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

Milan Bradonjić*
Mathematics of Networks and Communications Group
Bell Labs, Alcatel-Lucent
600 Mountain Avenue, Murray Hill, New Jersey 07974, USA
E-mail: milan@research.bell-labs.com

Michael Molloy
Department of Computer Science
University of Toronto
10 King's College Road, Toronto, ON, Canada, M5S 3G4
E-mail: molloy@cs.toronto.edu

Guanhua Yan
Information Sciences Group
Los Alamos National Laboratory
Los Alamos, New Mexico, 87545, USA
E-mail: ghyan@lanl.gov

April 24, 2013

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 exits 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

*Research partially supported by NIST grant 60NANB10D128. Part of this work was done at Los Alamos National Laboratory.

focus on sparse random networks. The viral containment strategy is to partition a large network into small components and then to ensure the sanity of all messages delivered across different components. 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 inter-component messages. We further propose heuristic-based approaches to partition large input graphs into small components. Finally, we study the performance of our proposed algorithms under different network topologies and different edge weight models.

1 Introduction

Albeit computer worms came to existence more than two decades ago, they are still severely threatening the Internet security nowadays. Modern computer malware have commonly applied social engineering tricks for their propagation, due to the fact that social trusts among computer users have made them less vigilant against potential malware threats. For instance, the ILOVEYOU worm managed to infect tens of millions of Windows computers worldwide through e-mail attachments in 2000 [13]. Moreover, the emerging popularity of online social network sites such as Facebook [9] and Twitter [29] has provided a new playground for computer malware, as evidenced by a few recently spotted worms (for example, Koobface [16] and W32/KutWormer [24]) that specifically targeted these networks. In contrast to traditional Internet worms such as Code Red [33] and Slammer [21] that use port scanning to discover vulnerable machines, computer worms based on social engineering pose an even more severe cyber threat to many enterprise networks as they can easily penetrate through enterprise firewalls or Intrusion Detection Systems (IDSes).

Similarly to 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 become one of the major news sources for many people. Unfortunately, misinformation can also be spread on these social media networks, which, in some cases, cause undesirable consequences such as public panic. One such example is the spread of rumors regarding swine flu on Twitter in 2009 [28].

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 an epidemic spread passes a critical threshold or a take-off point, it may break out to become a pandemic [2]. One well-studied strategy in containing epidemic spread is *immunization* [6], 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 different domains or organizations, 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 the sanity of all messages that are delivered across islands. 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 only take place within the nodes in the same island as the initial infection point. From a practical standpoint, the strategy considered is only applicable to scenarios where 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 removing 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 partition is too large. Note that the problem is different from the balanced graph partitioning problem [1] 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 communication messages to inspect in a large network like Twitter[1]. 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

[1] There are $50,000,000$ messages transmitted within the Twitter network every day [30].

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 [10], so in this work we only consider sparse random graphs where the numbers of nodes and edges are of the same asymptotic order. Extending this work beyond sparse random graph models remains as our future work.

In a nutshell, our main contributions are given 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) the threshold on the number of edges to be deleted such that every connected component in the remaining graph has a size of at most a given constant, and (ii) the 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 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 towards better heuristic-based approaches in 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 interests in other application domains as well, as they essentially deal with a fundamentally theoretical problem on how to disintegrate networks under the 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, or bounded, or unbounded i.i.d. random variables. In Section 4.2, 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 [34]. 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 *el 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 [32] 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 [31]; 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 [31] was done empirically on a specific network topology, and thus, does not have a strong theoretic foundation.

Another motivating application of this work is containment of rumor or misinformation diffusion in large social networks. Research on this topic is still at its infancy. Budak *et al.* considered the problem of limiting the spread of misinformation in social networks [5]. Their approach was to convince a small set of people from whom a campaign of “good” rumors is started to minimize the influence by the “bad” rumors. 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.

The percolation theory has established the critical threshold for wide-scale epidemic spreading and has been widely applied to study epidemic spreading in diverse network structures, such as small-world networks [23], heterogeneous networks [26], and sensor networks [27]. 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 [34, 25]. In both 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 come 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 $G_{n,M}$ [7]. 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 with 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 $R = R(c) = cL\left(1 + e^{-2cL}\right)$. The giant component of G has $Ln + o(n)$ vertices [8] 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 αn.

(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 needed to be removed from a tree with bounded degree, such that each of 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 equal $t = t(\beta, d) = \max\{4d, r\}$, where r is the largest root of $x - (3.5/\beta) \log x = 0$.*

Proof: Denote the number of nodes in the tree $n = |H|$. 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 any tree of size $\leq k$, where $k \geq t$. Pick a node v, which is not a leaf of the tree, and denote its degree $d_v = \deg(v)$. Since v is not a leaf $d_v \geq 2$ and moreover $d_v \leq d$ by the conditions of the lemma. Denote $K_1, K_2, \dots, K_{d_v}$ the corresponding tree components incident to v. From $|K_1| + \cdots + |K_{d_v}| = n - 1 \geq t$, by the pigeonhole principle it follows that at least one of those components, denoted 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 the sizes $a = |H'|$ and $b = |H''|$. Then $a \geq t/d$ and $b \geq 2$ since v is not a leaf. H', H'' both 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 necessary to remove 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 ϕ_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 β is given by statement of the lemma and α is to be chosen later. Notice that $\phi_t(k) \geq 0$ for any k, given t is at least the largest root (among the two) 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 needed to be removed, including e, is at most:

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

It suffices to show

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

which is equivalent to

$$\frac{1}{a} + \frac{1}{b} \leq e^{-1/\alpha}. \tag{1}$$

Recall $a \geq t/d$ and $b \geq 2$. Choosing $\alpha = 3.5$ and the conditions of the lemma yield (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. □

Now we can prove Theorem 3.1.

