Fall 10-30-2015

Network Visualization

Brittany Q. Reynolds
Valparaiso University, brittany.reynolds@valpo.edu

Karl Schmitt
Valparaiso University

Network Visualization Project

Brittany Reynolds & Dr. Karl Schmitt
Collaborators: Erin Moore & Dr. Ralucca Gera

VALPARAISO UNIVERSITY
NAVAL POSTGRADUATE SCHOOL

Abstract

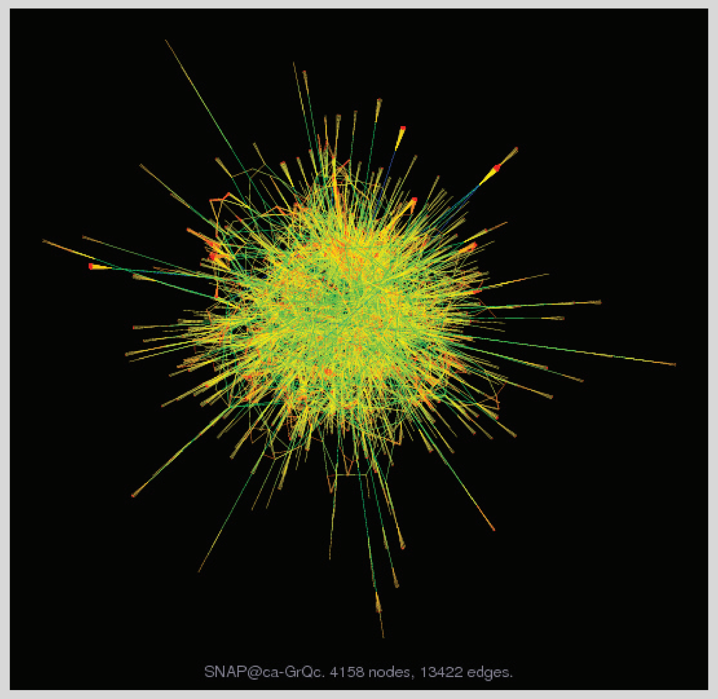
Network science has become increasingly popular over the last several years as people have realized that networks have the ability to represent the relationships or connections between any objects. While some networks are small and easy to gather information from, others can be very large. It can be very difficult and time consuming to map out these large networks if we collect data from all the nodes in the network.

Instead of examining all nodes, we seek to collect data incrementally from a portion of the network at a time to discover the whole network. This discovery occurs by successively placing **monitors** which can see a local portion of the graph. We then tested all of our algorithms on four different networks. Although there was no one algorithm that did best overall, we were able to see some of the strengths and weaknesses of each on various structures of networks.

Networks

The primary networks used to test algorithms were:

- Synthetic Networks:
  *Synthetic networks are sets of data made for the purpose of testing algorithms
  - → Erdos-Renyi
    - a random graph in generated by edges being placed between nodes using probability
  - → Barabasi-Albert
    - a random graph generated by preferential attachment
- Real Networks [3]:
  - → Facebook Combined
    - a social network from friend's lists on a subset of Facebook
  - → General Relativity on arXiv
    - a collaboration graph where nodes are authors and edges represent co-authorship



A visualization of the General Relativity Collaboration Network from SNAP at Stanford [3]

Algorithm Rules

The Basics:

- All of the Test Algorithms begin with a random start
- Monitors placed have the ability to see:
  - all of the edges adjacent to it
  - all of its neighbor nodes
  - the degrees of all of its neighbor nodes
- Every algorithm places monitors on 50% of the network

