MULTIGRID APPROACH FOR MODELING NETWORKS

FIELDS INSTITUTE

A. "Sasha" Gutfraind   Lauren A. Meyers   Ilya Safro

University of Illinois at Chicago
University of Texas at Austin
Clemson University

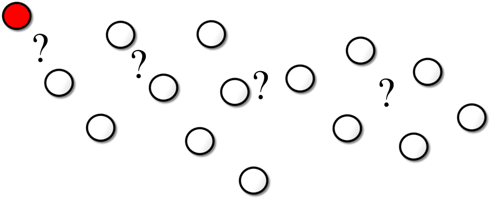
2014

Outline

Summary: The multiscale method (MUSKETEER) generates synthetic networks that match the properties of real networks.

MOTIVATION - THE MISSING DATA PROBLEM

1. Networks are the central part of many complex systems, e.g. infrastructure, social, neural systems
2. We need to evaluate ideas/methods/algorithms on them, & understand their structure
3. Limitations of empirical data:
    1. Difficult or Impossible to get
    2. Insufficient: want to show robustness on 10^2 to 10^6 networks



Methods for Network Modeling

1. Network model: Erdős-Rényi, Kronecker Graph, ERGM, Watts-Strogatz, Liu-Chung expected degrees, Barabási-Albert, etc.
2. Mechanistic model
3. Randomize empirical data
4. An application-specific topology generator: BRITE, INET, Tiers, GT-IGM, PLOD, GridG, GeNGe, etc.

New (5.):

Multiscale network generation (MUSKETEER)

Ref: "Multiscale Network Generation". Free and Open source. arxiv.org/abs/1207.4266

METHODS FOR NETWORK MODELING

1. Network model: Erdős-Rényi, Kronecker Graph, ERGM, Watts-Strogatz, Liu-Chung expected degrees, Barabási-Albert, etc.
2. Mechanistic model
3. Randomize empirical data
4. An application-specific topology generator: BRITE, INET, Tiers, GT-IGM, PLOD, GridG, GeNGe, etc.

New (5.):

Multiscale network generation (MUSKETEER)

Ref: "Multiscale Network Generation". Free and Open source. arxiv.org/abs/1207.4266

Multiscale Algorithms

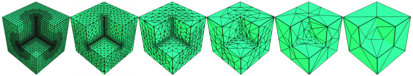
What is a **multiscale/multigrid algorithm**?

1. Iteratively *coarsen* i.e. reduce the number of variables in a problem:

$$L_0 \quad \rightarrow \quad L_1 \rightarrow \cdots \rightarrow \mathbf{L_k} \rightarrow \cdots \rightarrow L'_1 \rightarrow L'_0$$
$$\text{e.g.} \quad L_{i+1} = P^T L_i P$$

2. Solve in level k and then *refine* it back to level 0

- Strengths: $O(m)$ or $O(m \log m)$ performance for P or NP-hard problems
- Pitfalls: Enforcing constraints & Precision
- Very successful in large linear/nonlinear equation solvers

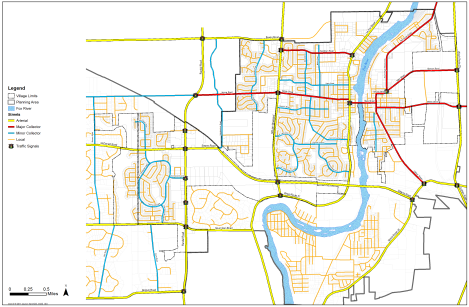


Ref: Knepley/UC - PETSc

Real Networks

Real Networks:

1. Organized hierarchically
   Refs: Ravasz & Barabasi
2. Levels are dissimilar
   Refs: Doyle et al.
3. Connections are usually local: low expansion, clustering, loops
   Ref: Barabasi, Spielman