Proof: *Part (a):* Let G' be formed by removing edges from $G = G_{n,M}$. A standard lemma (see Lemma A.1 in Appendix A) implies that there exists $\alpha = \alpha(\gamma, c) > 0$ such that every subgraph $H \subset G$ with fewer than αn vertices has fewer than $(1+\frac{1}{2}\gamma)|H|$ edges. Therefore, if every component of G' has size less than αn 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 $(Ln + o(n))(1 + \frac{1}{2}\gamma) < Ln+\frac{1}{2}\gamma n$, since $L < 1$. Therefore, we must have removed more than $(R-L-\gamma)n$ edges. This is contradiction, so there must be a component of size at least αn.

Part (b): Choose any spanning tree of the giant component. Remove all edges not in that tree. A.a.s. every other component that is not a tree has exactly one cycle. For each such component, remove an edge from that cycle. We expect to remove $O(1)$ such edges [4]. 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. [4]) show that the total number of edges incident with vertices of degree at least d is at most ϵn, where $\epsilon \to 0$ as $d \to \infty$. So we can choose d large enough in terms of γ, L, c so 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. □

3.1 Bounded edge weights

Now we extend Theorem 3.1 to the setting where the edges have weights. We begin with the case where 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.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 αn.

(b) We can remove edges of total weight at most $(R' - L' + \gamma)n$ so that every component has size at most t.

Proof: *Part (a):* Remove any set of edges. From Lemma A.1 it follows that for any $\gamma, W > 0$ there exists $\alpha > 0$ such that every subgraph $H \subset G$ with less, than αn vertices has at most $(1 + \frac{\gamma}{4W})|H|$ edges. Therefore, if every component has size less, than αn, then each such component H consists of a MaxST (maximum spanning tree) of H plus at most $\frac{\gamma}{4W}|H| + 1$ additional edges (since the MaxST 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.a.s. there are fewer than $\sqrt{n}$ components H with $|H| \leq 12W/\gamma$ containing any additional edges besides the MaxST.

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 MaxST of H_i plus $W \times \frac{\gamma}{3W}|H_i|$. The total weight of these MaxST's is at most $L'n$, the weight of a MaxST 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 MaxST's of all these H's. So the total weight of all remaining giant component edges is at most $W\sqrt{n} + L'n + \frac{\gamma}{3}\sum|H_i| \leq L'n + \frac{\gamma}{2}n$.

Part (b): We proceed as in the proof of Theorem 3.1 part (b). This time, we choose a maximal spanning tree of the giant component. The total weight of the giant component edges not in that tree is $(R' - L')n$. We choose d so 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 $F(x) = \int_0^w f(w)dw$ denote 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. $G_{n,M=cn}$ 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 αn.

(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 such 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 $\varphi(x) = \sum_{i=1}^{M} w_i \mathbb{I}\{w_i \geq x\}$ which depends on $x \geq 0$. Moreover $\mathbb{E}(\varphi(x)) = M\int_x^\infty wdF(w)$. By Lebesgue's dominated convergence theorem (see [3]), given finite mean $\int_0^\infty wdF(w) < \infty$, it follows that for any constants $\gamma, c > 0$ there exists sufficiently large W such that $M^{-1}\varphi(W) = \int_W^\infty wdF(w) \leq \gamma/(5c)$. (This statement would not necessarily hold for infinite mean $\int_0^\infty wdF(w) = \infty$.) Therefore, $\mathbb{E}(\varphi(W)) \leq M\gamma/(5c) = \gamma n/5$. 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.

Part (a): We proceed as in the proof of Theorem 3.3(a). As in that proof, the total weight of the remaining giant component 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 edges is less than $(L+\gamma)n$ and so we must have removed at least $(R-L-\gamma)n$ weight.

Part (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 $\frac{\gamma}{4}n$. □

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

Determining R' is straightforward. A.a.s. the number of edges in the giant component is $R = R(c)$, as given at the beginning of Section 3, above. 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 [11] yield the expected size of the $MaxSF$ of the entire graph. First determine the weights of the edges, without yet exposing their endpoints. Then sort these weights such that $w_1 \geq \cdots \geq w_n$. 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 $g(x)n + o(n)$ [8]. Analyzing Kruskal's Algorithm as in [11] yields that the probability that the edge with weight w_i belongs to the $MaxSF$ is $\pi(i/n) + o(1)$ where

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

Remark 3.5 *This is simply the probability, upon selecting the endpoints of the edges in order of their weights, that the ith 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 $MaxSF$ of the entire $G_{n,M}$ is given by

$$\mathbb{E}(MaxSF(G_{n,M})) = \sum_{i=1}^{M} \mathbb{E}(w_i)\pi(i/n) + o(n). \tag{3}$$

To obtain the weight of the $MaxST$ of just the giant component, we need to subtract the total weight of the rest of the forest. The graph outside of 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 of the giant component. There are $M - Rn$ such edges, and each has expected weight μ. This yields:

$$\mathbb{E}(L') = \sum_{i=1}^{M} \mathbb{E}(w_i)\pi(i/n) - (c - R)\mu n. \tag{4}$$

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 (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

$$\mathbb{E}(w_i) = M\binom{M-1}{i-1}\int_0^\infty wf(w)F(w)^{M-i}(1-F(w))^{i-1}dw\,. \tag{5}$$

In general (5) can be evaluated numerically. The following two cases when weights are drawn from the uniform or exponential distribution demonstrate a possibility to analytically express $\mathbb{E}(w_i)$.

Example 1: 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 2: Exponential weight distribution. The edge weights follow $f(x) = \lambda e^{-\lambda x}$ for $x \geq 0$. Then $\mu = 1/\lambda$ and $\mathbb{E}(w_i) = \lambda^{-1}M\binom{M-1}{i-1}\sum_{k=0}^{M-1}\binom{M-1}{k}(-1)^k/(k+i)^2$.