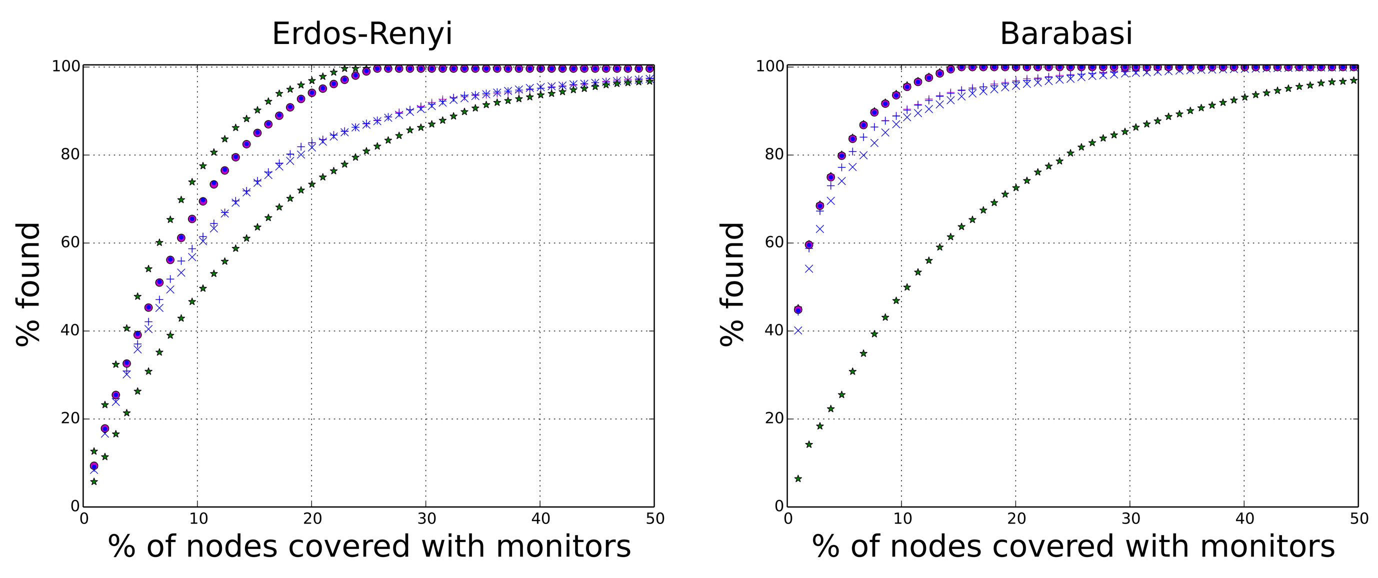
Bound Algorithms:

These algorithms were written to provide somewhat of an upper and lower bound for our test algorithms.

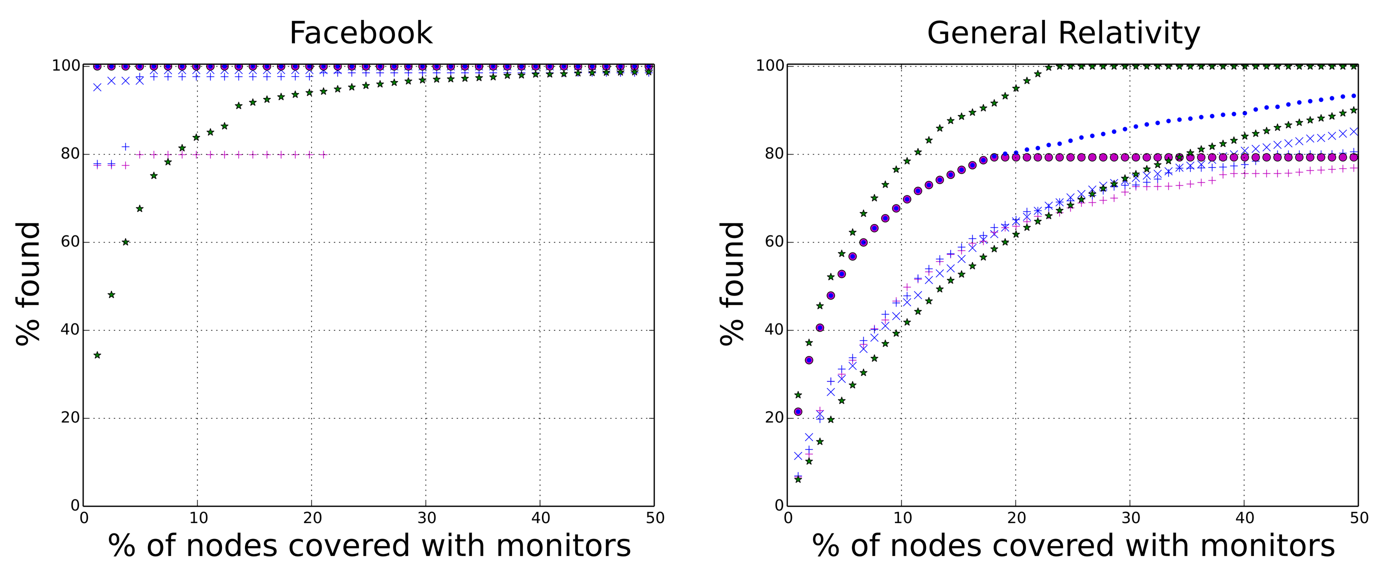
- **Random Placement Smart (RPS)**:
  - → Randomly places monitors on nodes
  - → No strategy, just random
  - → Will not place monitor on a node that already has a monitor
- **Upper Bound Discovery Nodes/Upper Bound Discovery Edges (UBDn/UBDe)**:
  - → Ideal algorithms for nodes and edges respectively
  - → Under the assumption that we can see the entire network
  - → Place monitor on node with the highest "fake degree"
  - → Used to approximate an "upper bound" on our discovery