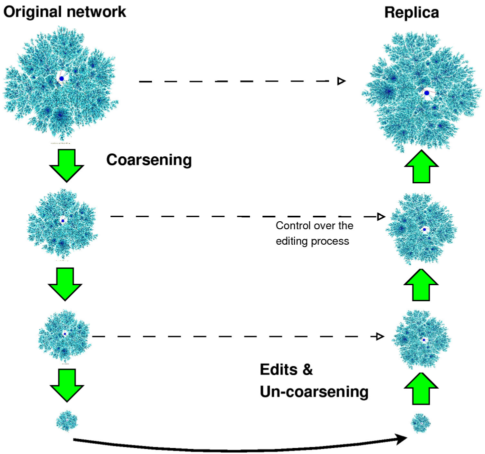


A Road Network

THE MULTISCALE APPROACH

The multiscale network modeling approach:

1. Generates a hierarchy of coarsened networks
2. Edits at any level of coarsening
3. Synthethic nodes are resampled
4. Synthetic edges preserve locality



Version 1.2 (Dec): Fast editing algorithm

APPROACH - 2

The central algorithm: ReviseGraph(G) function

1: $G_{i+1} \leftarrow \text{Coarsen}(G_i)$
2: $\tilde{G}_{i+1} \leftarrow \text{ReviseGraph}(G_{i+1})$
3: $G'_i \leftarrow \text{Interpolate}(\tilde{G}_{i+1})$
4: $\tilde{G}_i \leftarrow \text{EditEdgesAndNodes}(G'_i)$
5: $\tilde{G}_i \leftarrow \text{UserDefinedAdjustment}(\tilde{G}_i)$
6: **Return** $\tilde{G}_i$

- Editing does not specifically attempt to enforce properties like degree distribution or clustering
- Preservation of local and global graph properties emerges as an approximate invariant of the editing process

APPROACH - 2

The central algorithm: ReviseGraph(G) function

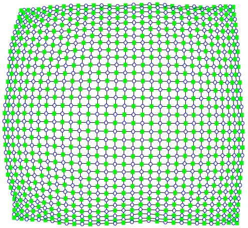
1: $G_{i+1} \leftarrow$ Coarsen(G_i)
2: $\tilde{G}_{i+1} \leftarrow$ ReviseGraph(G_{i+1})
3: $G'_i \leftarrow$ Interpolate$(\tilde{G}_{i+1})$
4: $\tilde{G}_i \leftarrow$ EditEdgesAndNodes(G'_i)
5: $\tilde{G}_i \leftarrow$ UserDefinedAdjustment$(\tilde{G}_i)$
6: **Return** $\tilde{G}_i$

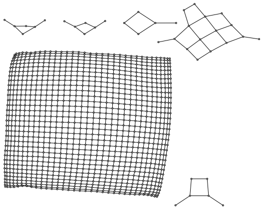
- Editing does not specifically attempt to enforce properties like degree distribution or clustering
- Preservation of local and global graph properties emerges as an approximate invariant of the editing process

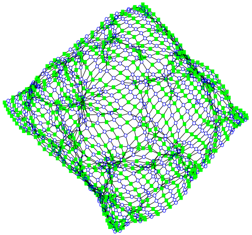
NETWORKS

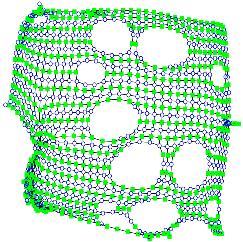
Let's make some networks ...

Preservation of Hidden Properties



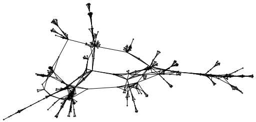


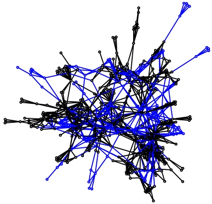




EXAMPLE: COAUTHORSHIP

Collaboration network (Newman): GCC 379 nodes

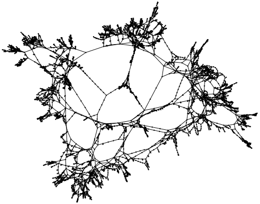


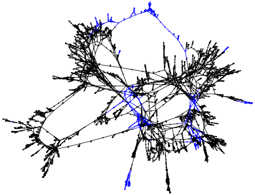


growth rate: nodes [0, 0.3]; edges:[0, 0.1]