3.3 Other random graph models

One of the largest and most comprehensive recent studies on the structure of networks has been done in [18]. 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 $5,000$ to $14,000,000$ nodes, and from $6,000$ to $100,000,000$ edges. A key conclusion relevant to our work is that all of the studied networks showed to be very sparse, with average degree ≈ 2.5 for the networks of blog-posts, 400 for the network of movie ratings from Netflix [22], 10 (with median 6) for the social networks as well as most of the other studied networks, see [18]. 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 [17] shows that more than half 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, α such that
   (a) a.a.s. the sum of the degrees of all vertices of degree at least d is less than ϵn;
   (b) a.a.s. every subgraph S of size at most αn 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 non-trivial modification is that, in part (a), we must account for the at most ϵn remaining edges which 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 Appendix A 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 have 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 disintegrate 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* or *Edge Betweenness Centrality*. The steps of Algorithm 1 are as follows.

For an input graph G, find the maximum spanning forest $F = MaxSF(G)$. 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 k be the number of edges in F. At every time step $\tau = 1, \dots, k$, find an edge e^*_τ that maximizes $\Upsilon(F_\tau, e)$ of the current graph F_τ, defined by (6) or (7). 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.

Algorithm 1 Heuristic-Based Algorithm

```
Find the maximum spanning forest $F := MaxSF(G)$ of an input graph $G$.
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 $k$ be the number of edges in $F$.
for $\tau \leftarrow 1, k$ do
    Pick uniformly at random $e^*_\tau \in \arg\max_{e \in F_\tau} \Upsilon(F_\tau, e)$ (apply (6) or (7), respectively.)
    $B := B + w(e^*_\tau)$.
    $F_{\tau+1} := F_\tau - e^*_\tau$.
end for
```

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 susceptibility [12] of the graph $H - e$. That is, let connected components $C_1, C_2, \dots$ partition $H - e$. Then define the score to be:

$$\Upsilon_S(H, e) = \frac{1}{|V(H)|} \sum_i |C_i|^2 \,. \tag{6}$$

The concept of susceptibility is imported from theoretical physics, see [12]. Moreover, notice that (6) equals the expected component size of the given graph. We stress that susceptibility in this work should not be misinterpreted with susceptibility in epidemic models [15]. For example, susceptibility for a random graph with a given degree sequence is given in [14]. Informatively, 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 shortest paths that traverse e,

$$\Upsilon_{BC}(F_\tau, e) = \sum_{u,v \in F_\tau} \frac{\sigma(u, v|e)}{\sigma(u, v)}\,, \tag{7}$$

where $\sigma(u, v)$ is the number of shortest paths between u and v, and $\sigma(u, v|e)$ 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 two following 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 different 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) Algorithm described in Section 4.1 with the greedy heuristics (**MaxSF-Susceptibility**),

2) 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 the 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 in Appendix A 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 boxplot diagrams are standard box diagrams in statistics: representing median, and the 25th and 75th percentiles (the lower and upper quartiles, respectively). We heuristically chose width of the boxes to be 5.

From the results we make the following observations.

- For the two used heuristics, the susceptibility heuristic performs consistently better than the betweenness centrality heuristic in different combinations of network graphs and edge weight distributions. We, however, note 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 pay 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.

- 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 when 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 our future work.

5 Conclusions

This work is motivated by the challenges that arise in containing malware spread based on social engineering tricks and rumor spread in large social networks. We consider the 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 a 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 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 interests in other application domains as well as it essentially deals with a fundamental theoretic problem.

We hope that the theory developed in this study would 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 only effectively in the following scenarios. First, if we want to take a preventive approach against viral spread of computer malware or misinformation in large-scale online social networks, we need to ensure that the system have sufficient computational resources to prevent these information from crossing 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 among different islands in a large online social network. Hence, filtering techniques that can efficiently detect inter-island messages with patterns of interests (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 where 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 infections to only the nodes that reside on the same islands as those initially contaminated ones. If, however, the viral spread has already entered a number of islands before the containment scheme starts sanitizing suspicious inter-island 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.

References

[1] ANDREEV, K., AND RÄCKE, H. Balanced graph partitioning. In *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures* (New York, NY, USA, 2004), SPAA '04, ACM, pp. 120–124.

[2] BEN-NAIM, E., AND KRAPIVSKY, P. L. Size of outbreaks near the epidemic threshold. *Phys. Rev. E 69* (May 2004), 050901.

[3] BILLINGSLEY, P. *Probability and measure*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1979.

[4] BOLLOBÁS, B. *Random Graphs*, Second ed. Cambridge University Press, 2001.

[5] BUDAK, C., AGRAWAL, D., AND EL ABBADI, A. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web* (New York, NY, USA, 2011), WWW '11, ACM, pp. 665–674.

[6] COHEN, R., HAVLIN, S., AND BEN AVRAHAM, D. Efficient immunization strategies for computer networks and populations. *Physical Review Letters 91* (December 2003).

[7] ERDŐS, P., AND RÉNYI, A. On random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.* (1959).

[8] ERDŐS, P., AND RÉNYI, A. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.* (1960).

[9] `http://facebook.com/`.

[10] FALOUTSOS, C., MCCURLEY, K. S., AND TOMKINS, A. Connection subgraphs in social networks. In *Proceedings of Workshop on Link Analysis, Counterterrorism, and Privacy. Affiliated with SIAM International Conference on Data Mining* (2004).

[11] FRIEZE, A. M. On the value of a random minimum spanning tree problem. *Discrete Applied Mathematics 10*, 1 (1985), 47–56.

[12] GRIMMETT, G. *Percolation*, second ed. No. 321 in Grundl. Math. Wissen. Springer-Verlag, New York, 1999.

[13] `http://news.cnet.com/2100-1001-240112.html`.

[14] JANSON, S. Susceptibility of random graphs with given vertex degrees. *Journal of Combinatorics 1*, 3-4 (2010), 357–387.

[15] KERMACK, W. O., AND MCKENDRICK, A. G. 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.