Data & Results

Synthetic Networks: Nodes



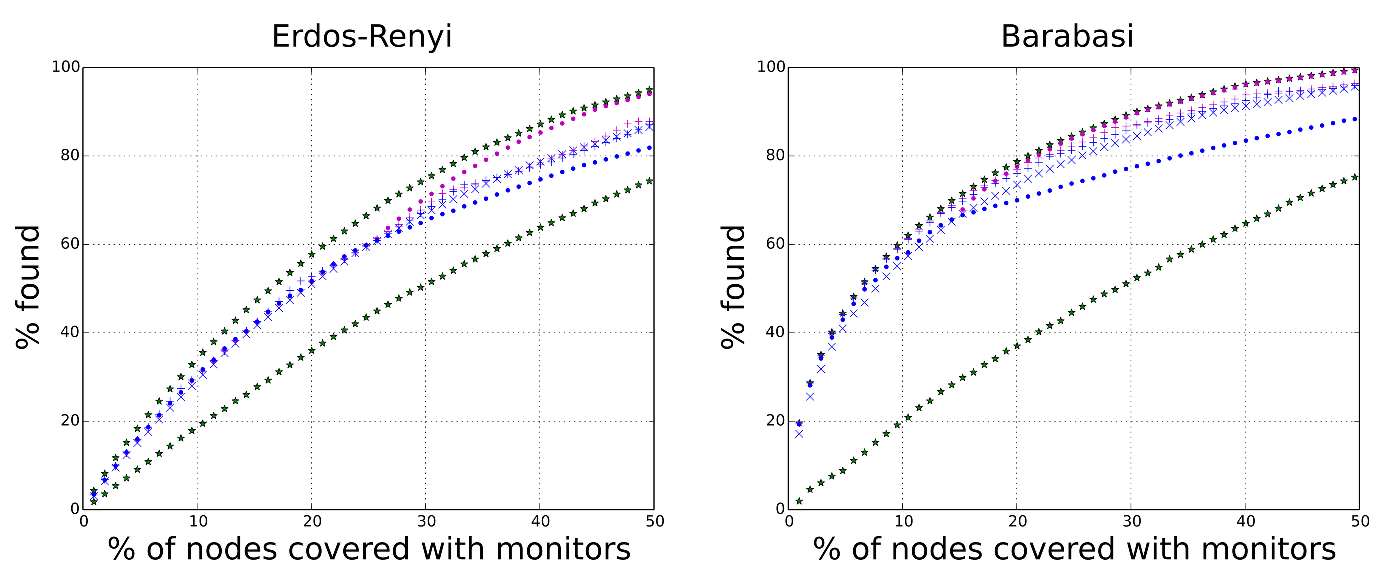
Real Networks: Nodes



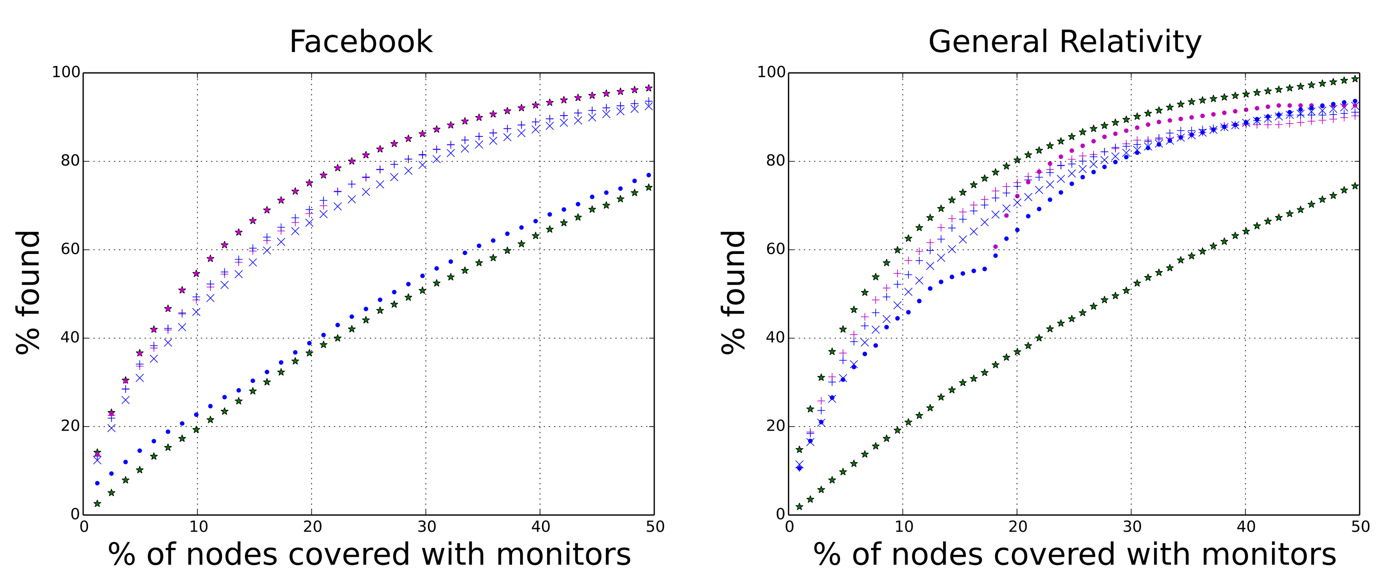
UBDn/e	HGD_LS	FDD	HC
RPS	HGD_LS_wR	FDD_wR	

UBDn is only graphed on the node graphs & UBDe is only graphed on the edge graphs

Synthetic Networks: Edges



Real Networks: Edges



Test Algorithms

The following are the 5 different algorithms we created along with a description of what exactly they do:

- **Hill Climbing (HC)**:
  - → Places monitor on neighbor of current monitor with highest degree
  - → If this neighbor already has a monitor, restarts to a seen node in the network with the highest degree
- **High Global Degree Least Seen (HGD_LS)**:
  - → Places monitor on any previously seen node with the highest degree
  - → If multiple nodes have the maximum degree, the node that has been seen the least is chosen for the monitor
- **Fake Degree Discovery (FDD)**:
  - → Places monitor on any previously seen node with highest "fake degree"
  - → The "fake degree" is defined as the degree of the node minus the number of neighbors it has that have already been seen by a previously placed monitor
- **High Global Degree Least Seen with Restart (HGD_LSwR)**
  Fake Degree Discovery w/ Restart (FDD_wR2):
  - → Same as HGD_LS and FDD respectively except that it has a random restart once two monitors in a row are placed with no additional discovery

Conclusions

Table 1

	Best Nodes	Best Edges	Best Overall
Erdos-Renyi	1. FDD_wR 2. FDD	1. FDD 2. HGD_LS	FDD
Barabasi-Albert	1. FDD 2. FDD_wR	1. FDD 2. HGD_LS	FDD
Facebook (subset)	1. FDD & FDD_wR	1. FDD 2. HGD_LS	FDD
General Relativity	1. FDD_wR 2. RPS	1. FDD_wR 2. HC	FDD_wR

