Are we there yet?
When to stop a Markov chain while generating random graphs

Ali Pinar, Jaideep Ray, and C. Seshadhri

Sandia National Labs

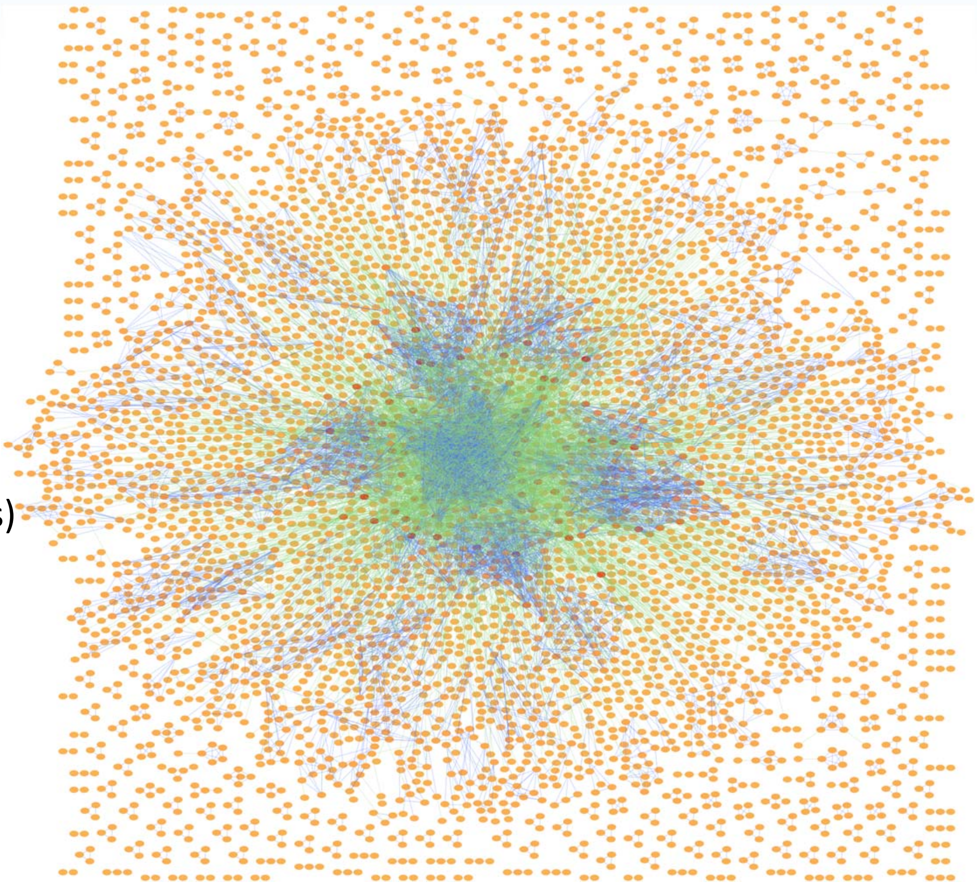
U.S. Department of Energy
Office of Advanced Scientific Computing Research

U.S. Department of Defense
Defense Advanced Research Projects Agency

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Why generate random graphs?

- Enable sharing of surrogate data
  - Computer network traffic
  - Social networks
  - Financial transactions
- Statistical analysis
  - Sample uniformly from a specified space
- Testing graph algorithms
  - Scalability
  - Versatility (e.g., vary degree distributions)
  - Characterizing algorithm performance
- Insight into…
  - Generative process
  - Community structure
  - Comparison
  - Evolution
  - Uncertainty



Block Two-Level Erdös-Rényi (BTER) graph; image courtesy of Nurcan Durak.

Markov Chains: common method to generate random graphs

- For this talk, a Markov chain (MC) is a graph whose nodes are realizations of a graph with desired features
  - Normally, MC graph is never constructed
  - We generate its vertices, as we walk on the graph
- A random walk on an MC (with the right features) can yield a random graph.
- To generate a random graph using an MC
  - Find an arbitrary node of the MC
  - Take a loooong random walk
  - You will arrive at a uniform random vertex of the graph
    - given that you have a ``good" MC
- Challenges
  - Generating a graph with given properties
  - Rewiring a graph to preserve desired features
  - Patience