[16] `http://www.kaspersky.com/news?id=207575670`.

[17] KUMAR, R., NOVAK, J., AND TOMKINS, A. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2006), KDD '06, ACM, pp. 611–617.

[18] LESKOVEC, J., LANG, K. J., DASGUPTA, A., AND MAHONEY, M. W. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics 6*, 1 (2009), 29–123.

[19] ŁUCZAK, T. A note on the sharp concentration of the chromatic number of random graphs. *Combinatorica 11* (1991), 295–297.

[20] MOLLOY, M., AND REED, B. A critical point for random graphs with a given degree sequence. *Random Struct. Algorithms 6* (March 1995), 161–179.

[21] MOORE, D., PAXSON, V., SAVAGE, S., SHANNON, C., STANIFORD, S., AND WEAVER, N. Inside the slammer worm. *IEEE Security and Privacy 1*, 4 (2003), 33–39.

[22] `www.netflix.com/`.

[23] NEWMAN, M. E. J., AND WATTS, D. J. Scaling and percolation in the small-world network model. *Physical Review E 60*, 6 (1999).

[24] `http://www.statemaster.com/encyclopedia/Orkut`.

[25] PASTOR-SATORRAS, R., AND VESPIGNANI, A. Immunization of complex networks. *Physical Review E 65* (2002).

[26] SANDER, L. M., WARRENA, C. P., SOKOLOVB, I. M., SIMONC, C., AND KOOPMAND, J. Percolation on heterogeneous networks as a model for epidemics. *Mathematical Biosciences 180*, 1-2 (2002).

[27] SUN, B., YAN, G., XIAO, Y., AND YANG, T. A. Self-propagating mal-packets in wireless sensor networks: Dynamics and defense implications. *Ad Hoc Networks 7*, 8 (2009).

[28] `http://neteffect.foreignpolicy.com/posts/2009/04/25/swine_flu_twitters_power_to_misinform`.

[29] `http://twitter.com/`.

[30] `http://blog.twitter.com/2010/02/measuring-tweets.html`.

[31] YAN, G., CHEN, G., EIDENBENZ, S., AND LI, N. Malware propagation in online social networks: nature, dynamics, and defense implications. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security* (New York, NY, USA, 2011), ASIACCS ’11, ACM, pp. 196–206.

[32] YAN, G., XIAO, Z., AND EIDENBENZ, S. Catching instant messaging worms with change-point detection techniques. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats* (Berkeley, CA, USA, 2008), USENIX Association, pp. 6:1–6:10.

[33] ZOU, C. C., GONG, W., AND TOWSLEY, D. Code red worm propagation modeling and analysis. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security* (New York, NY, USA, 2002), ACM, pp. 138–147.

[34] ZOU, C. C., TOWSLEY, D. F., AND GONG, W. Email worms modeling and defense. In *Proceedings of ICCCN* (2004).

A Appendix

The following lemma is standard and has appeared in many other places. It first appeared in [19].

Lemma A.1 *Consider $G_{n,M=cn}$. For any $\gamma > 0$ there exists $\alpha = \alpha(\gamma, c) > 0$ such that a.a.s. every subgraph $H \subseteq G_{n,M}$ with less than αn 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 constant d such that the sum of all degrees greater than d is less than ϵn. 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 A.1 to random graphs on such a degree sequence. We do so by adapting the standard proof.

Lemma A.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 αn vertices has at most $(1+\gamma)|H|$ small edges.*

Proof: We use the configuration model of Bollobás (see eg. [4]). 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 eg [20]) 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)) as 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 the random pairing is

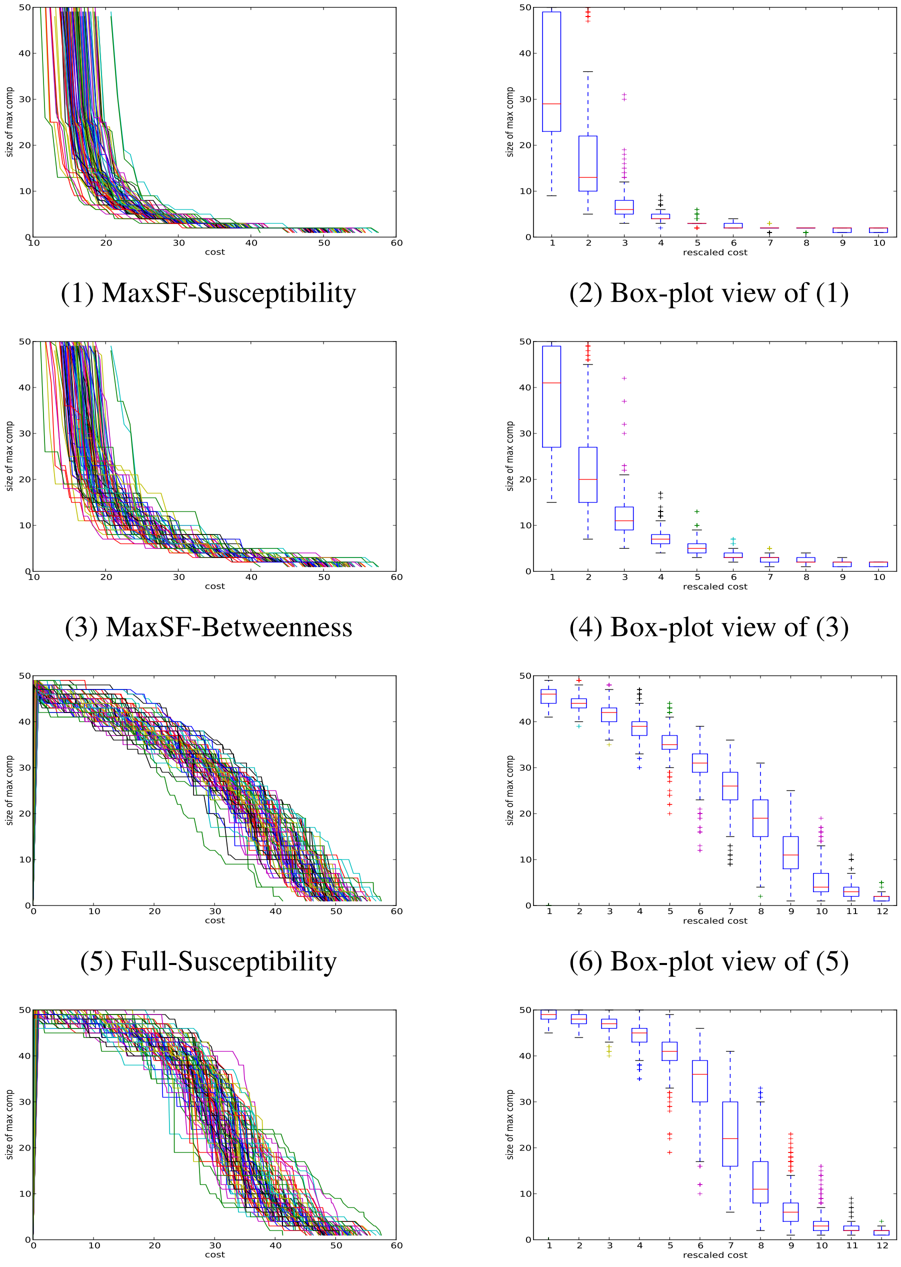
$$\frac{1}{2M-1} \cdot \ldots \cdot \frac{1}{2M-2(1+\gamma)a+1} < \left(\frac{1}{cn}\right)^{(1+\gamma)a}$$

