Network Reliability: Approximation Algorithms

Elizabeth Moseman

in collaboration with
Isabel Beichl, Francis Sullivan

Applied and Computational Mathematics Division
National Institute of Standards and Technology
Gaithersburg, MD

March 30, 2012

NIST

The Problem

- Network Reliability
- Motivation

Single Variable Case

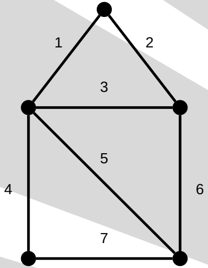
- Monte Carlo Markov Chain (MCMC)
- Sequential Importance Sampling (SIS)
- Improving Computational Efficiency

Multi-variate Case

- Subgraph Search Tree
- Tutte-like Search Tree
- Comparing the Methods

Future Work

Definitions



A graph G (or network) is a pair of sets (V, E).
A subgraph is a subset of the vertices and edges.
A spanning subgraph contains all the vertices.
A connected subgraph has paths between all vertices.

Problem Statement

Define $R(G;p)$ as the probability of a network remaining connected when edges are reliable with probability p.

Goal: Calculate $R(G;p)$.

When p is constant for every edge, we have

$$R(G;p) = \sum_{k=0}^{m-n+1} f_k p^{m-k}(1-p)^k$$

where f_k is the number of connected spanning subgraphs of G with $m-k$ edges. In this case, it is sufficient to calculate the values f_k for every k. In the more general case, such coefficients do not exist.

Motivation

- Develop measurement science for massive networks.
- Measure the reliability of infrastructure networks
  - Power grid: probability of getting power to all consumers.
  - How much reliability will be improved with incremental network changes.
- Exact computation is prohibitively expensive.
- Improved computational efficiency of Monte Carlo methods.
  - Supercomputers everywhere are running MCMC processes.

Monte Carlo Markov Chain

- Method of sampling from a large sample space without knowing the whole sample space.

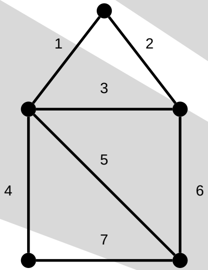
- Based on making moves inside the sample space.

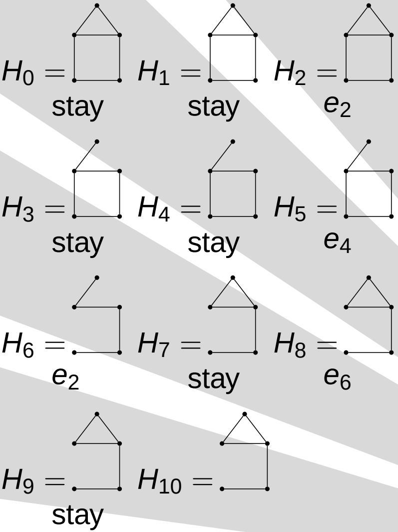
Monte Carlo Markov Chain

Currently at subgraph H_i.

With probability $\frac{1}{2}$, set $H_{i+1} = H_i$.
Select $e \in E$ uniformly at random.
if $e \in H_i$ and $H_i - \{e\}$ is connected **then**
Set $H_{i+1} = H_i - \{e\}$.
else if $e \notin H_i$ **then**
Set $H_{i+1} = H_i + \{e\}$.
else
Set $H_{i+1} = H_i$
end if

Example


1
2
3
4
5
6
7


$H_0 =$ stay
$H_1 =$ stay
$H_2 =$ e_2
$H_3 =$ stay
$H_4 =$ stay
$H_5 =$ e_4
$H_6 =$ e_2
$H_7 =$ stay
$H_8 =$ e_6
$H_9 =$ stay
$H_{10} =$

Monte Carlo Markov Chain

Currently at subgraph H_i.