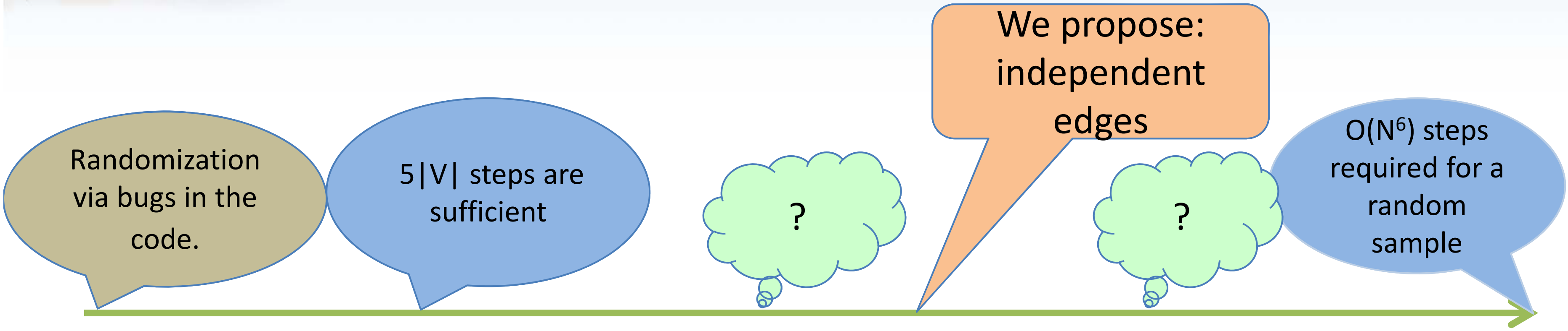
Math can prove convergence, but cannot grant you patience

Source: http://metsmerizedonline.com/wp-content/uploads/2013/02/Are-We-There-Yet.jpg

Can we find principled and practical metrics to think about convergence?

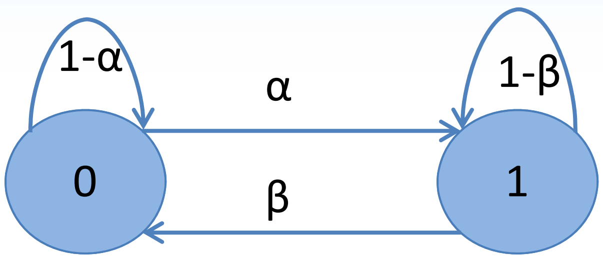
- In theory, we need to prove the MC eventually produces a random graph.
- In practice, bounds for convergence may be impractical or nonexistent.
- Practitioners use unprincipled methods.
  - e.g., 10K steps on the MC
- Interpretations of statistical tools may be hard.
  - What does Gelman Rubin test mean from a graphs perspective?

Can we find principled and practical metrics for convergence?



- What is a mathematically sound definition of “random enough?”
- Goals: practical, sound, and interpretable.
- An imperfect analogy:
  - To solve *Ax=b*, we do not compute $A^{-1}b$, we compute an *x*, that yields a small residual for *Ax-b*.
  - We learn how to deal with this imperfection.

Testing independence of edges



State 0: edge is absent

State 1: edge is present

α: probability that the edge will be inserted

β: probability that the edge will be deleted

$$T = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$$

T: transition matrix of the edge

- Assume the addition/deletion of an edge can be approximated as a Markov process.
- The full Markov chain (MC) can be approximated as a collection of smaller Markov chains.
- Convergence of the smaller MCs is a necessary condition for convergence of the full MC.

Convergence of smaller Markov chains

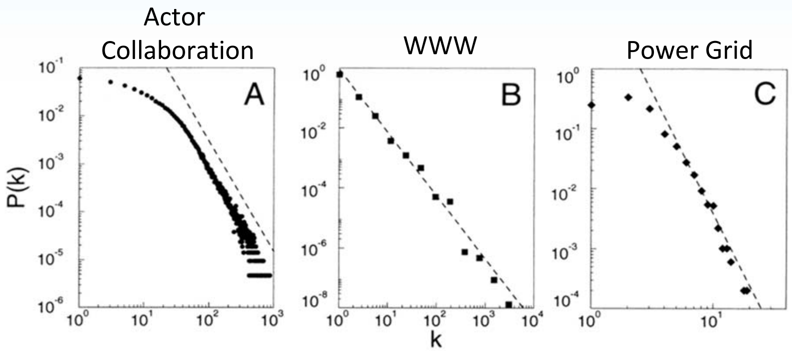
- Eigenvalues of *T* are *1* and $1 - (\alpha+\beta)$
- Eigenvalues form a basis, so initial state *v* can be written as $v=c_1e_1+c_2e_2$.
- After N iterations, we have