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

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

where $Z = Z(d,\gamma) = Z(\epsilon,\gamma)$ is a constant. Taking $a < \alpha n$ where $\alpha^{\gamma} < \frac{1}{2Z}$, the rest now follows as in the proof of Lemma A.1. □

The following diagrams represent the experiments in the study on the performance of the heuristic-based algorithm and are accompanied with Section 4.2.



(1) MaxSF-Susceptibility

(2) Box-plot view of (1)

(3) MaxSF-Betweenness

(4) Box-plot view of (3)

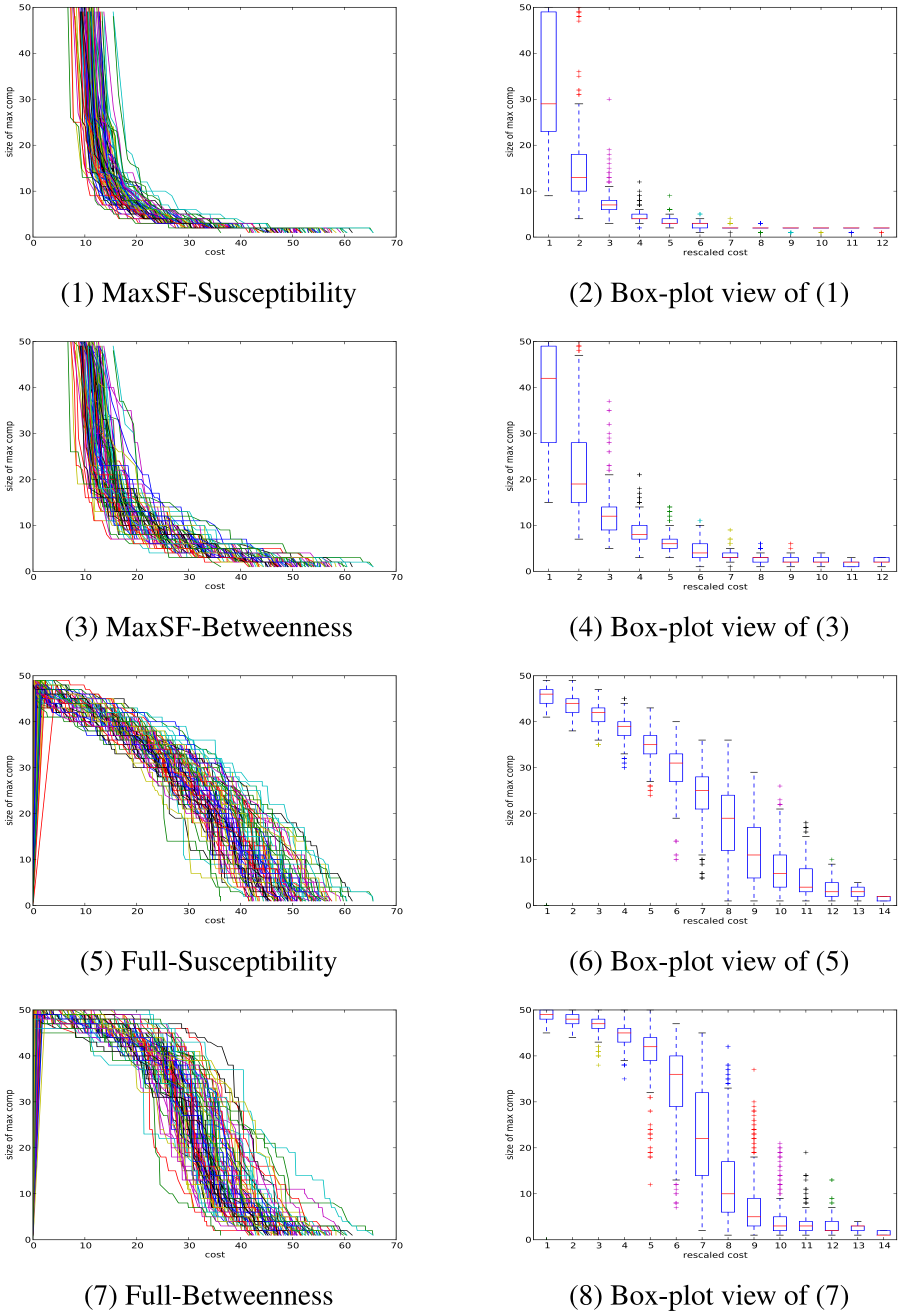
(5) Full-Susceptibility

(6) Box-plot view of (5)

(7) Full-Betweenness

(8) Box-plot view of (7)

Figure 1: Size of maximum component vs. edge removing cost for $G_{n,M}$ model with uniform edge weights.



