The Marginal Benefit of Monitor Placement on Networks

Chapter · January 2016

DOI: 10.1007/978-3-319-30569-1_7

5 authors, including:

Ralucca Gera
Naval Postgraduate School
60 PUBLICATIONS **623** CITATIONS
SEE PROFILE

Gary Lazzaro
United States Naval Academy
21 PUBLICATIONS **18** CITATIONS
SEE PROFILE

The Marginal Benefit of Monitor Placement on Networks

Benjamin Davis, Ralucca Gera, Gary Lazzaro, Bing Yong Lim and Erik C. Rye

Abstract Inferring the structure of an unknown network is a difficult problem of interest to researchers, academics, and industrialists. We develop a novel algorithm to infer nodes and edges in an unknown network. Our algorithm utilizes monitors that detect incident edges and adjacent nodes with their labels and degrees. The algorithm infers the network through a preferential random walk with a probabilistic restart at a previously discovered but unmonitored node, or a random teleportation to an unexplored node. Our algorithm outperforms random walk inference and random placement of monitors inference in edge discovery in all test cases. Our algorithm outperforms both methodologies in node inference in synthetic test networks; on real networks it outperforms them in the beginning of the inference. Finally, a website was created where these algorithms can be tested live on preloaded networks or custom networks as desired by the user. The visualization also displays the network as it is being inferred, and provides other statistics about the real and inferred networks.

1 Introduction

The exploration of complex networks is a continuously evolving study as technology progresses and networks change. In today's world, there are many networks that are unknown. How do we gain insight into these unknown networks without having to traverse every vertex and edge within the network? Is there a way to place monitors at different areas of the network to gain this insight? The objective of this paper is to explore the topic of monitor placement on network vertices in an attempt to gain insight into the true network topology.

B. Davis · R. Gera (✉) · G. Lazzaro · B.Y. Lim · E.C. Rye
Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, USA
e-mail: RGera@nps.edu

H. Cherifi et al. (eds.), *Complex Networks VII*, Studies in Computational Intelligence 644, DOI 10.1007/978-3-319-30569-1_7

1.1 Motivation

In this paper we assume no knowledge of the true network, except for a rough approximation of the number of nodes so that the algorithm has a stopping condition. The algorithm used for network inference is tested on different synthetic and real-world complex networks of same order. The test networks are introduced in Table 1. Comparison of performance of an algorithm amongst these different test networks are normalized by looking at percentages, that is, the number of inferred nodes divided by the approximate number of total nodes, or the number of inferred edges divided by the approximate number of total edges.

In this paper we answer the following questions: As we increase the number of monitors placed up to 50 % of the nodes of the true network, what is the percent gain of new information inferred from the original network? At what percentage of monitor placement does the discovery of inferred network information begins to diminish towards a flat rate of change of the monitors discovered per monitor added? What is the minimum percentage of monitors needed to discover all nodes?

The website http://faculty.nps.edu/rgera/projects.html [4] was created where these algorithms can be tested live on preloaded networks or custom networks uploaded by the user. The visualization also displays the network as it is being inferred and that correlation to the percent edges and nodes inferred, and it provides other statistics about the real and inferred networks. Figure 1 shows two snapshots of the website, displaying the network as it is being inferred in green (top left of each figure), the leftover part of the network in white (top right), the plot of edges and nodes inferred (bottom left), and a heat-map of accuracy at each step in the inference (bottom right). Confidence intervals around the percent edges and nodes can be displayed by using multiple runs.