$$p = T^N v = c_1 e_1 + c_2 (1 - (\alpha + \beta))^N e_2$$

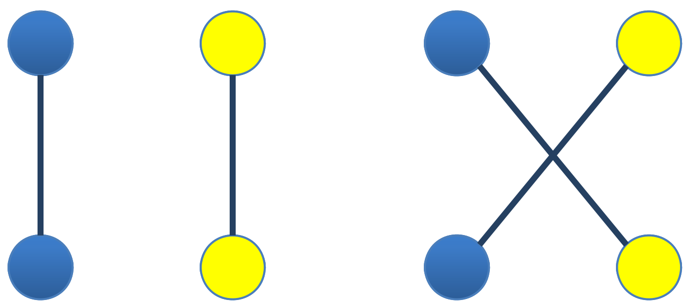
- The second term decays and p converges to c_1e_1, which indicates the probability the edge is present/absent in a random graph.
- For tolerance *ε,* the number of iterations required, *N*, is

$$N = \ln(1/\varepsilon)/(\alpha+\beta)$$

Preserving the degree distributions



A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. Science, 286(5349):509-512, 1999.

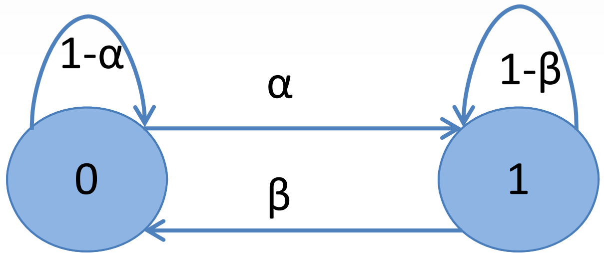


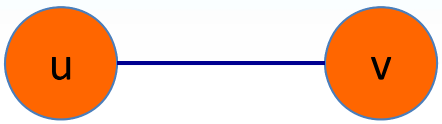
Choose two Random edges

Swap them

- Degree distribution is like a histogram of degrees.
- It is one of the critical features that distinguish real graphs from arbitrary sparse graphs.
- Rewiring scheme has long been used to perturb graphs while preserving the degree distribution.
  - Converges in $O(|E|^6)$-time.
- Havel and Hakimi described the first algorithm to construct a graph with a given degree distribution.

Transition matrix for preserving degree distribution





d_u: degree of vertex u

m: total number of edges

α: probability that the edge will be inserted

β: probability that the edge will be deleted

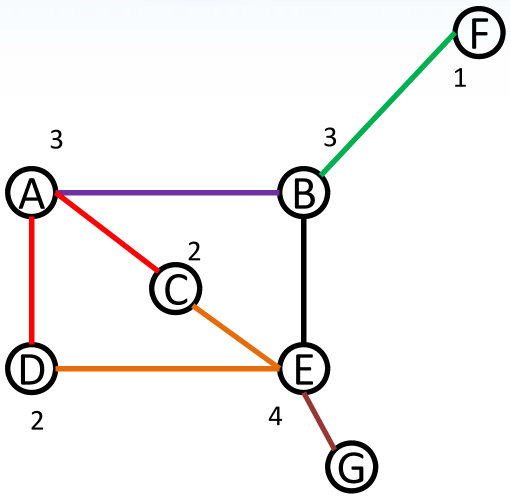
$$N = \ln(1/\varepsilon)/(\alpha+\beta)$$

$$\alpha = \frac{d_u d_v}{2m^2} \qquad \beta = 1-(1-\frac{1}{m})^2$$

$$\alpha+\beta \geq \frac{2}{m}$$

To generate a graph with independent edges with a specified degree distribution we need $N = \frac{m}{2}\ln\varepsilon^{-1}$