(1) MaxSF-Susceptibility

(2) Box-plot view of (1)

(3) MaxSF-Betweenness

(4) Box-plot view of (3)

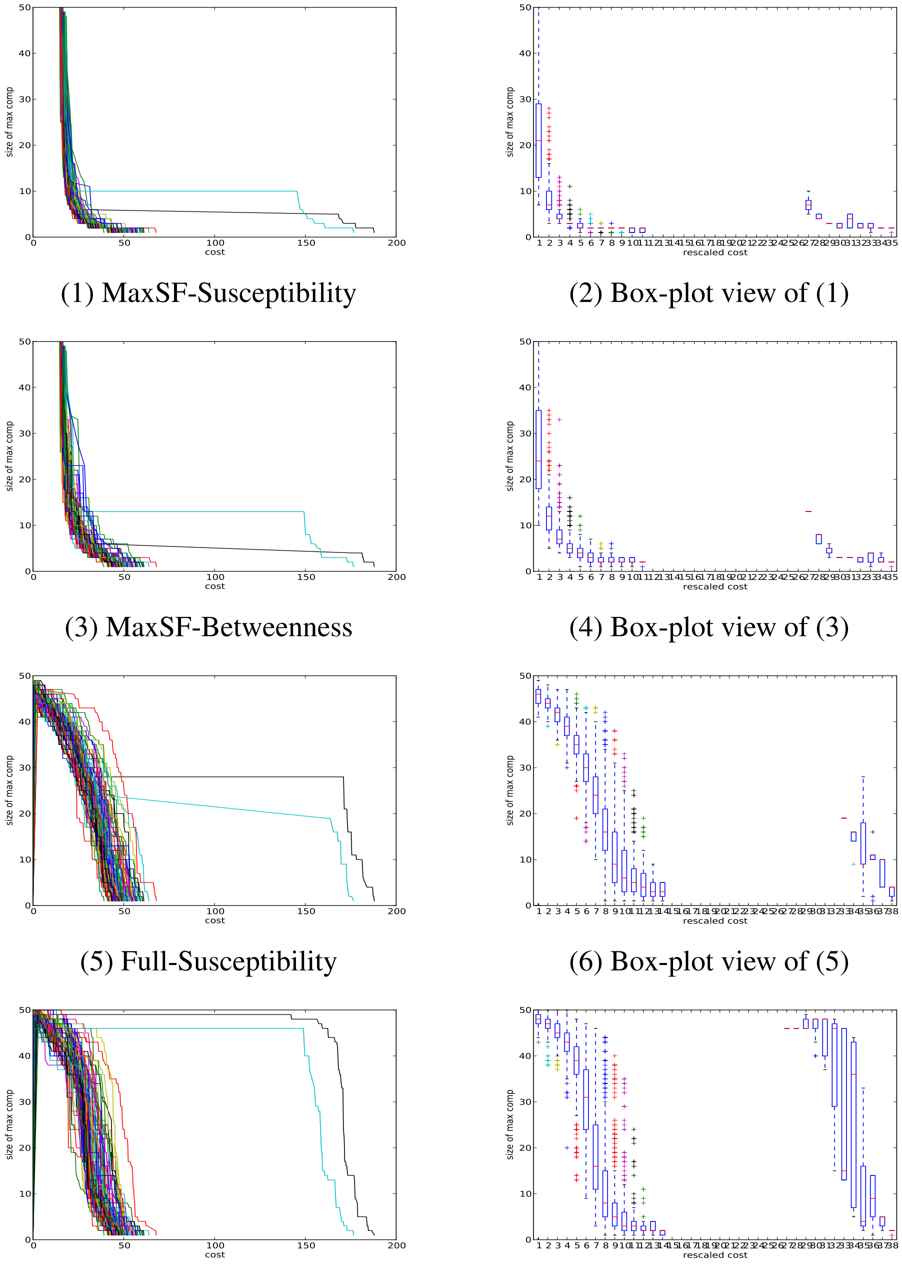
(5) Full-Susceptibility

(6) Box-plot view of (5)

(7) Full-Betweenness

(8) Box-plot view of (7)

Figure 2: Size of maximum component vs. edge removing cost for $G_{n,M}$ model with exponential edge weights.



(1) MaxSF-Susceptibility

(2) Box-plot view of (1)

(3) MaxSF-Betweenness

(4) Box-plot view of (3)