With probability $\frac{1}{2}$, set $H_{i+1} = H_i$.
Select $e \in E$ uniformly at random.
if $e \in H_i$ and $H_i - \{e\}$ is connected **then**
Set $H_{i+1} = H_i - \{e\}$ with probability
$\min\{1, \mu\}$.
else if $e \notin H_i$ **then**
Set $H_{i+1} = H_i + \{e\}$ with probability
$\min\{1, 1/\mu\}$.
else
Set $H_{i+1} = H_i$
end if

fugacity

Monte Carlo Markov Chain

This yields a steady state distribution π_μ where

$$\pi_\mu(H) = \frac{\mu^{m-|H|}}{Z(\mu)}$$

where

$$Z(\mu) = \sum_{k=0}^{m-n+1} f_k \mu^k.$$

Problems with MCMC

- **Mixing Time** is the number of steps that must be taken before the state distribution is close enough to the steady state.
  Previous Solution: If it's not enough, take more steps.
- **Sample size** is the number of samples to take to get a good estimate of whatever is being measured.
  Previous Solution: Get many, many more samples than required.
- **Fugacity** is the value of μ used in the algorithm. Different fugacities explore different sections of the sample space.
  Previous Solution: Guess values, and pick more if parts of the sample space are not explored.

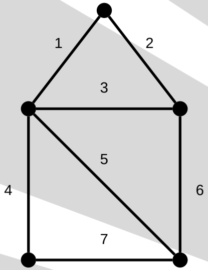
Sequential Importance Sampling

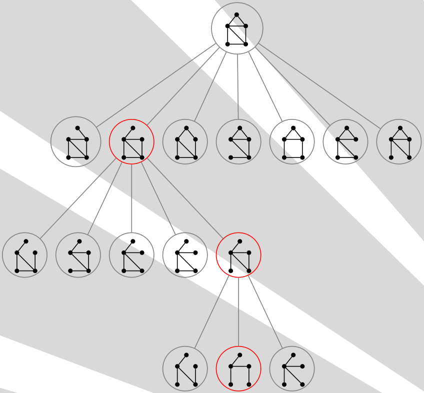
Based on previous work by Beichl, Cloteaux, and Sullivan.

- Uses Knuth's method of estimating the size of a backtrack tree. $\sum f(X) = E(f(X)p(X)^{-1})$.
- Form a tree with a subgraph at each node.
- Children are subgraphs with one edge removed.
- To estimate the number of subgraphs
  - Start with the whole graph.
  - Take out one edge at a time, without disconnecting.
  - Note the number of choices at each step.

Example

$$a_1 = 7$$
$$a_2 = 5$$
$$a_3 = 3$$
$$a_4 = 0$$





$$f_0 = 1$$
$$f_1 = 7$$
$$f_2 = \frac{7 \cdot 5}{2} = 17.5$$
$$f_3 = \frac{7 \cdot 5 \cdot 3}{3!} = 17.5$$

Actual Values:

$$f_1 = 7$$
$$f_2 = 19$$
$$f_3 = 21$$

Problems with SIS

- **Sample size** How fast does the average converge? On many graphs, it appears to converge very quickly, but there are pathological examples where is doesn't.
- People don't use this method. (We're trying to solve this by telling them about it.)

Using SIS to speed up MCMC

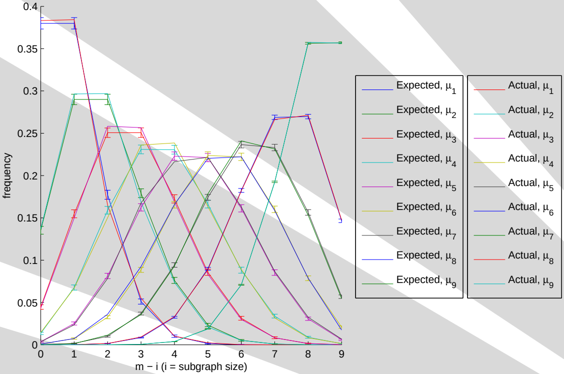
How can we use these methods together and make it more efficient?

- Run SIS first.
- Use the SIS results to select fugacity, calculate mixing time, and bound the sample size for use with MCMC.

Fugacity

- Fugacity changes resulting steady state distribution, indicating which area of the sample space (which subgraphs) we are exploring.
- Optimal fugacity of $\mu = f_i / f_{i+1}$ causes subgraphs of size $m - i$ and $m - i - 1$ to be equally likely, all other sizes less likely.
- Idea: Estimate f_i and f_{i+1} from SIS.