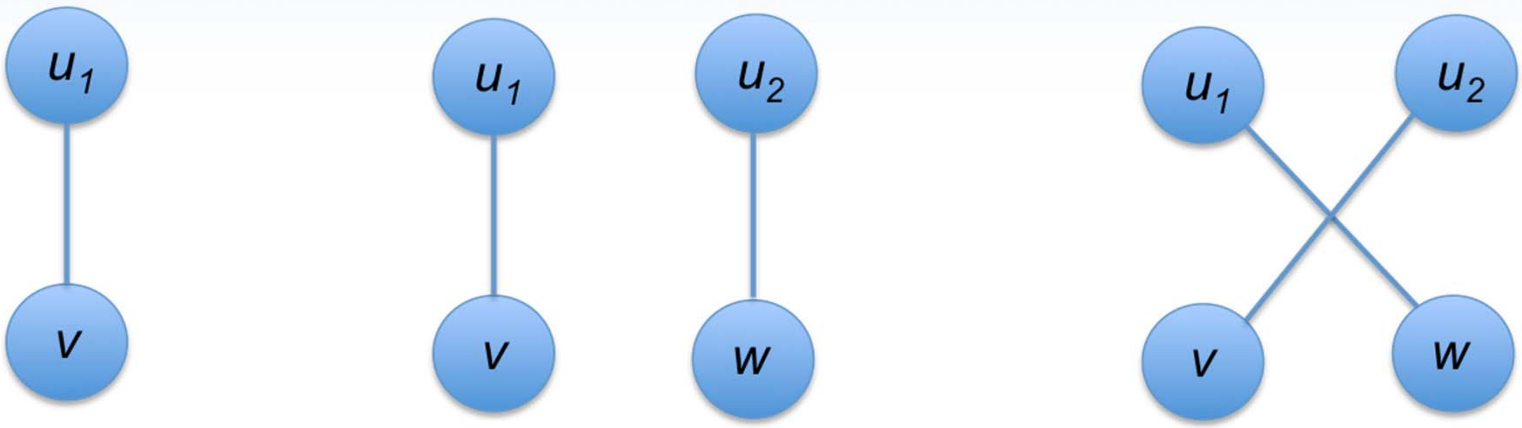
Joint Degree Distribution



Degree	1	2	3	4
1	0	0	1	1
2	.	0	2	2
3	.	.	1	1
4	.	.	.	0

- Joint Degree Distribution (JDD) specifies the number of *edges* between vertices of specified degrees.
- *JDD provides more information abot a graph.*
  - *The degree distribution is implicitly defined by JDD.*
- *Work on JDD is more recent and sparse.*

Preserving JDD



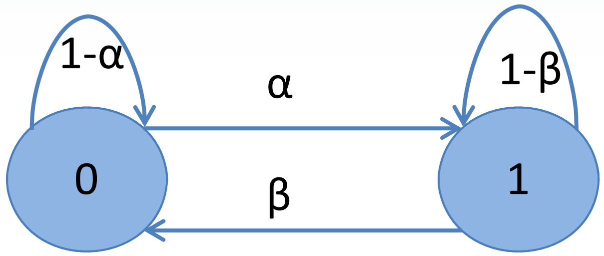
Step 1: Pick an edge (u_1,v), and pick one of its vertices, e.g., u_1

Step 2: Pick another edge (u,w), such that $d(u_1)=d(u_2)$ or $d(u_1)=d(w)$

Step 3: Swap edges

- This Markov chain can be used to construct uniformly random instances of a graph with a specified degree distribution.
- No theoretical bounds on convergence.
- A graph with a specified (feasible) joint degree distribution can be constructed in linear time.
- Stanton & P., *ACM J. Experimental Algorithmics, 2012*

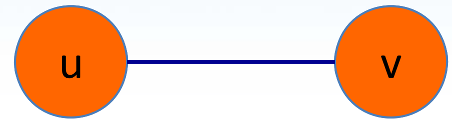
Transition matrix for preserving degree distribution



α: probability that the edge will be inserted

β: probability that the edge will be deleted

$$N = \ln(1/\varepsilon)/(\alpha+\beta)$$



d_u: degree of vertex u m: # edges

$f(d_u)$: #vertices of degree d_u

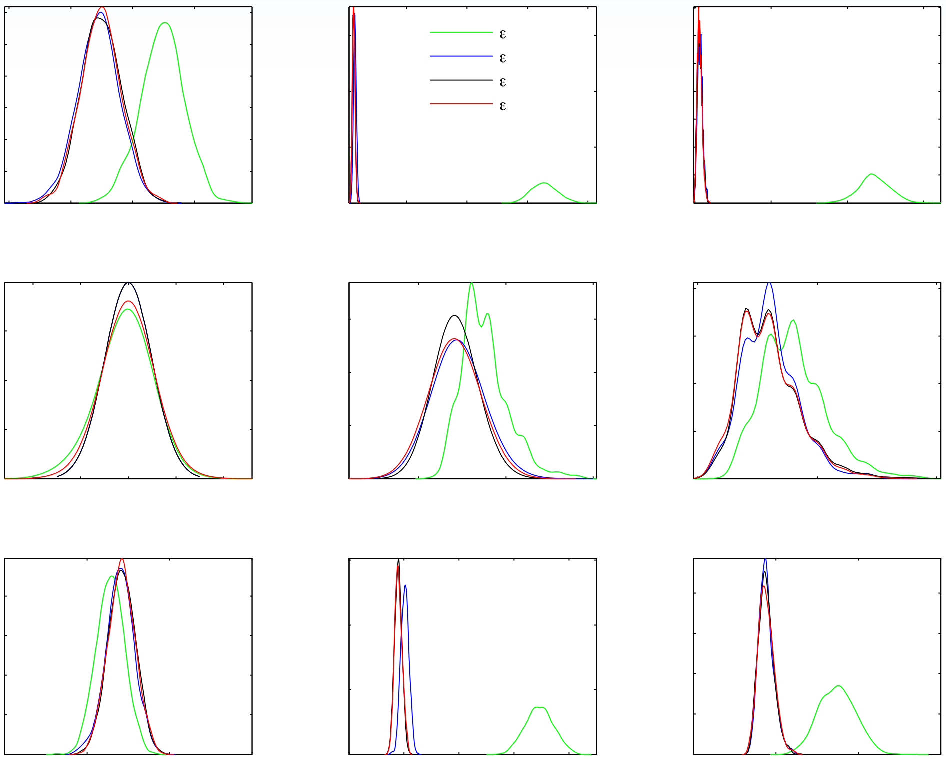
$J(d_u, d_v)$: #edges between d_u and d_v