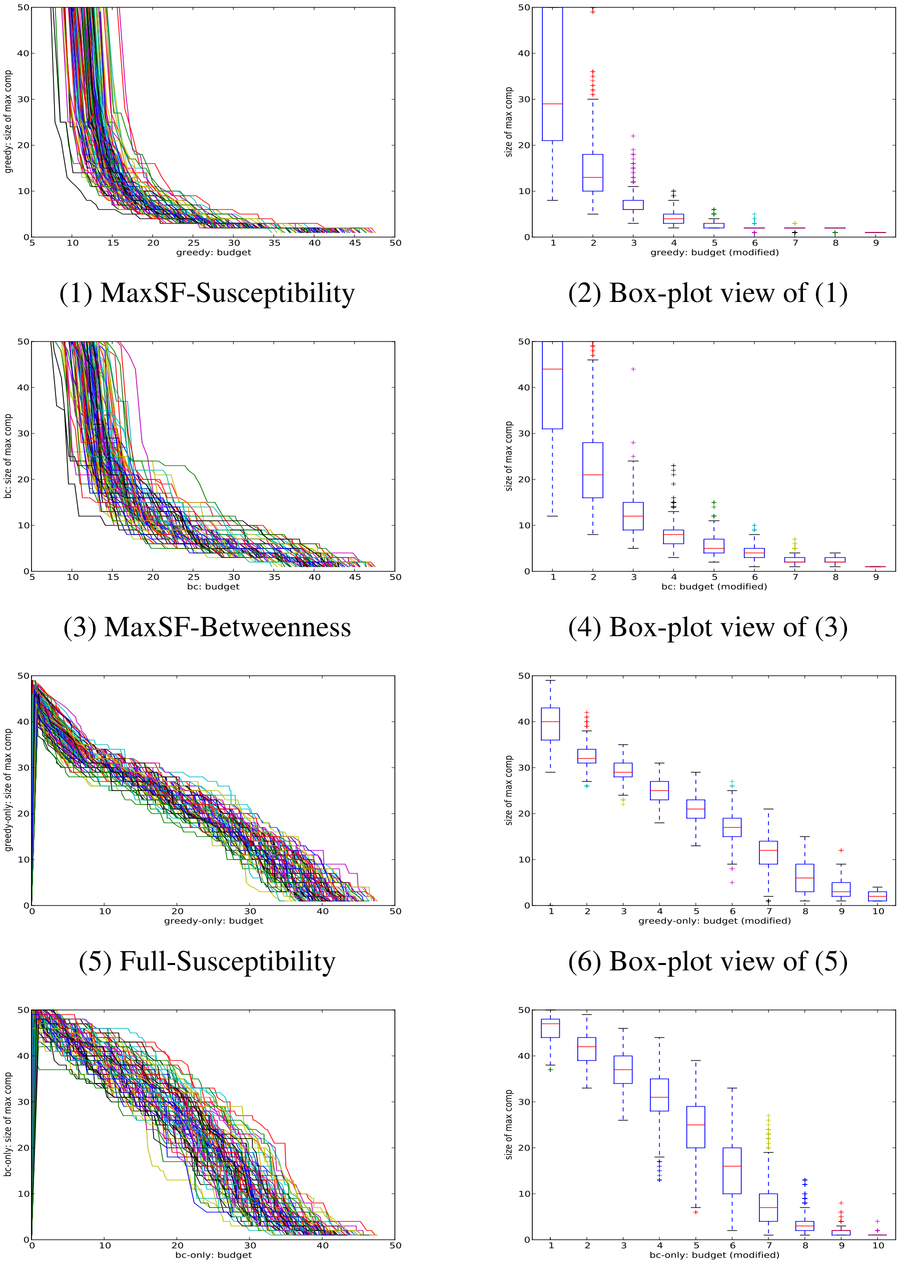
(5) Full-Susceptibility

(6) Box-plot view of (5)

(7) Full-Betweenness

(8) Box-plot view of (7)

Figure 3: Size of maximum component vs. edge removing cost for $G_{n,M}$ model with power-law edge weights.



(1) MaxSF-Susceptibility

(2) Box-plot view of (1)

(3) MaxSF-Betweenness

(4) Box-plot view of (3)