Example: Power Grid

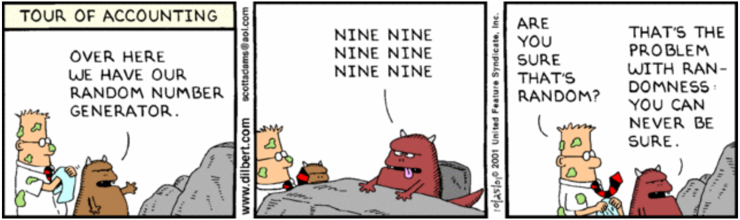
Western Interconnection - a power grid with 4941 nodes





edit rate: nodes [0, 0.1]; edges:[0, 0.1]

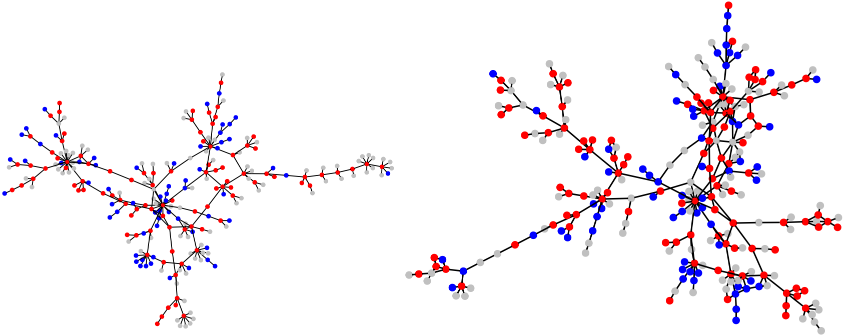
Evaluation of Random Networks



Quality of Random Networks - 1

Experimental simulation

- Level 0 edits: 8% nodes, 8% edges
- Level 1 edits: 7% nodes, 7% edges
- Generally, the choice of edit rates is based on the problem

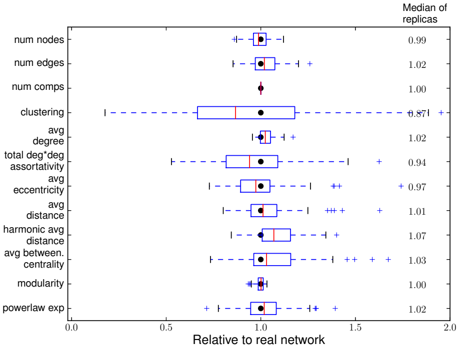


Colorado Springs HIV (left) and replica (right)

Ref: Potterat et al.

Quality of Random Networks - 1

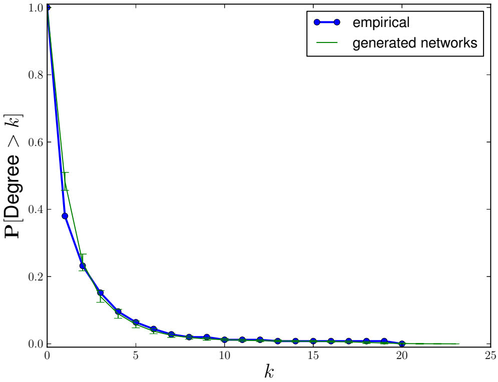
FIGURE: Colorado Springs Network



Diversity: 30% of nodes and 60% of edges are new or removed

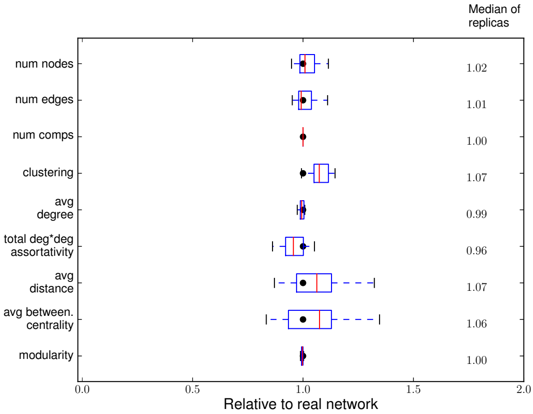
Quality of Random Networks - 2

FIGURE: Colorado Springs Network



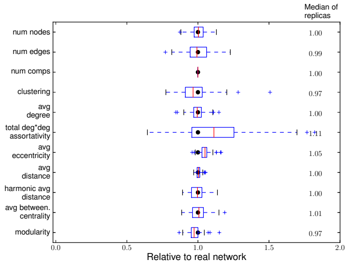
Quality of Random Networks - 3

FIGURE: Western Interconnection (Watts & Strogatz)