$$\beta = \frac{1}{m} + \frac{f(d_u)-1}{2mf(d_u)} + \frac{f(d_v)-1}{2mf(d_v)}$$

$$\alpha \cong \frac{2J(d_u, d_v)}{mf(d_u)f(d_v)} \qquad \alpha+\beta \ge \frac{1}{m}$$

To generate a graph with independent edges with a specified degree distribution we need $N = m \ln \varepsilon^{-1}$

How does edge practice wok in practice?



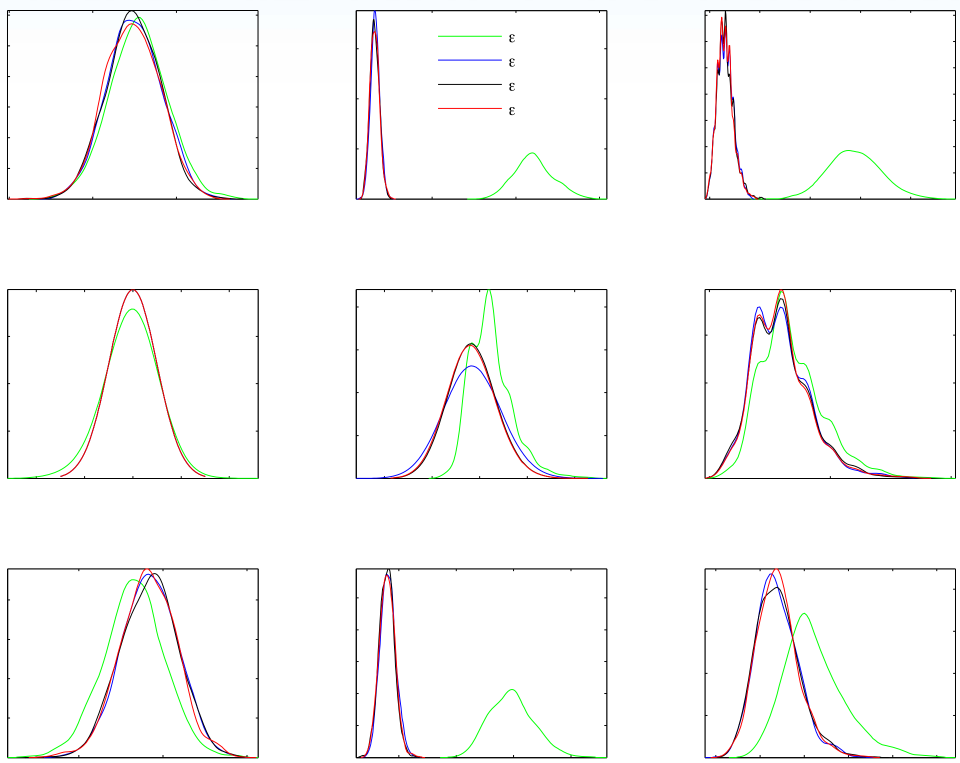
C. Elegans
297 vertices,
4296 edges

Netscience
1461 vertices,
5484 edges

Power
4941 vertices,
13188 edges

- Preserving degree distribution
- Errors correspond to 0.5|E|, 2.5|E|, 5|E|, and 7.5|E| iterations
- 1000 graphs generated starting from the original
- 5|E| iterations seem to be sufficient.