Table 1 Overview of the discovered data

Metrics	GR	ER	BA	FB
Node count (True network)	5242	5242	5242	4039
Edge count (True network)	14496	14496	15717	88234
Node count ($p = 0$)	4387	5083	5225	4002
Edge count ($p = 0$)	12598	12373	1418	82378
Node count ($p = 1$)	3823	5078	5223	3935
Edge count ($p = 1$)	12598	12358	14432	82179
Node count (Ideal)	5182	5201	5242	4039
Edge count (Ideal)	14348	13864	15675	85485
Node count (RW)	4491	4924	5162	3550
Edge count (RW)	12095	10976	13370	75971
Node count (RP)	4746	5090	5056	4009
Edge count (RP)	10530	10898	11747	66643

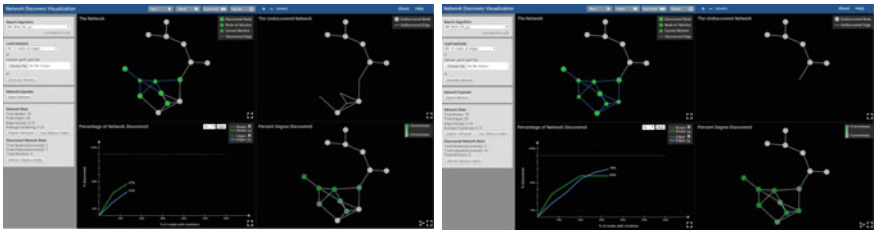


Fig. 1 Two steps in the inference of an Erdős-Rényi network from [4]

1.2 Related Work

Inferring a network can be done with no knowledge of the network at all (other than some random starting node), with partial information collected from network devices (such as knowledge of some of the nodes present in the network), or with complete information (in which case one could use the current knowledge to further monitor the network, or to re-infer an evolving network). Bliss, Danforth and Dodds [3] present recent techniques of inferring the topology of complex networks. These techniques are based on sampling nodes, sampling edges, the exploration of networks using random walks, or snowball sampling based on chain referral sampling ([2, 5]). Of course, the most relevant question is measuring the inferred network against the true network: random edge selection, depth and breath first search graph traversal, do not perform well overall; simple uniform random node selection performs surprisingly well; the best performing methods are based on random-walks starting at an arbitrary seed node (with the added probability of p at each node to teleport out of the random walk to the seed node or another arbitrary node) [5]. High degree nodes play the important role of hubs in communication and networking, and different local search strategies in power-law graphs that have costs scaling sublinearly with the size of the graph were introduced in [1]. However, the monitors in this paper infer more than just the node and edge incident with it, and thus the techniques perform differently.

Other current techniques not necessarily using complex networks are based on differential equations given one observation of one collective dynamical trajectory [11], statistical dependence between observations [14], as well as machine learning based on frequency of small subgraphs [7]. Extensions to multilayered networks have recently been published in [12, 13].

2 Preliminaries

We define a monitor to be able to see the node where it is placed, the edges incident to it, its neighbors, and possessing the ability to detect the degree and labels of its neighbors (the labels of the true topology as it is being inferred). For example, if

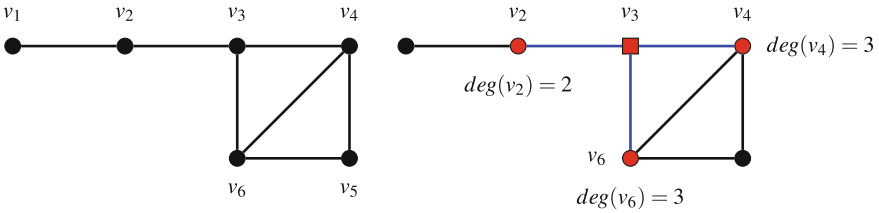


Fig. 2 A graph and a monitor placed at node v_3

each of two monitors i and j individually detect node k, they identify that it is the same k. We introduce this formally below.

Definition 1 We say that a *monitor on node i detects a node j* if (a) $d(i, j) \leq 1$, and (b) i knows the label of j and the deg j. A *monitor on node i detects an edge ij* if i and ij are incident.

Notice that a monitor always detects its closed neighborhood $N[v]$, but it infers more than just its neighbors. This is the idea used behind the domination number, $\gamma(G)$, in graphs introduced by Ore [10]. We say that a vertex *dominates* itself and its neighbors. Recall that a *dominating set* is a subset of the nodes such that each vertex of $V(G)$ is dominated by some vertex in the dominating set. The *domination number* is the cardinality of a minimum dominating set of G. The domination number could definitely be used to monitor a network, if the network is known. But in our approach of discovering the network, this is not useful since we do not assume to have much knowledge of the network (Fig. 2).