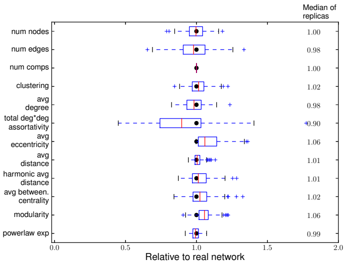


Quality of Random Networks - 4

Erdős-Rényi template

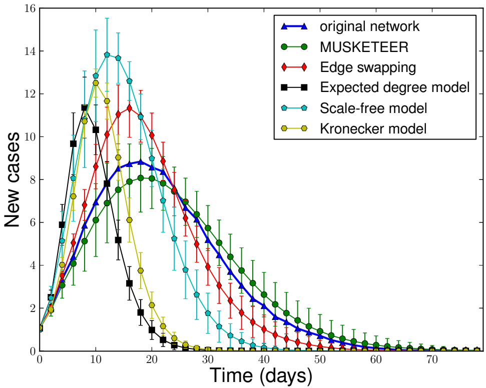


Barabási-Albert template



Dynamics on Synthetic Networks

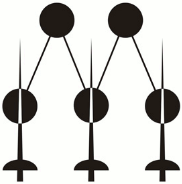
FIGURE: SEIR cascade on Colorado Springs Network



Select use stories

S Leyffer, I Safro

- Developed an algorithm for blocking cyber attacks on large networks
- Replicas helped discover implementation errors
- Replica data provide performance evaluation



M Bergner, ME Lübbecke, J Witt

- Investigate the “packed cuts” problem
- Developed a new Branch-Price-and-Cut Algorithm
- Replica data provide performance evaluation

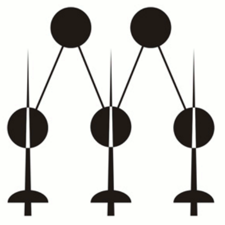
Open problems

- Fundamental limitations:
  What are some of the fundamental limitations of multiscale generation?
- Degree distribution:
  Could the editing process be designed to preserve the degree distribution?
- Auto-tuning:
  Find the best editing structure for each network?

Summary & Evaluation

Multiscale Network Modeling

- Synthetic data with realistic properties
- Controlable: fine and global editing; size expansion
- Suitable for many types of networks
- $O(m)$ running time



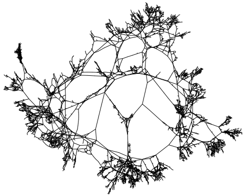
agutfrai@uic.edu

G, Meyers and Safro. "Multiscale Network Generation".
www.cs.clemson.edu/~isafro/musketeer

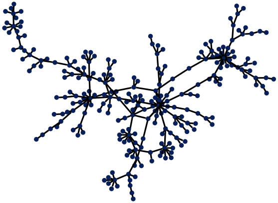
Thanks

DTRA & Los Alamos LDRD program, Argonne Cybersec LDRD, NIH/MIDAS; many colleagues

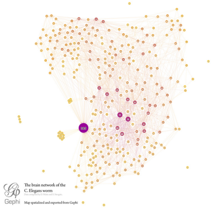
Network Science



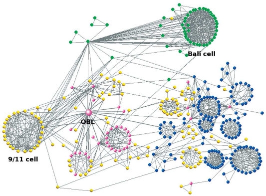
Power grid (Watts and Strogatz)



Colorado Springs HIV (Potterat et al.)

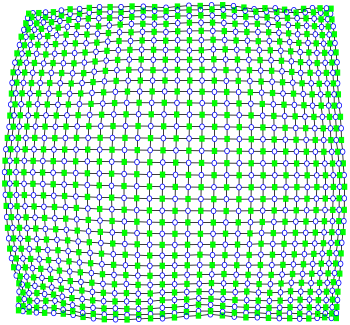


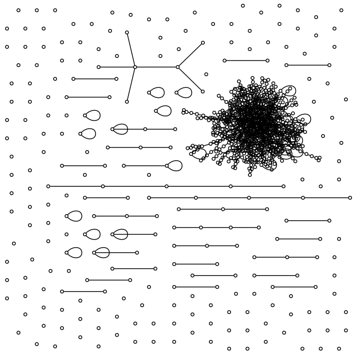
C. elegans brain (White)



Al-Qaida (Xu, Sageman et al.)