Calculated Fugacities



Fugacity chosen appropriately: Sample with fugacity μ_i gives a high proportion of sample subgraphs with $m - i$ edges. (As predicted)

Aggregation

- The transition matrix of the Markov Chain is stochastic matrix containing the probabilities of transitioning between each state (subgraph).
- There are too many states, so calculating the transition matrix exactly is prohibitively expensive.
- To reduce the number of states, we combine states that are “similar” in a process called aggregation.
- In this case, we are recording subgraph size, so we combine all subgraphs of the same size into one state.

Mixing Time

- Aggregated transition matrix:

$$\begin{bmatrix} 1-A_0-B_0 & A_1 & 0 & \cdots & 0 \\ B_0 & 1-A_1-B_1 & A_2 & \cdots & 0 \\ 0 & B_1 & 1-A_2-B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1-A_\ell-B_\ell \end{bmatrix}$$

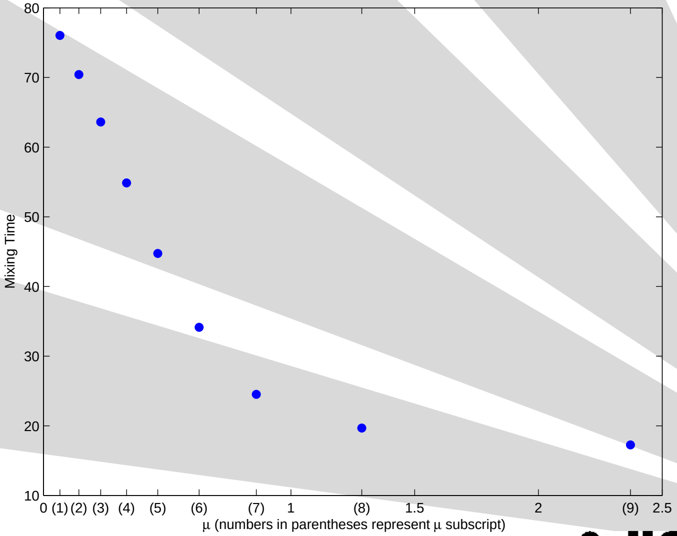
where $A_i = i/(2m)\min\{1, 1/\mu\}$,
$B_i = \frac{if_i}{2mf_{i-1}}\min\{1, \mu\}$ and $\ell = m - n + 1$.

- Values B_i can be estimated from SIS.
- The mixing time is then given by the formula

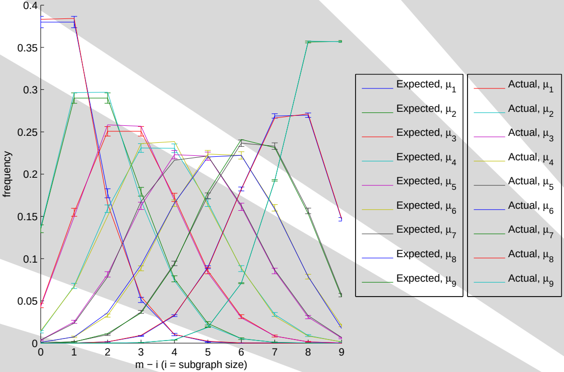
$$(1-\lambda_\mu)^{-1}(\ln m + \ln \epsilon^{-1})$$

where λ_μ is the second eigenvalue.

Calculated Mixing Time



Calculated Mixing Time



Mixing time chosen appropriately: Sample subgraph size distribution follows the expected distribution. 71% within 1 standard deviation.