The k-Vertex Maximum Domination, introduced by Miyano and Ono in [8], is the parameter that gives the ideal placement of monitors if complete network information is known. Given a positive integer k, k-Vertex Maximum Domination (k-MaxVD) finds a subset DN of the nodes with size k that maximizes the cardinality of dominated nodes. That is, maximize $\cup_{v \in DN} N[v]$. Note that this optimization may produce a dominating set for some values of k, but does not need to, because in general not all nodes in the network are dominated. In [8], the authors show that a simple greedy strategy achieves an approximation ratio of $1 - \frac{1}{e}$ for k-MaxVD, and this approximation ratio is the best possible for k-MaxVD unless P = NP. We thus plot our inference algorithms against a greedy approach as an upper bound, and a random placement and a random walk as a lower bound on the performance of the algorithms. We refer to [9] for additional terminology not included in this paper.

3 Methodology

In this section we describe the approach used in placing monitors to infer an unknown network. We create a hill-climbing algorithm starting at some random node, with a probabilistic restart. Our algorithm first picks an initial "seed" node at random to

place the first monitor. The monitor discovers the labels of its neighbors and incident edges to the monitored node. Next, the highest degree node neighbor to the monitor is chosen for the next monitor. If multiple highest degree neighbors exist, one is chosen at random.

If the process attempts to place a monitor at a node where a monitor already exists, then a stopping condition is reached. The next "seed" node could be either a previously unseen node that is discovered at random, i.e. it teleports (when $p = 0$), or the next highest degree node that was previously discovered and not used as a monitor (when $p = 1$), or a combination of both approaches (when $0 < p < 1$).

We present an initial bound on the number of monitors needed for network inference based on our algorithm. The best case is if the network topology is star. Either the first or second monitor would be placed at the center node in the best case. The worse case scenario is when the graph is a path, and the first monitor is placed at a leaf node. In that case, it would take $n - 1$ monitors to discover all n nodes of the network. We thus have the following remark, and the bounds are sharp given by the star and path described above: $1 \leq num\_monitors \leq n - 1$.

Algorithm 1 Hill-Climbing: High-degree neighbor with restart by teleportation or large seen degree

p, a given probability
$monitor \leftarrow$ randomly chosen from the network
$seen\_nodes\_list \leftarrow \emptyset$
$inferred\_graph \leftarrow \emptyset$
while 50% of the nodes in the network unmonitored **do**
  Add $monitor$ to $seen\_nodes\_list$
  Add all edges and nodes attached to the $monitor$ to the $inferred\_graph$
  Add neighbors of the $monitor$ that have not yet been discovered to the $seen\_nodes\_list$
  $highest\_deg\_node \leftarrow$ neighbor of $monitor$ with highest degree
  if $highest\_deg\_node$ does not have a monitor **then**
    $monitor \leftarrow highest\_deg\_node$
  else
    With probability $(1 - p)$, choose $monitor \leftarrow$ node randomly chosen from the complement of the $inferred\_graph$
    Otherwise, $monitor \leftarrow$ node with max degree in $seen\_nodes\_list$

4 Results and Discussion

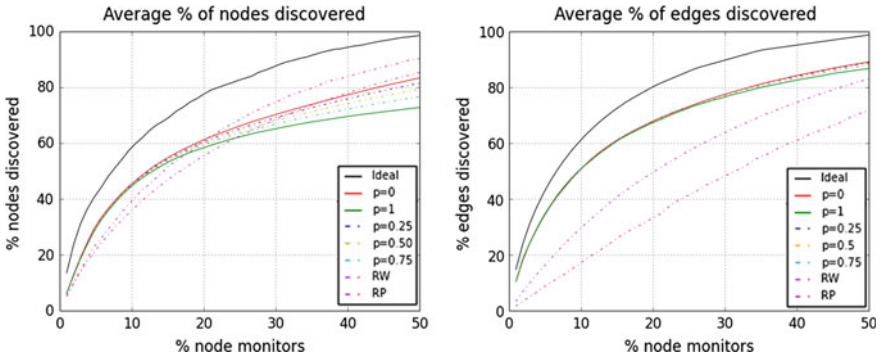
Table 1 presents general information regarding the four data sets used in this paper: One Erdős-Rényi (ER) network, one Barabási-Albert (BA) network, one Facebook (FB) network and one General Relativity collaboration (GR) network. The real networks are from the Stanford large network data set collection [6]. The node count, edge count and number of components is shown.