Edge independence in practice



C. Elegans
297 vertices,
4296 edges

Netscience
1461 vertices,
5484 edges

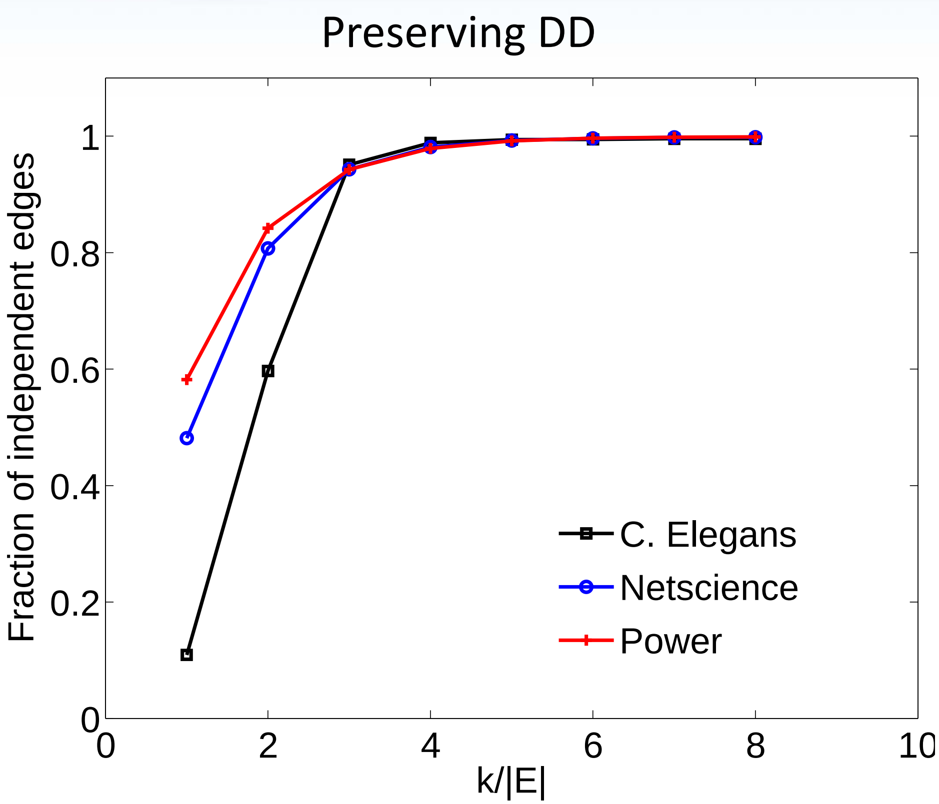
Power
4941 vertices,
13188 edges

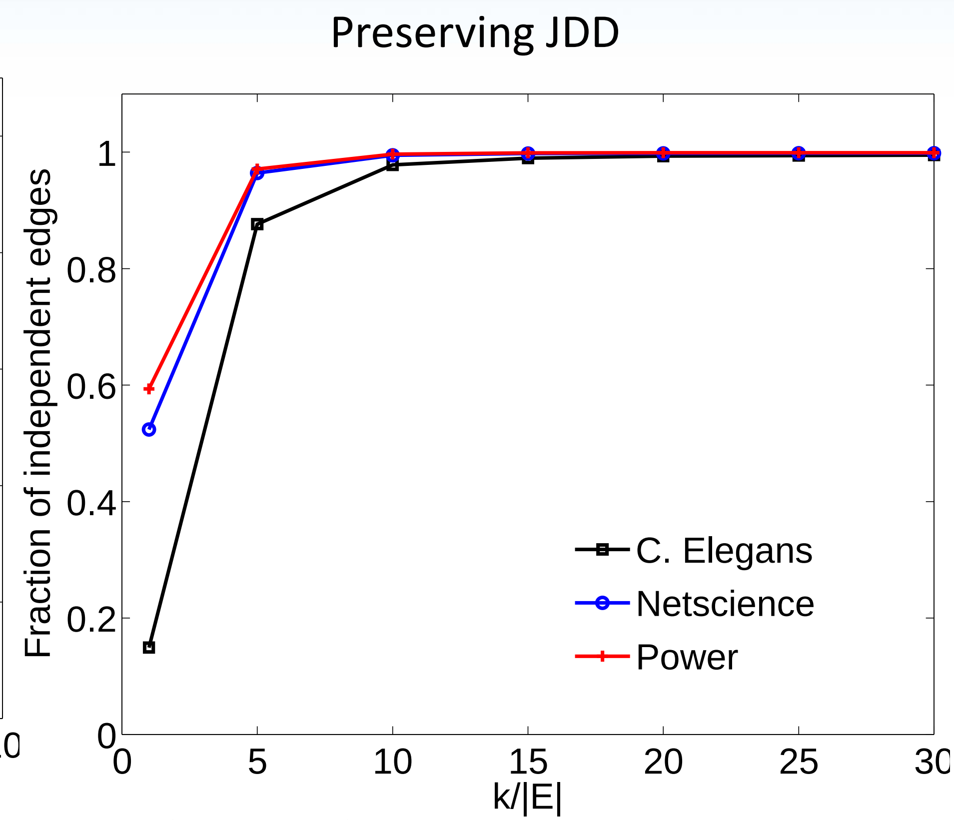
- Preserving JDD
- Errors correspond to |E|, 5|E|,10|E|, and 15|E| iterations.
- 1000 graphs generated starting from the original
- 10|E| iterations seem as sufficient.

