory developed in this study will shed light on the capability, or the limitations, of a scheme that relies on graph partitioning to contain propagation of computer malware and misinformation in real-world social networks. Admittedly, the containment strategy considered in this study is not a panacea, and it works effectively only in the following scenarios. First, if we want to take a preventive approach against the viral spread of computer malware or misinformation in large-scale online social networks, we need to ensure that the system has sufficient computational resources to prevent such information from passing among different islands. Although the proposed strategy relieves us from monitoring all the messages in the network, there can still be a large number of messages that traverse different islands in a large online social network. Hence, filtering techniques that can efficiently detect interisland messages with patterns of interest (for example, those that embed URLs pointing to suspicious domain names, or those carrying sensitive words indicative of rumors) would be complementary to the containment strategy considered in this study.

In cases in which the accuracy of a message is difficult to verify in real time (for example, a claim that there is an epidemic disease in Alaska), the system can attach a caution flag along with the message before it is delivered to a different island. Such a caution flag at least raises awareness among users before they blindly spread the message further, and can thus slow down the propagation process of computer malware or misinformation and win time for a more effective method (for example, a counter-campaign) to be further deployed. Second, in a dynamic environment where there are only sporadic adoptions of misinformation or computer malware, the containment strategy considered in this study can limit the infection to only the nodes that reside on the same islands as the initially contaminated nodes. If, however, the viral spread has already entered a number of islands before the containment scheme starts sanitizing suspicious interisland messages, it may become too late for the containment strategy to be effective.

For such cases, other complementary containment strategies such as launching a counter-campaign can be put in place to cancel out the effects of viral spread of computer malware or misinformation.



## 6. Appendix

The following lemma is standard and has appeared in many other places. It first appeared in [Łuczak 91].

**Lemma 6.1.** *Consider  $G_{n,M=cn}$ . For every  $\gamma > 0$ , there exists  $\alpha = \alpha(\gamma, c) > 0$  such that a.a.s. every subgraph  $H \subseteq G_{n,M}$  with fewer than  $\alpha n$  vertices has at most  $(1 + \gamma)|H|$  edges.*

Next, we show that property 1(b) of Section 3.3 holds for random graphs on a fixed degree sequence whenever that sequence satisfies property 1(a). Recall that property 1(a) says that for all  $\epsilon > 0$ , there exists a constant  $d$  such that the sum of all degrees greater than  $d$  is less than  $\epsilon n$ . Given  $d$ , we say that a *small edge* is an edge whose endpoints both have degree less than  $d$ . So we simply need to extend Lemma 6.1 to random graphs on such a degree sequence. We do so by adapting a standard proof.

**Lemma 6.2.** *Let  $G$  be a random graph on a given degree sequence, as described above. For any  $\epsilon, \gamma > 0$ , there exists  $\alpha > 0$  such that  $H \subseteq G$  with fewer than  $\alpha n$  vertices has at most  $(1 + \gamma)|H|$  small edges.*

**Proof.** We use the configuration model of Bollobás (see, e.g., [Bollobás 01]). So we create  $\deg(v)$  copies of each vertex  $v$ , and take a random pairing of all the copies. This yields a random multigraph in the obvious way. Other common properties of the degree sequence (see, e.g., [Molloy and Reed 95]) can allow us to condition on there being no loops or multiedges and hence obtain results for a random simple graph.

Suppose the total number of edges is  $M = cn$ ; then the total number of vertex copies is  $2M = 2cn$ .

We bound the expected number of subgraphs  $H$  with  $a \leq \alpha n$  vertices and fewer than  $(1 + \gamma)|H|$  small edges. Clearly, we can assume that  $H$  has no vertices of degree greater than  $d$  (where  $d = d(\epsilon)$  comes from property 1(a)), since a vertex of degree greater than  $d$  cannot contribute any small edges.

First, choose a set of  $a$  vertices. Choose  $2(1 + \gamma)a$  copies of vertices from that set. Then choose one of the

$$\frac{(2(1 + \gamma)a)!}{2^{(1+\gamma)a}((1 + \gamma)a)!} < \left( \frac{2(1 + \gamma)a}{e} \right)^{(1+\gamma)a}$$

ways to group those copies into pairs. The probability that each of those pairs arises in a random pairing is

$$\frac{1}{2M-1} \cdots \frac{1}{2M-2(1+\gamma)a+1} < \left(\frac{1}{cn}\right)^{(1+\gamma)a}$$

for  $a < \alpha n$  with  $\alpha$  sufficiently small. Putting everything together, and using the fact that each of the chosen vertices has degree at most  $d$ , the expected number of sets is at most

$$\begin{aligned} & \binom{n}{a} \binom{da}{2(1+\gamma)a} \left(\frac{2(1+\gamma)a}{e}\right)^{(1+\gamma)a} \left(\frac{1}{cn}\right)^{(1+\gamma)a} \\ & < \left(\frac{en}{a}\right)^a \left(\frac{eda}{2(1+\gamma)a}\right)^{2(1+\gamma)a} \left(\frac{2(1+\gamma)a}{e}\right)^{(1+\gamma)a} \left(\frac{1}{cn}\right)^{(1+\gamma)a} \\ & < Z^a \left(\frac{a}{n}\right)^{\gamma a}, \end{aligned}$$

where  $Z = Z(d, \gamma) = Z(\epsilon, \gamma)$  is a constant. Taking  $a < \alpha n$ , where  $\alpha^\gamma < 1/2Z$ , the rest now follows as in the proof of Lemma 6.1.  $\square$

**Acknowledgments.** Milan Bradonjić's research was partially supported by NIST grant number 60NANB10D128. Part of this work was done at Los Alamos National Laboratory.