The performance of each algorithm is shown and discussed for the average of the 50 trials. We plot our inference algorithms against a greedy approach as an upper bound (called $Ideal$ shown in black), and two lower bounds shown in different shades of freesia representing Random Placement (RP) and Random Walk (RW). The monitors for all choices of Ideal, RW and RP are the same as we introduced for our research.

4.1 General Relativity Collaboration Network

The General Relativity collaboration network is comprised of 5242 nodes with 14496 edges, where an edge connects two nodes representing authors who have published a scholarly article together. This network consists of 355 distinct components. Figure 3 displays our inference of this network.

In this network, we achieve the best results using our algorithm with $p = 0$, outperforming all other p values, as well as the random walk (RW) and random placement (RP) strategies until about 20 % of the nodes are monitored. After the 20 % mark, RP captures a higher percentage of the nodes in the underlying network. The success of $p = 0$ initially and RP afterward is likely attributable to their preference for jumping to distinct topological components, thus capturing topology unlikely to be "seen" by inference algorithms that tend to stay within a component. RW and our algorithm when $p = 1$ tend to exhibit this "component-bound" behavior. In terms of edges, our algorithm with all p values tested ($p = 0, 0.25, 0.5, 0.75, 1$) discovered significantly more edges than the RW and RP inference algorithms as the number of monitors increased. We believe this effect is attributable to the preference for higher degree neighbors when selecting the next monitor. Neither the RW or RP algorithms



Percent of nodes inferred from General Relativity Network

Percent of edges inferred from General Relativity Network

Fig. 3 General relativity network: percent of nodes and edges in the inferred graph

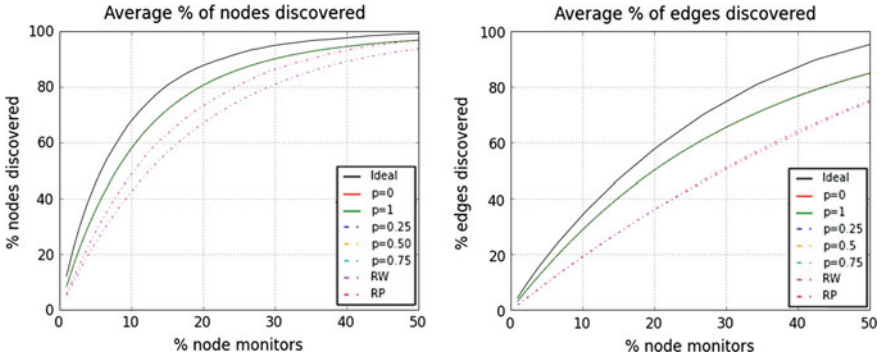
prefer high-degree neighbors when selecting a successor monitor, which contributes to their under-performance in edge discovery. Figure 3 displays this difference in edge discovery.

4.2 *Erdős-Rényi Random Graph*

In this section, we examine the results of our inference on an Erdős-Rényi random graph, of comparable order and size to the collaboration network studied above. Of note, however, our Erdős-Rényi graph consists of only 19 connected components, compared to 355 in the General Relativity collaboration network. Figure 4 displays the results of our inference trials.

When $p = 0$, 7 of the 12 connected components are discovered, accounting for 98 % of the network nodes when 50 % of the nodes are monitored. The rate of discovery is quite high initially, with roughly 80 % of the nodes discovered after approximately 20 % of the nodes in the network are monitored. When $p = 1$, we achieve nearly identical results in terms of nodes discovered with 50 % of the nodes in the network monitored, and a slightly higher number of edges inferred (11804 vs. 12455 for $p = 0$ and $p = 1$, respectively.) Interestingly, in the $p = 1$ case, all nodes and edges discovered were contained within a single component. This reinforces the "component-bound" behavior of $p = 1$, and poses an interesting question to a potential customer of our algorithm: given an approximately equivalent amount of topological inference, is the discovery of more components within a network more or less desirable? We believe there are cases to be made for each elsewhere; here, we merely highlight this distinction.