Alternative way to measure independence

- Does knowing the current status (present/absent) of an edge help us predict its status in the next iteration better?
  - How about its status after 10 steps? 20 steps?
  - How many steps will be sufficient for the prediction to fail?
- The point we fail, the edge becomes independent
- A popular method in statistics
- Method:
  - Generate a long sequence
  - Fit a model to predict k steps ahead
  - Thin this sequence with smallest k for which the prediction fails

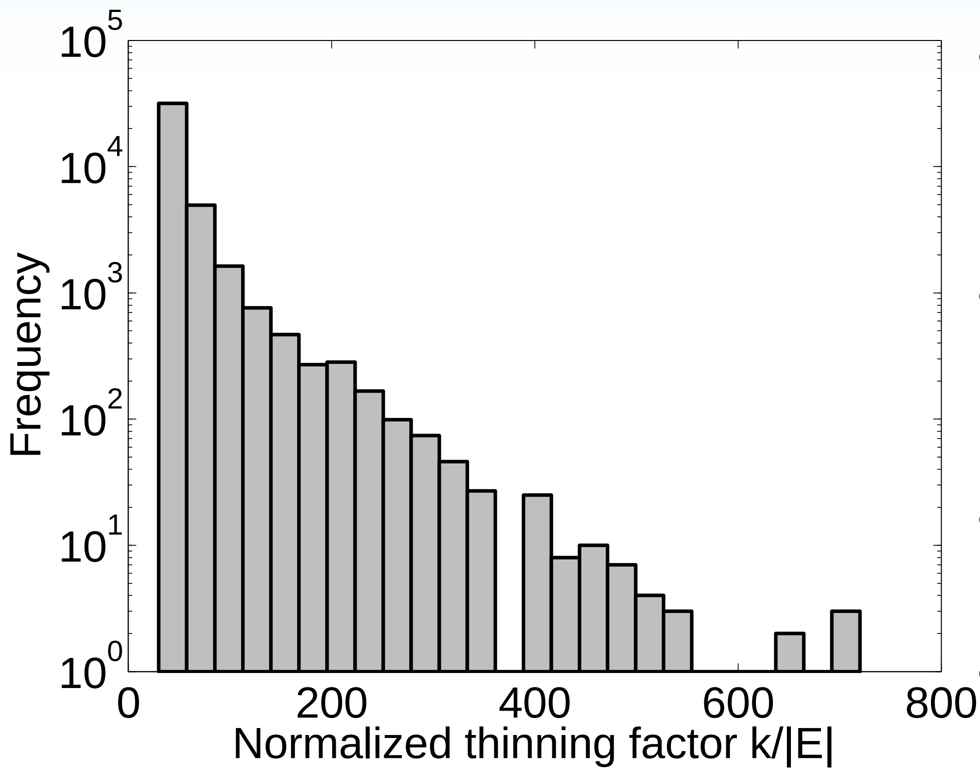
Edges become independent rapidly





- All potential edges are included in the analysis.
- Only a few remain after 7.5|E| and 15|E| iterations for preserving DD and JDD, respectively.

