Sequential Algorithms for Generating Random Graphs

Amin Saberi
Stanford University

Outline

- Generating random graphs with given degrees
  (joint work with M Bayati and JH Kim)

- Generating random graphs with large girth
  (joint work with M Bayati and A Montanari)

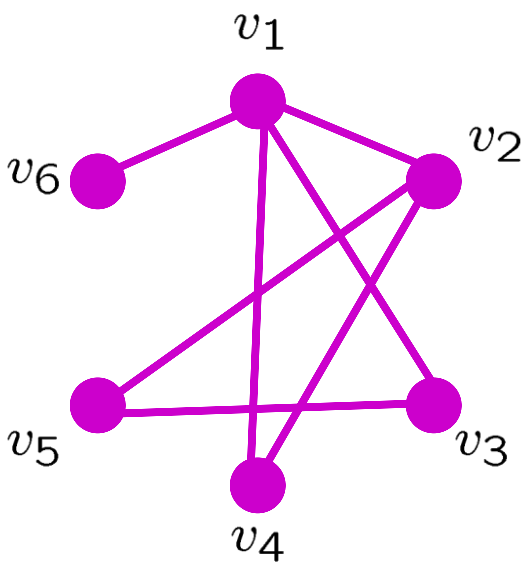
Problem

Given integers $d_1, \ldots, d_n$ $m = \frac{1}{2}\sum_{i=1}^{n} d_i$

Generate a *simple* graph with that degree sequence chosen uniformly at random

Example

$(d_1, \ldots, d_6) = (4, 3, 2, 2, 2, 1)$



Application

Generating large sparse graphs, with sparse (typically power-law) degree sequence

- Simulating Internet topology (e.g. Inet)
- Biological Networks (motif detection)

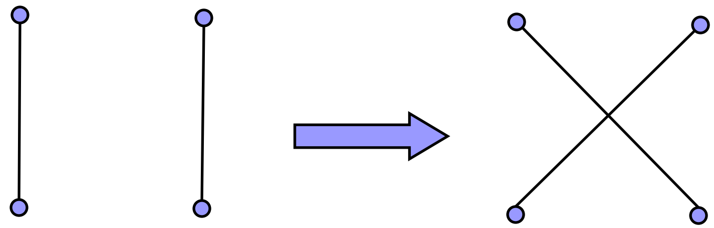
  motif: sub-networks with higher frequency than random

- Coding theory: bipartite graphs with no small stopping sets

Existing methods (Theory)

Markov Chain Monte Carlo method

- The switch chain



It is ``Rapidly mixing'':
[Kannan-Tetali-Vampala 99]
[Cooper-Dyer-GreenHill 05]
[Feder-Guetz-S.-Mihail 07]

← Running time at least $O(n^7)$

- Jerrum-Sinclair chain

(Walk on the self-reducibility tree)

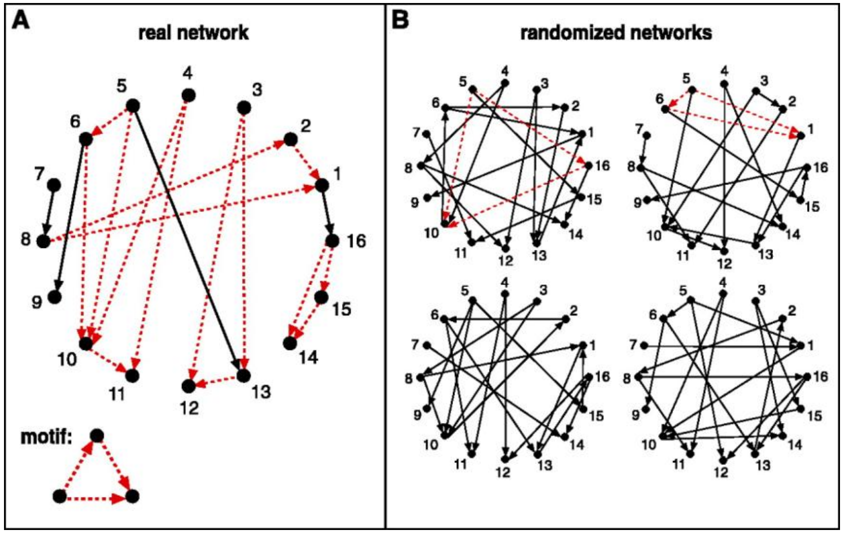
← Running time at least $O(n^4)$

A month on Pentium 2 for n = 1000

Existing methods (Practice)