Our investigation of variable p values are bounded by $p = 0$ and $p = 1$. Due to the tightness of the limiting p values, the variable p values do not provide much



Nodes inferred from ER Network Edges inferred from ER Network

Fig. 4 ER random networks: percent of nodes and edges in inferred graph

additional value, besides highlighting increased component discovery tendencies of low p and additional edge discovery as p increases.

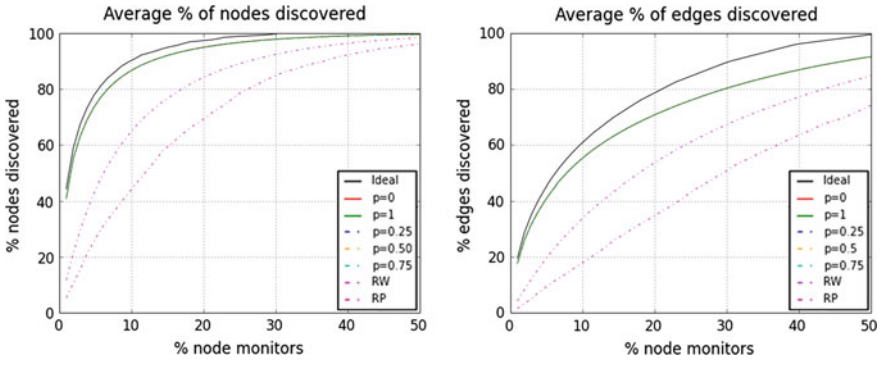
4.3 *Preferential Attachment Model: Barabási Albert Networks*

Our graph inference trial involving a Barabási Albert-model graph was performed on a network consisting of 5242 nodes and 15717 edges, matching the number of nodes in the General Relativity example.

By construction, this network is connected, and it has hubs, unlike the General Relativity and the Erdős-Rényi Random Graphs. Due to the propensity of high degree hubs to form in the Barabási-Albert network construction model, our algorithm captures a large percentage of the nodes in the ground truth topology using relatively few monitors regardless of the choice of p value. This is evident in Fig. 5, in which the inference results for different values of p overlap throughout, significantly outperforming the random placement and random walk inferences.

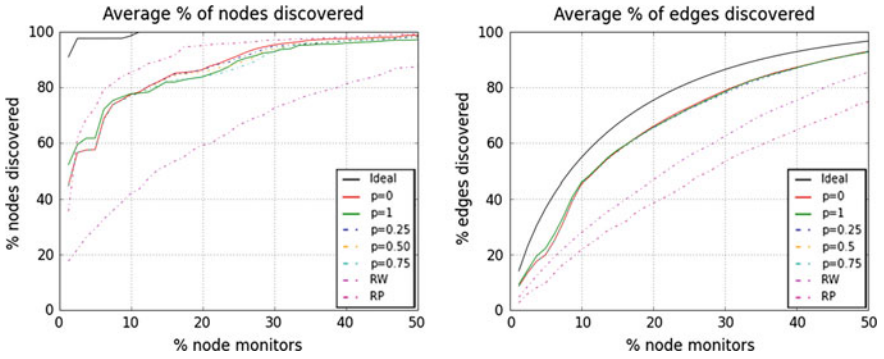
This effect is due to hubs being discovered within a couple of steps from the seed, and selected for monitor placement early in the algorithm's execution. Further, we can see a diminishing return on investment as the number of monitors placed in the original graph increases, both on edges and nodes as the hubs are close to each other.

As we increase the number of monitors placed from zero up to 50 % of the nodes in the true network, the percent gain of new information per new monitor added quickly tends toward zero. In terms of nodes discovered, the derivative of the function given by our our curve in Fig. 5 decreases from a maximum of about 0.5% marginal gain at the first fifty-monitor step to about 0.1% marginal gain nodes discovered when 20% of the nodes in the graph are monitors.