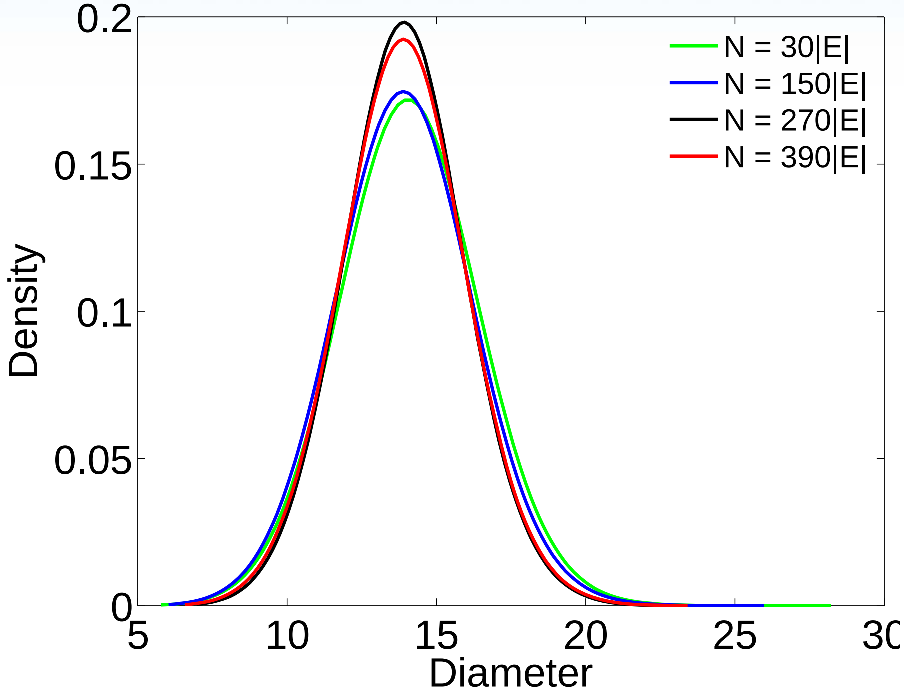
Some edges are tougher than others

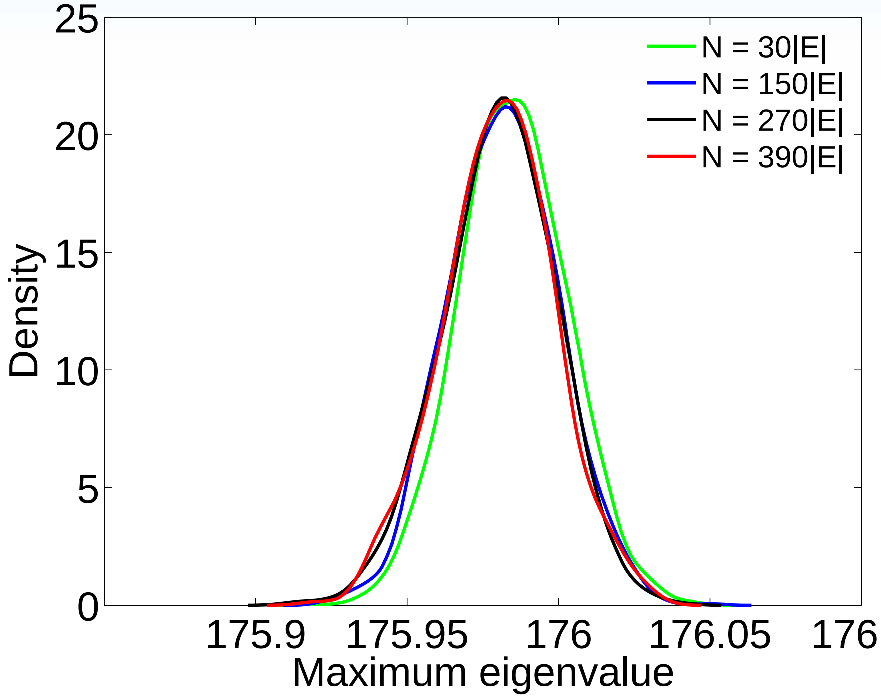


Soc-Epinions1
75879 vertices,405740 edges

- Preserving JDD on Soc-Epinions
  - Edges are sampled down to 10%.
- After 30|E| iterations 90% of the edges become independent.
- Most of the remaining ones are close to independence.
- There are a few outliers.

Diminishing returns for extra steps





- Preserving JDD on soc-epinions1
- Distributions are very similar

Conclusions

- Generating uniformly random instances of a graph with given properties is a fundamental problem in graph analysis.
- Markov chains are commonly used for this purpose, but guaranteeing/testing their convergence is a challenge.
- We proposed to use
  - edge independence as a practical metric.
  - smaller Markov chains for presence/absence of edges as a guide.
- We showed how the method applies to DD and JDD preserving MCs.
- Empirical studies on several graphs validated the approach.
- We are not guaranteeing convergence of the chain, but providing a metric that quantifies what is satisfied.
  - Results should be interpreted accordingly.
- The same approach can be used to guarantee independence of a bigger structures.

A new workshop

- SIAM Workshop on Network Science
- Dates: July 7-8, 2013
- Place: San Diego, CA
- Co-located with SIAM Annual Meeting
- There will be a call for posters
- Contact:
  - Ali Pinar (apinar@sandia.gov), Sandia National Labs
  - Madhav Marathe (mmarathe@vbi.vt.edu), Virginia Tech

Relevant Publications

- J. Ray, A. Pinar, and C. Seshadhri, ``A stopping criterion for Markov chains when generating independent random graphs," arXiv:1210.8184.
- J. Ray, A. Pinar, and C. Sehadhri, Are we there yet? When to stop a Markov chain while generating random graphs," Proc. WAW 12. .
- I. Stanton and A. Pinar, "Constructing and uniform sampling graphs with prescribed joint degree distribution using Markov Chains," ACM Journal on Experimental Algorithmics, Vol. 17, No. 1, 2012.
- I. Stanton and A. Pinar, "Sampling graphs with prescribed joint degree distribution using Markov Chains," ALENEX'11.