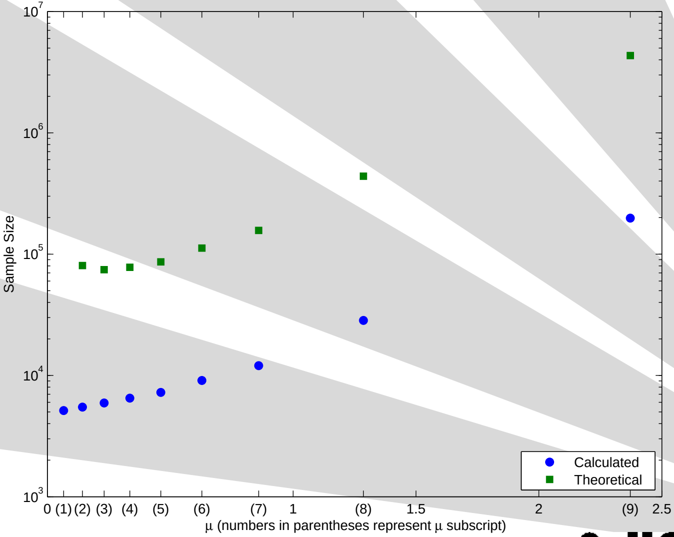
Sample Size Calculation

- Let X be a sample subgraph chosen with distribution π_μ.
- Measure the random variable $Z_i = (\mu_{i-1}/\mu_i)^{m-|X|}$.
- Expected value: $\mathrm{E}[Z_i] = Z(\mu_{i-1})/Z(\mu_i)$.
- Relative variance: $\mathrm{Var}[Z_i]/(\mathrm{E}[Z_i])^2 \leq Z(\mu_i)/Z(\mu_{i-1})$.

Sample Size

- Sample size depends on variance.
- Variance depends on ratio $Z(\mu_i)/Z(\mu_{i-1})$.
- $Z(\mu)$ may be estimated from SIS.

Calculated Sample Size



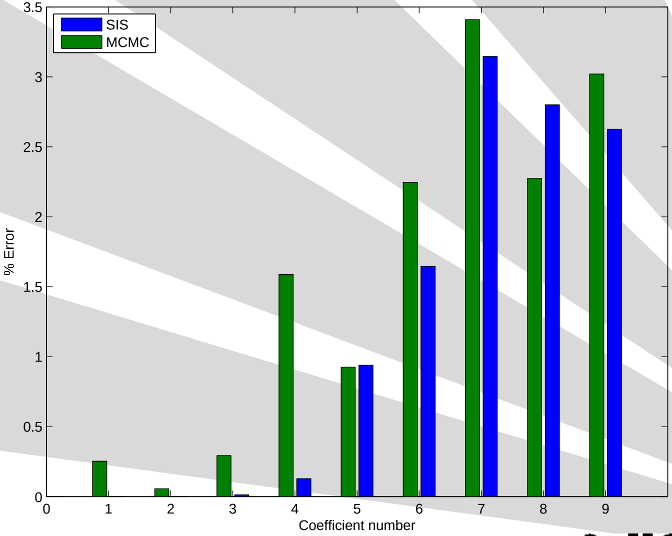
Example Estimation

- $\mu_5 = 0.4618$, $\mu_6 = 0.6291$
- From SIS, we estimate $Z(\mu_5) = 277.7$, and $Z(\mu_6) = 1295$. So the relative variance of Z_6 is approximately bounded by 4.663.
- We run the MCMC, and get a sample variance for Z_6 as 0.3020, well below the bound.
- Compare to the actual values: $Z(\mu_5) = 275.3$, $Z(\mu_6) = 1277.$, to bound the relative variance of Z_6 by 4.640. The population variance for Z_6 is 0.2995.

Calculated Coefficients

Index k	Actual f_k	SIS	% Error	MCMC	% Error
0	1	1	0.00	1	0.0
1	15	15	0.00	14	0.25
2	105	105	0.00	92	0.06
3	454	454	0.01	405	0.29
4	1350	1356	0.13	1317	1.59
5	2900	2933	0.94	2737	0.93
6	4578	4698	1.65	4282	2.25
7	5245	5454	3.15	4943	3.41
8	4092	4307	2.80	3506	2.28
9	1728	1799	2.63	1586	3.02

Calculated Coefficients



Comparison

- **Fugacity:**
  - We always need many different values of the fugacity.
  - The method currently used in practice (guess and check) does not predict the number that will be needed.
  - This method ensures that only the minimum number ($m - n$) of fugacities are needed.
- **Mixing Time:**
  - For this problem, there is no theoretical bound on the mixing time.
  - This method calculates a mixing time on the fly for the actual graph being measured, ensuring that the minimum number of steps are taken.
- **Sample Size:**
  - Estimation using SIS methods leads to significant reduction in sample size from the theoretical bounds.