Nodes inferred from BA Network Edges inferred from BA Network

Fig. 5 BA network: percent of nodes and edges in the inferred graph



Nodes inferred from FB Network Edges inferred from FB Network

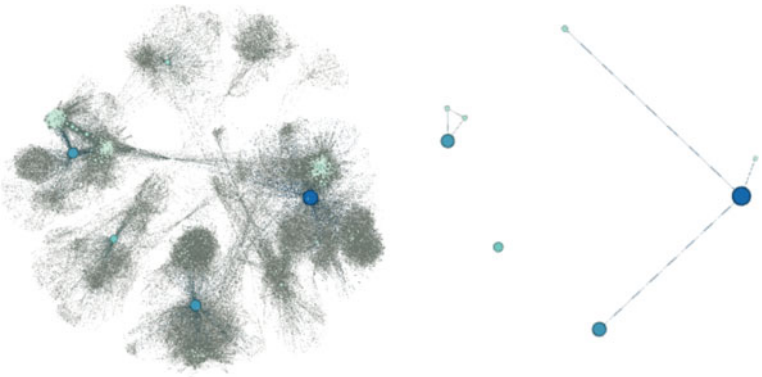
Fig. 6 FB network: percent of nodes and edges in the inferred graph

4.4 Facebook Network

The final network we consider is a Facebook ego network from [6], consisting of 4039 nodes and 88234 edges. It forms a connected graph, and of note, is much more highly connected than any network studied above. The results of our inference trials are presented in Fig. 6.

In terms of edges, not much difference exists between the $p = 0$ and $p = 1$ algorithms, and our algorithms clearly outperform RW and RP. When we consider node inference, however, some variation is observed. Well-defined hubs enable node inference very quickly with few monitors; however, the existence of distinct communities tends to create "steps", due to all nodes within a particular community being exhausted as monitors before new nodes can be discovered in a disparate community. Due to this effect, out of all the choices of p, the value of $p = 0$ performs best due to its tendency to restart inference at a randomly selected node. This allows the inferences that are based just on random walks and placements to outperform our inferences after a certain point.

As a concrete demonstration of this phenomenon, consider only the vertices of degree greater than 250. Figure 7a shows the entire network, and the plot of Fig. 7b shows the graph induced on the vertices with degree greater than 250 which reveals 3 components. Consider these 3 components as communities for the entire Facebook network. One cluster has only one hub, and the second cluster has a triangle of hubs. The third cluster is centered around a star of hubs. Recall that since the entire Facebook graph is connected, we know that these clusters must be connected to each other through lower degree vertices. Thus, interconnecting paths between the clusters of the Facebook Network must contain an edge that is incident to vertices that have a low degree (since these clusters are not connected). The affinity of core hubs for each other can be measured by the Pearson correlation coefficient as mentioned in [9]. The Pearson correlation coefficient for the Facebook network is computed as 0.064,



Facebook: Overall Network Facebook: Vertices with degree above 250

Fig. 7 Facebook network visualizations

which is not indicative of either strong assortative or disassortative mixing of the graph vertices. On the Facebook network, we discovered a vertex degree of 66 to connect the entire network. Thus, we discover the nodes as a step function since many high degree nodes need to be used as monitors before getting to the lower degree node connecting the clusters.

5 Conclusions

In this research, we introduced a hill-climbing algorithm that infers a network with no knowledge of the network other than random nodes to start (or restart) the algorithm. Our algorithm has a probabilistic restart once it wants to place a monitor on a node that is occupied by a monitor: when $p = 1$ the algorithm restarts at a large degree node that has been discovered, versus when $p = 0$ the algorithm restarts at a random node of the network, and there are all the choices in between for the variable $0 \leq p \leq 1$ as expected. The value of p is chosen before the algorithm starts.