Replication with a Random Kronecker Graph



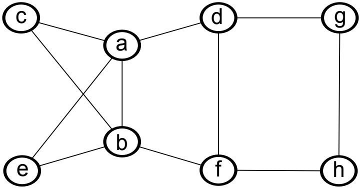


Key notions of graph theory

Definition

Graph is the pair, (V, E) where V is a set called *nodes,* and E are unordered pairs (i, j) called *edges* such that $(i, j) \in V \times V$ and $i \neq j$.

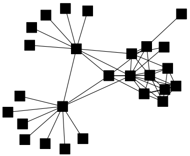
- Annotation: numbers, labels on nodes and/or edges
- Degree of node u = the number of neighbors of u
- Clustering coefficient, modularity, distance



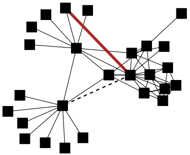
The Editing Process: Edges

To create a new edge (u, v)

- **Measure**: $d_2(i, j) =$ distance of two neighbors through the shortest path not through their common edge.
- Estimate $\mathbb{P}[d_2]$.



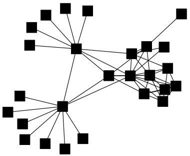
1. Sample x from the distribution $\mathbb{P}[d_2]$
2. Randomly select u, and find node v at distance x from u
3. Pick a random edge, **measure** the number of internal connections, and create the same number of connection between u and v.

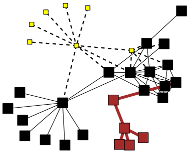


The Editing Process: Nodes

To create a new node

1. Take a random node from original network & **measure** its degree *D*
2. The new node *u* will have *D* neighbors
3. Select the first neighbor at random & the remaining neighbors by the edge creation process above
4. Pick a random node *w* and **copy** its aggregate into *u*





Applications of synthetic data

Synthetic data are needed to

- Model networked populations
- Simulate “what-if” scenarios
- Compensate for missing/insufficient data
- Anonymize data

The Data Problem for Networks

Want: synthetic dataset $\Gamma = \{G_t\}$, such that:

1. Large: $|\Gamma| \gg 1$
2. Diverse: $d(G,H) > \varepsilon$ for all $G,H \in \Gamma$
3. Realistic: for all $q \in Q,\ G \in \Gamma$:

$$\mathbb{P}[\|q(G) - W_q\| < T] > p$$

- Realism could be measured structurally, e.g. clustering coefficient
- Emergent properties are also important for realism

The Data Problem for Networks

Want: synthetic dataset $\Gamma = \{G_t\}$, such that:

1. Large: $|\Gamma| \gg 1$
2. Diverse: $d(G,H) > \varepsilon$ for all $G,H \in \Gamma$
3. Realistic: for all $q \in Q$, $G \in \Gamma$:

$$\mathbb{P}\left[\|q(G) - W_q\| < T\right] > p$$

- Realism could be measured structurally, e.g. clustering coefficient
- Emergent properties are also important for realism

ABSTRACT

In the talk I will introduce a flexible strategy for modeling networks using ideas inspired by multigrid methods. The strategy, termed MUSKETEER, is to start from a known network dataset, perform a series of mappings that repeatedly coarsen and later repeatedly uncoarsen the network, while applying perturbations to create diversity. Using examples from several domains, I will show that MUSKETEER can generate diverse ensembles of networks, including their edge and node labels. Statistical analysis shows that MUSKETEER also achieves greater realism than most network modeling strategies.
Bio: A. "Sasha" Gutfraind - University of Illinois at Chicago Sasha Gutfraind received a Bachelor's and a Master's from the University of Waterloo in Applied Mathematics and a Ph.D. from Cornell University. He develops mathematical models to illuminate problems in complex networks, public health and security using methods from the theories of complex systems, mathematical optimization and dynamical systems. Prior to coming to UIC, he worked at Los Alamos National Laboratory and at the University of Texas at Austin