Lots of heuristics: INET, PEG, …

- Example: Milo et al., Science 2002; Kashtan et al. 2004 on Network Motifs



- The heuristic used for generating random graphs has a substantial bias

New method: Sequential Importance Sampling

Very successful in practice:
Knuth'76: for counting self-avoiding random walks
estimating the running time of heuristics

More recently for random graph generation:
Chen-Diaconis-Holmes-Liu'05, Blitzstein-Diaconis'05

No analysis! (with the exception of this work and Blanchet 06)

Our Algorithm

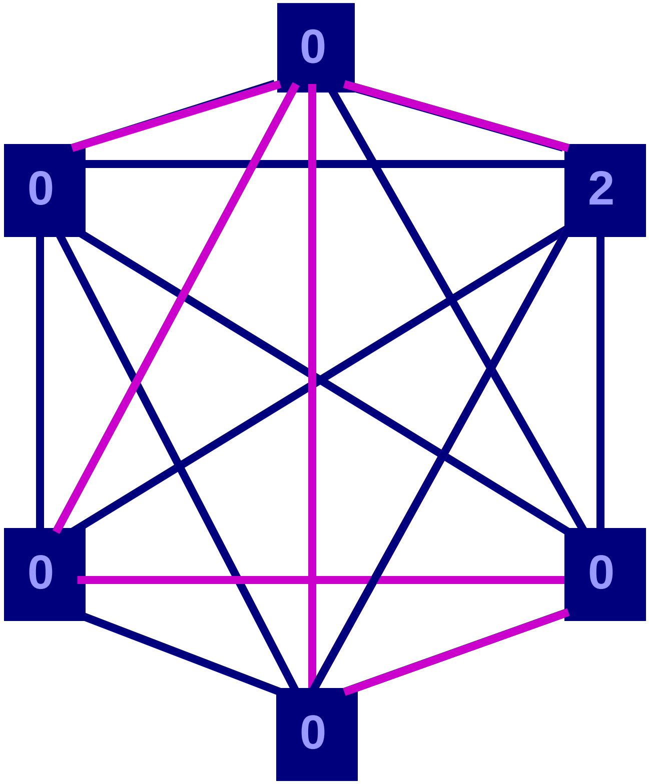
remaining degree → $\hat{d}_i$; degree → $d_i d_j$

Repeat

Add an edge between (i,j) with probab. $p_{ij} \propto \hat{d}_i \hat{d}_j \left(1 - \frac{d_i d_j}{4m}\right)$.

Until $m = \sum_{i=1}^{n} d_i/2$ edges are added

or there are no valid choices left



failure

Analysis of the Algorithm

Theorem 1 (Bayati-Kim-S. 07):

The running time of the algorithm is $O(m\, d_{\max})$.

Furthermore, if $d_{\max} = O(m^{0.25-\tau})$

Or if the degree sequence is regular and $d = O(n^{0.5-\tau})$

Algorithm is successful with probability $1 - o(1)$

The probability of generating any graph is $\frac{1}{L}(1 \pm o(\frac{1}{\log m}))$

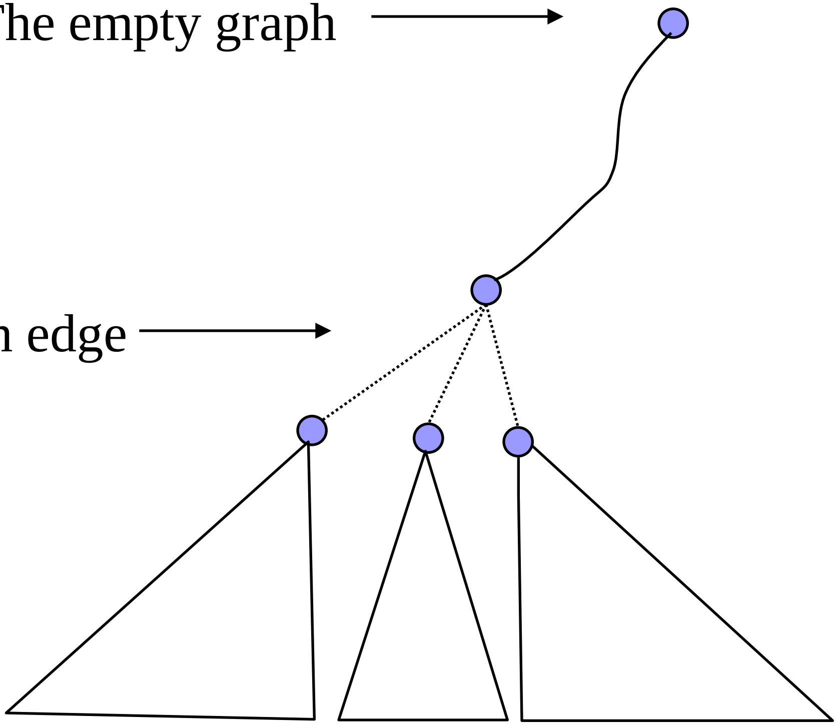
where L is the number of graphs

Basic Idea of $p_{ij} \propto \hat{d}_i \hat{d}_j (1 - \frac{d_i d_j}{4m})$.