Extending to the Multi-variate Case

- In the general problem of calculating $R(G;p)$, we let p_e be the probability that an edge e is reliable. These values may be distinct for different edges.

- There is no longer a notion of coefficients, so we must estimate the actual value $R(G;p)$.

- First algorithm uses the same search tree as in the single variable case.

Subgraph Search Tree

For any connected $H \subseteq G$, let $c(H) = \prod_{e \in H} p_e \prod_{e \notin H}(1 - p_e)/(m - |H|)!$ and D_H the set of edges in H that are not bridges.

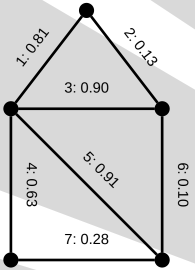
For any $e \in D_H$, let $P(e|H) = (1 - p_e)/\sum_{e \in C_H}(1 - p_e)$.

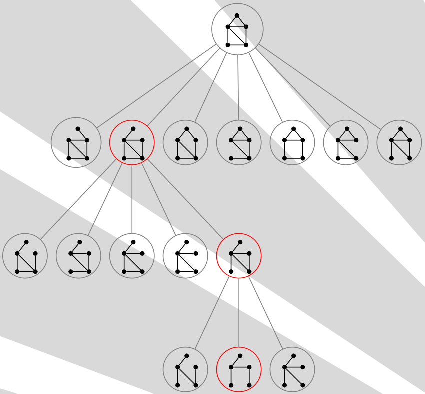
To get the estimate, start with $H_0 = G$, and the estimate $R = c(G)$. For $k = 1$ to $m - n + 1$:

- Set $H_k = H_{k-1} - \{e\}$ with probability $P(e|H_{k-1})$, and set $a_k = P(e|H_{k-1})^{-1}$.
- Set $R = R + c(H_k) \prod_{i=1}^{k} a_i$

Example

$$a_1 = \frac{3.24}{1-0.13}$$
$$a_2 = \frac{2.18}{1-0.28}$$
$$a_3 = \frac{1.09}{1-0.91}$$





$R_{est} = 0.3696$
$R_{actual} = 0.5294$
From 1000 samples, $R = 0.5355$
with variance 0.1162.

Problems with the Subgraph Search Tree

- Unknown variance.
- Sometimes, single runs return values greater than 1.

Tutte-like Search Tree

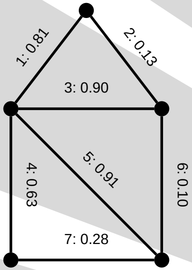
Order the edges as $e_1, \ldots e_m$ with probabilities $p_1, \ldots, p_m$, respectively.

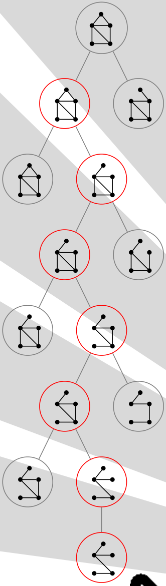
Start with $H = G$ and $R = 1$. For $i = 1$ to m

- If $H - e_k$ is connected, set $H = H - e_k$ with probability $1 - p_k$.
- Otherwise, set $R = p_k \cdot R$.

Note: It is provably optimal that edges be ordered so that $p_i \leq p_{i+1}$.

Example





$R_{est} = 0.28$
$R_{actual} = 0.5294$
From 1000 samples, $R = 0.5282$
with variance 0.0231.

Problems with the Tutte-like Search Tree

- Unknown variance.
- Works poorly on extremely sparse graphs.

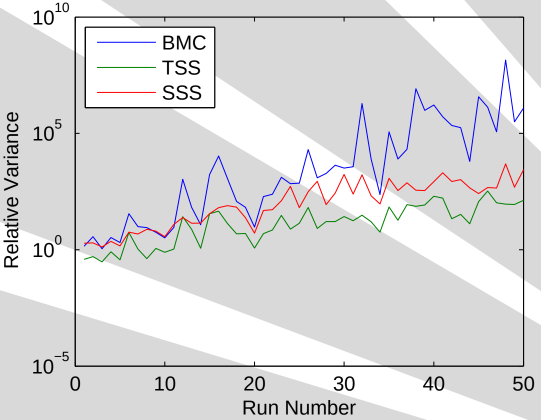
Comparison

Compare to existing methods: Karger, basic Monte Carlo.