Algorithms that incorporate in a restart are highlighted

- Table 1 shows which algorithms perform best on each of the four networks
- FDD appears to perform the best out of all these algorithms
- When FDD does not perform best, it is due to a restart being necessary
- We do not have an algorithm that runs exceptionally well on General Relativity
- Some algorithms do exclusively better on the synthetic graphs
  - → These synthetic graphs can be helpful for looking at patterns in graphs, but they might not be the most accurate representations of actual networks.

Future Work

- Speed up our algorithms so that they can run larger networks efficiently (as all of the networks shown here only had $\sim$ 5000 nodes)
- Create new algorithms that pick monitors by some value other than degree
- Create new algorithms that can run on directed networks in addition to undirected networks
- Run current algorithms on more graphs to see if there are any patterns of algorithms that perform better on certain types of networks

References & Acknowledgements

[1] Gera, Ralucca. "Network Science Support to the Department of Defense". June 2015. PowerPoint Presentation

[2] Newman, M.E.J. *Networks: An Introduction.* Oxford: Oxford UP, 2010. Print.

[3] Krevl, Andrej and Leskovec, Jure. "Stanford SNAP." Stanford Large Network Dataset Collection. N.p., June 2014. Web. 1 June 2015.

The data used in this work was funded by the MSEED NSF Award #1068346 and the Department of Defense

Special thanks to the MSEED Program for this opportunity and Nick Juliano for his contributions. An additional thanks goes to all those who worked on this project before us.