The empty graph →

The tree of execution

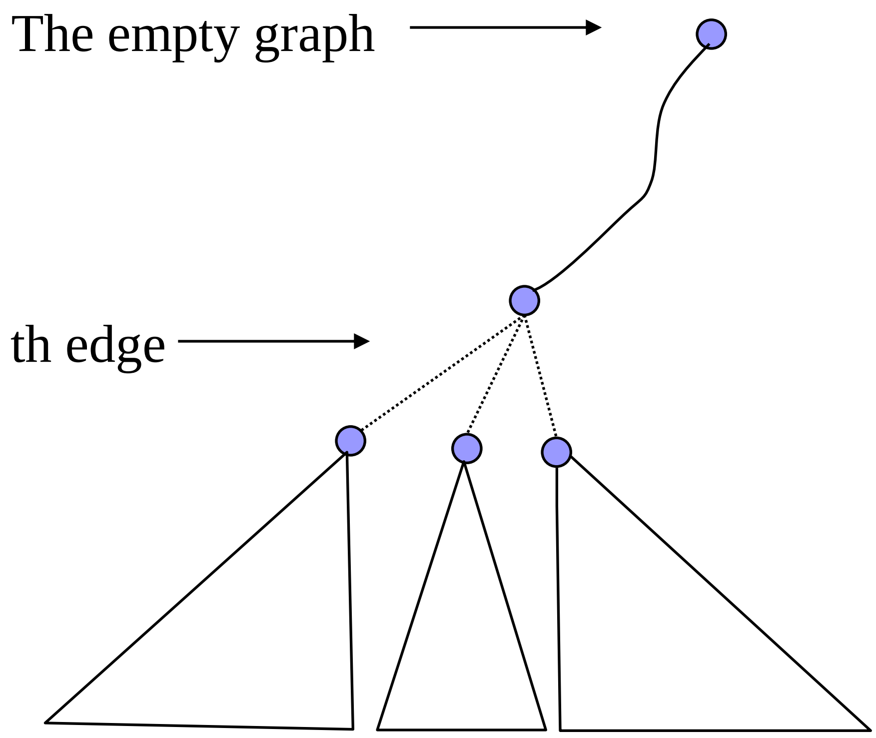
Choosing the k th edge →



the prob. of choosing a sub-tree
should be proportional to
the number of valid leaves

Basic Idea of $p_{ij} \propto \hat{d}_i \hat{d}_j (1 - \frac{d_i d_j}{4m}).$

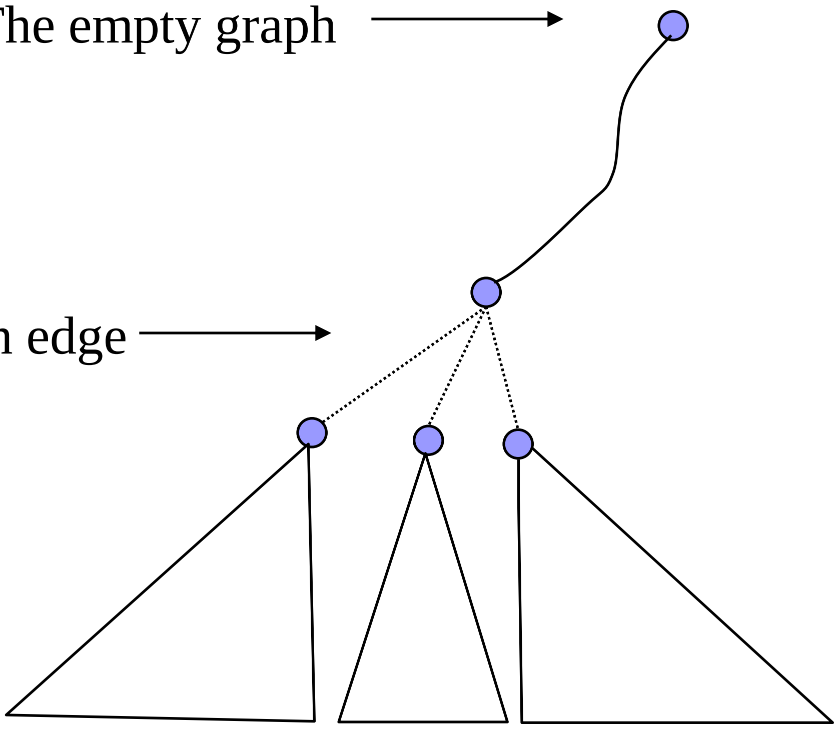
The tree of execution



the prob. of choosing a sub-tree
should be proportional to
the number of valid leaves

Basic Idea of $p_{ij} \propto \hat{d}_i \hat{d}_j (1 - \frac{d_i d_j}{4m})$.