Compare on sparse graphs.

Tested dependence on size, density, and variance of edge probabilities.

Size Dependence

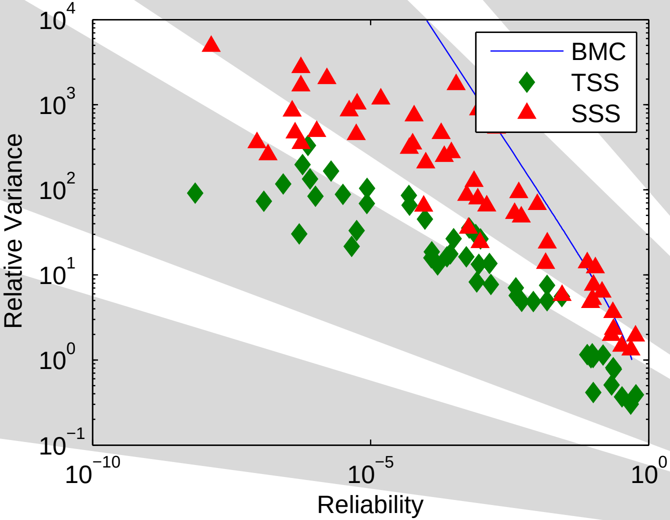


Five graphs per n, n varies from 10 to 100 (increments of 10).

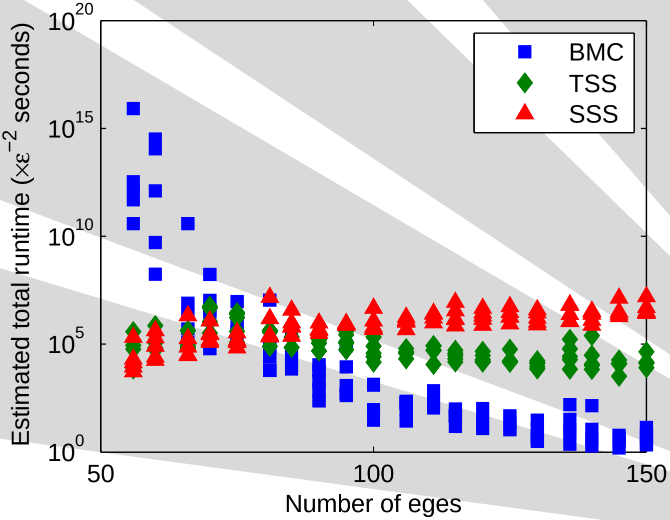
$m = 2n$

Early run numbers have fewer nodes.