We analyzed real and synthetic networks, and present an analysis based on 50 runs of the algorithm for several values of p concluding that there is very little difference between the algorithms when we are concerned with edges being inferred. If the inference of nodes is the main goal, it is interesting to see the clear difference between the real networks and the synthetic networks. On the synthetic networks, there is no difference between any of our algorithms and they outperform the random placement and random walk, being extremely close to the ideal case in the presence of hubs. On the real networks, we see lots of variance between the algorithms, our inferences outperforming the random walks and placement towards the beginning of the inference, at which point random placement performs better since it does not

choose nodes in the same clusters or component as our algorithms do. This suggests that the current algorithms should be used for quick inferences with a few monitors. Also, on the real networks, we observed that if there are no random restarts, our algorithm infers the denser part of the graph in more detail.

A user that desires to infer an unknown complex network with this algorithm needs to know a rough estimate of the size of the network to define the budget of total monitors, which was set to $n/2$ in this research. Secondly, the user needs to have a goal of inferring nodes or edges. If the inferred nodes are the goal, then select $p = 0$, and for edges edges select $p = 1$. The variable parameter (probability of staying within the current component) p of the algorithm combines the two different kinds of search methodologies, namely edge-finding or node-finding allowing a better discovery of both nodes and edges on average.

6 Further Studies

It is assessed that the real world networks could be accurately correlated to

Open question 1: One possible extension for the detection algorithm would be to increase the capability of the detection monitor. Future work could consider a monitor that has the capability to detect a triangle, that is, the ability to detect neighboring vertices, the edges to neighboring vertices, and edges between those neighbors; or nodes at further steps from the monitors.

Open question 2: Another possible improvement is to combine algorithms after a certain number of steps, or to add restarts more often. This will avoid the step increases observed in the Facebook network due to the clusters of hubs.

Open question 3: The biggest improvement that the authors see is finding a way of comparing the topology of inferred networks to the true network that uses other metrics besides the percent nodes and percent edges discovered. This requires a different type of analysis complementing this article.

Acknowledgments The authors would like to thank the DoD for partially sponsoring the current research. We would also like to thank and acknowledge the Naval Postgraduate School's Center for Educational Design, Development, and Distribution (CED3) for creating the live visualization [4] for this project.

References

1. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in power-law networks. Phys. Rev. E **64**(4), 046135 (2001)
2. Biernacki, Patrick: Waldorf, Dan: Snowball sampling: problems and techniques of chain referral sampling. Sociol. Methods Res. **10**(2), 141–163 (1981)
3. Bliss, C.A., Danforth, C.M., Dodds, P.S.: Estimation of global network statistics from incomplete data. PloS One **9**(10), e108471 (2014)

4. Gera, R.: Network discovery visualization: an analysis of network discovery (2015). http://faculty.nps.edu/rgera/projects
5. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 631–636. ACM (2006)
6. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection (2014). http://snap.stanford.edu/data
7. Middendorf, M., Ziv, E., Wiggins, C.H.: Inferring network mechanisms: the drosophila melanogaster protein interaction network. Proc. Natl. Acad. Sci. U.S.A. **102**(9), 3192–3197 (2005)
8. Miyano, E., Ono, H.: Maximum domination problem. In: Proceedings of the Seventeenth Computing: The Australasian Theory Symposium, vol. 119, pp. 55–62. Australian Computer Society Inc (2011)
9. Newman, Mark: Networks: An Introduction. Oxford University Press Inc, New York (2010)
10. Ore, O.: Theory of graphs. Am. Math. Soc. Colloq. Publ. **38**, (1962)
11. Shandilya, S.G., Timme, M.: Inferring network topology from complex dynamics. New J. Phys. **13**(1), 013004 (2011)
12. Sharma, R., Magnani, M., Montesi, D.: Missing data in multiplex networks: a preliminary study. In: Third International Workshop on Complex Networks and their Applications (2014)
13. Sharma, R., Magnani, M., Montesi, D.: Investigating the types and effects of missing data in multilayer networks. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (2015)
14. Tieu, K., Dalley, G., Grimson, W.E.L.: Inference of non-overlapping camera network topology by measuring statistical dependence. In: Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005, vol. 2, pp. 1842–1849. IEEE (2005)