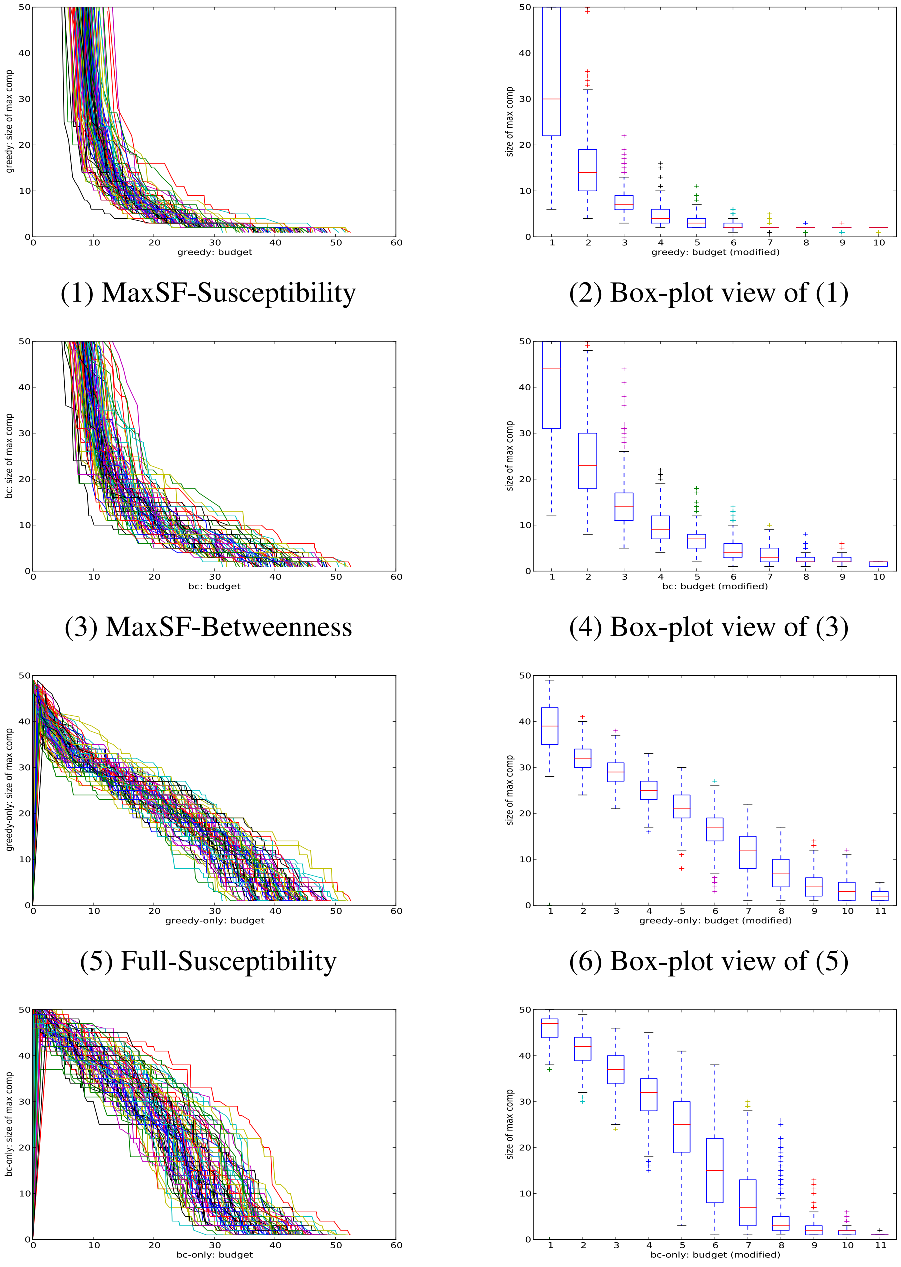
(5) Full-Susceptibility

(6) Box-plot view of (5)

(7) Full-Betweenness

(8) Box-plot view of (7)

Figure 4: Size of maximum component vs. edge removing cost for random graph model with power-law degree sequence, with uniform edge weights.



(1) MaxSF-Susceptibility

(2) Box-plot view of (1)

(3) MaxSF-Betweenness

(4) Box-plot view of (3)

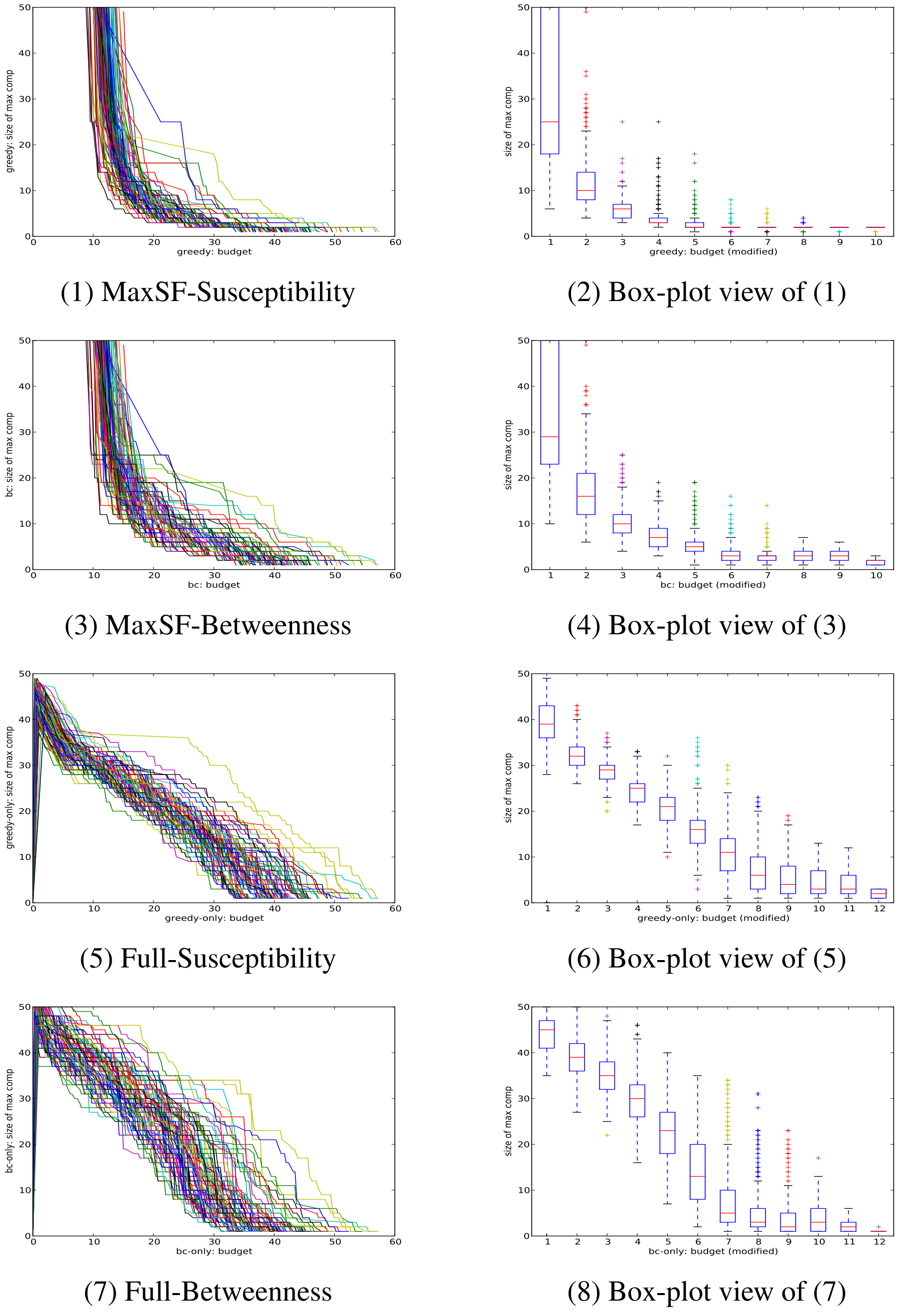
(5) Full-Susceptibility

(6) Box-plot view of (5)

(7) Full-Betweenness

(8) Box-plot view of (7)

Figure 5: Size of maximum component vs. edge removing cost for random graph model with power-law degree sequence, with exponential edge weights.



(1) MaxSF-Susceptibility

(2) Box-plot view of (1)

(3) MaxSF-Betweenness

(4) Box-plot view of (3)

(5) Full-Susceptibility

(6) Box-plot view of (5)

(7) Full-Betweenness

(8) Box-plot view of (7)

Figure 6: Size of maximum component vs. edge removing cost for random graph model with power-law degree sequence, with power-law edge weights.