Size Dependence



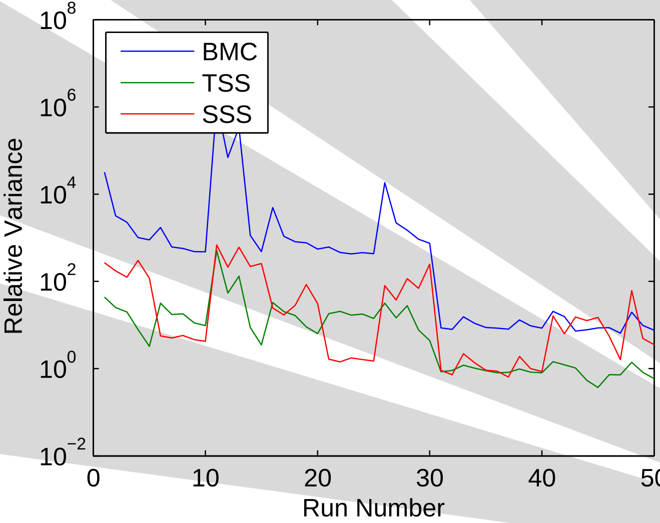
Density Dependence



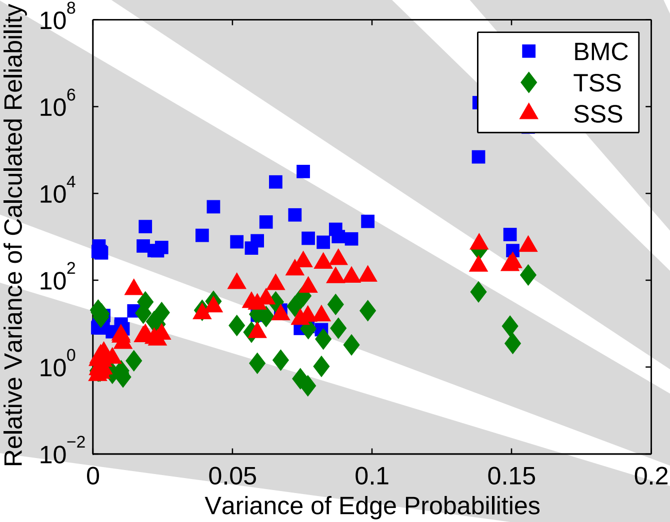
Edge Variance Dependence

Trials 1–5: Uniform on $(0, 1)$
Trials 6–10: Uniform on $(0.25, 0.75)$
Trials 11–15: Uniform on $(0, 0.25) \cup (0.75, 1)$
Trials 16–20: Normal with $\mu = 0.5$, $\sigma = 0.25$
Trials 21–25: Normal with $\mu = 0.5$, $\sigma = 0.05$
Trials 26–30: Normal with $\mu = 0.5$, $\sigma = 0.5$
Trials 31–35: Uniform on $(0.8, 1)$
Trials 36–40: Normal with $\mu = 0.9$, $\sigma = 0.05$
Trials 41–45: $1 - x$, where x is exponential with $\lambda = 0.5$
Trials 46–50: $1 - x$, where x is exponential with $\lambda = 0.1$

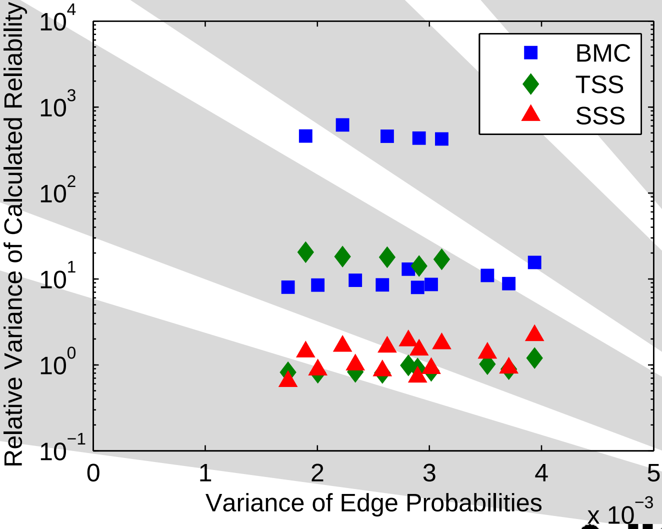
Edge Variance Dependence



Edge Variance Dependence



Edge Variance Dependence



Future Work

- Apply to larger graphs and networks, preferably real ones.
- Theoretical mixing time bound.
- Explore methods of reducing the sample size for large μ.
- Use SIS on other problems where we have an MCMC to increase the efficiency of the MCMC algorithm.
- Theoretical results on when one multi-variate algorithm is better than another.
- Apply to other Tutte polynomial calculations.

References

- I. Beichl, B. Cloteaux, and F. Sullivan. An approximation algorithm for the coefficients of the reliability polynomial. *Congr. Numer.*, 197:143–151, 2009.
- I. Beichl, E. Moseman, and F. Sullivan. Computing network reliability coefficients. *Congr. Numer.*, 207:111–127, 2011.
- D. R. Karger. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM J. Comput.*, 29(2):492–514 (electronic), 1999.
- D. E. Knuth. Estimating the efficiency of backtrack programs. *Math. Comp.*, 29:122–136, 1975. Collection of articles dedicated to Derrick Henry Lehmer on the occasion of his seventieth birthday.