## References

- [Andreev and Räcke 04] Konstantin Andreev and Harald Räcke. “Balanced Graph Partitioning.” In *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '04, pp. 120–124. ACM, 2004.
- [Ben-Naim and Krapivsky 04] E. Ben-Naim and P. L. Krapivsky. “Size of Outbreaks near the Epidemic Threshold.” *Phys. Rev. E*, 69:050901, May 2004.
- [Billingsley 79] Patrick Billingsley. *Probability and Measure*, Wiley series in probability and mathematical statistics. Wiley, 1979.
- [Bollobás 01] Béla Bollobás. *Random Graphs*, second edition. Cambridge University Press, 2001.
- [Budak et al. 11] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. “Limiting the Spread of Misinformation in Social Networks.” In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pp. 665–674. ACM, 2011.
- [Cohen et al. 03] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. “Efficient Immunization Strategies for Computer Networks and Populations.” *Physical Review Letters* 91, December 2003.
- [Erdős and Rényi 59] P. Erdős and A. Rényi. “On Random Graphs I.” *Publ. Math. Inst. Hungar. Acad. Sci.* 6 (1959), 290–297.

- [Erdős and Rényi 60] P. Erdős and A. Rényi. “On the Evolution of Random Graphs.” *Publ. Math. Inst. Hungar. Acad. Sci.* 5 (1960), 17–61.
- [Faloutsos et al. 04] Christos Faloutsos, Kevin S. McCurley, and Andrew Tomkins. “Connection Subgraphs in Social Networks.” In *Proceedings of Workshop on Link Analysis, Counterterrorism, and Privacy*, Affiliated with SIAM International Conference on Data Mining, 2004.
- [Frieze 85] Alan M. Frieze. “On the Value of a Random Minimum Spanning Tree Problem.” *Discrete Applied Mathematics* 10:1 (1985), 47–56.
- [Grimmett 99] Geoffrey Grimmett. *Percolation*, second edition, Grundle Math. Wissen. 321. Springer-Verlag, 1999.
- [Janson 10] Svante Janson. “Susceptibility of Random Graphs with Given Vertex Degrees.” *Journal of Combinatorics* 1:3-4 (2010), 357–387.
- [Kermack and McKendrick 33] William O. Kermack and Anderson G. McKendrick. “Contributions to the Mathematical Theory of Epidemics III: Further Studies of the Problem of Endemicity.” *Proceedings of the Royal Society of London* 141 (1933), 94–122.
- [Kumar et al. 06] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. “Structure and Evolution of Online Social Networks.” In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’06, pp. 611–617. ACM, 2006.
- [Leskovec et al. 09] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. “Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters.” *Internet Mathematics* 6:1 (2009), 29–123.
- [Łuczak 91] Tomasz Łuczak. “A Note on the Sharp Concentration of the Chromatic Number of Random Graphs.” *Combinatorica* 11 (1991), 295–297.
- [Molloy and Reed 95] Michael Molloy and Bruce Reed. “A Critical Point for Random Graphs with a Given Degree Sequence.” *Random Struct. Algorithms* 6 (1995), 161–179.
- [Moore et al. 03] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. “Inside the Slammer Worm.” *IEEE Security and Privacy* 1:4 (2003), 33–39.
- [Newman and Watts 99] M. E. J. Newman and D. J. Watts. “Scaling and Percolation in the Small-World Network Model.” *Physical Review E* 60:6 (1999), 7332–7342.
- [Pastor-Satorras and Vespignani 02] R. Pastor-Satorras and A. Vespignani. “Immunization of Complex Networks.” *Physical Review E* 65 (2002), 036104.
- [Sander et al. 02] L. M. Sander, C. P. Warren, I. M. Sokolov, C. Simonc, and J. Koopmand. “Percolation on Heterogeneous Networks as a Model for Epidemics.” *Mathematical Biosciences* 180:1-2 (2002), 293–305.
- [Sun et al. 09] B. Sun, G. Yan, Y. Xiao, and T. A. Yang. “Self-Propagating Mal-Packets in Wireless Sensor Networks: Dynamics and Defense Implications.” *Ad Hoc Networks* 7:8 (2009), 1489–1500.
- [Yan et al. 08] Guanhua Yan, Zhen Xiao, and Stephan Eidenbenz. “Catching Instant Messaging Worms with Change-Point Detection Techniques.” In *Proceedings of the*

*1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pp. 6:1–6:10, USENIX Association, 2008.

- [Yan et al. 11] Guanhua Yan, Guanling Chen, Stephan Eidenbenz, and Nan Li. “Malware Propagation in Online Social Networks: Nature, Dynamics, and Defense Implications.” In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS ’11, pp. 196–206. ACM, 2011.
- [Zou et al. 02] Cliff Changchun Zou, Weibo Gong, and Don Towsley. Code Red Worm Propagation Modeling and Analysis. In *CCS ’02: Proceedings of the 9th ACM conference on Computer and Communications Security*, pp. 138–147. ACM, 2002.
- [Zou et al. 04] C. C. Zou, D. F. Towsley, and W. Gong. “Email Worms Modeling and Defense.” In *Proceedings of ICCCN*, 2004.

---

Milan Bradonjić, Bell Labs, Alcatel-Lucent, 600 Mountain Avenue, Murray Hill, NJ 07974, USA (milan@research.bell-labs.com)

Michael Molloy, Department of Computer Science, University of Toronto, 10 King’s College Road, Toronto, ON M5S 3G4, Canada (molloy@cs.toronto.edu)

Guanhua Yan, Information Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA (ghyan@lanl.gov)