The tree of execution



the prob. of choosing a sub-tree
should be proportional to
the number of valid leaves

Technical ingredient: concentration results on the
distribution of leaves in each sub-tree
(improving Kim-Vu 06, McKay-Wormald 91)

Analysis of the Algorithm

Theorem 1 (Bayati-Kim-S. 07):

The running time of the algorithm is $O(m\, d_{\max})$.

Furthermore, if $d_{\max} = O(m^{0.25-\tau})$

Or if the degree sequence is regular and $d = O(n^{0.5-\tau})$

Algorithm is successful with probability $1 - o(1)$

The probability of generating any graph is $\frac{1}{L}(1 \pm o(\frac{1}{\log m}))$
where L is the number of graphs

Sequential Importance Sampling

Consider a run of the algorithm

Let P_r be the probability of the edge chosen in step r

Define $$X = \begin{cases} \frac{1}{\prod_{r=1}^{m} p_r} & \text{if Alg. is successful} \\ 0 & \text{if Alg. fails} \end{cases}$$

Crucial observation: $\mathbb{E}(X) = L$

X is an unbiased estimator for the number of graphs

Using SIS to get an FPRAS

By taking several samples of X, we can have a good estimate of L

Then using the right rejection sampling:

Theorem 2 (Bayati-Kim-S. '07):

Can generate any graph with probability $1 \pm \epsilon$ of uniform.

In time $O\left(\epsilon^{-2}\, m\, d_{\max}\right)$.

An FPRAS for counting and random generation

Outline

- Generating random graphs with given degrees
  (joint work with M Bayati and JH Kim)

- Generating random graphs with large girth
  (joint work with M Bayati and A Montanari)

Graphs with large girth

$(0,1,0,1)$ $(0,1,0,1,\underbrace{0,1,0})$

Check bits

x_1

x_2

x_3

x_4

x_5

x_6

x_7

(mod 2)

$$x_1 + x_4 + x_5 + x_6 = 0$$

$$x_2 + x_4 + x_5 = 0$$

$$x_3 + x_4 + x_6 + x_7 = 0$$

Challenge:

Construct graphs with given degrees with no short cycles

Amraoui-Montanari-Urbanke'06.

Example: Triangle free graphs.

Consider all graphs with *n* vertices and *m* edges.

Let *G* be one such graph chosen uniformly at random.

Can think of *G* as Erdös-Renyi graph $G(n,p)$ where $p = \frac{m}{\binom{n}{2}} \approx \frac{2m}{n^2}$.

$n_3(G)$ = number of triangle sub-graphs of *G*.

$$\mathbb{P}\left(n_3(G)=0\right) \approx e^{-n_3(G)} = e^{-\binom{n}{3}p^3} \to \begin{cases} 1 & \text{if } np \to 0 \\ 0 & \text{if } np \to +\infty \end{cases}$$

Same phase transition holds when we need graphs of girth k.

Our Algorithm

Initialize G by an empty graph with vertices $V = (v_1, \ldots, v_n)$.

Repeat

Choose a pair (v_i, v_j) with probability P_{ij} from the set of suitable pairs and set $G = G \cup \left\{(v_i, v_j)\right\}$.

Until m edges are added or

there are no suitable pair available (failure)

Theorem (Bayati, Montanari, S. 07)

- For a suitable P_{ij} and $m = O\left(n^{1+\frac{1}{2k(k+3)}}\right)$ the output distribution of our algorithm is asymptotically uniform. i.e.

$$\lim_{n\to\infty} d_{\mathsf{TV}}(\mathbb{P}_A, \mathbb{P}_U) = 0.$$

Output dist. of algorithm

Uniform probability

Furthermore, the algorithm is successful almost surely.

Remark: can be extended to degree sequences applicable to LDPC codes…

What is P_{ij}

- Consider the partially constructed graph *G* with *t* edges.
- Let *S* be the n£n matrix of all suitable pairs.
- Let A, A^c be adjacency matrix of *G* and its complement.

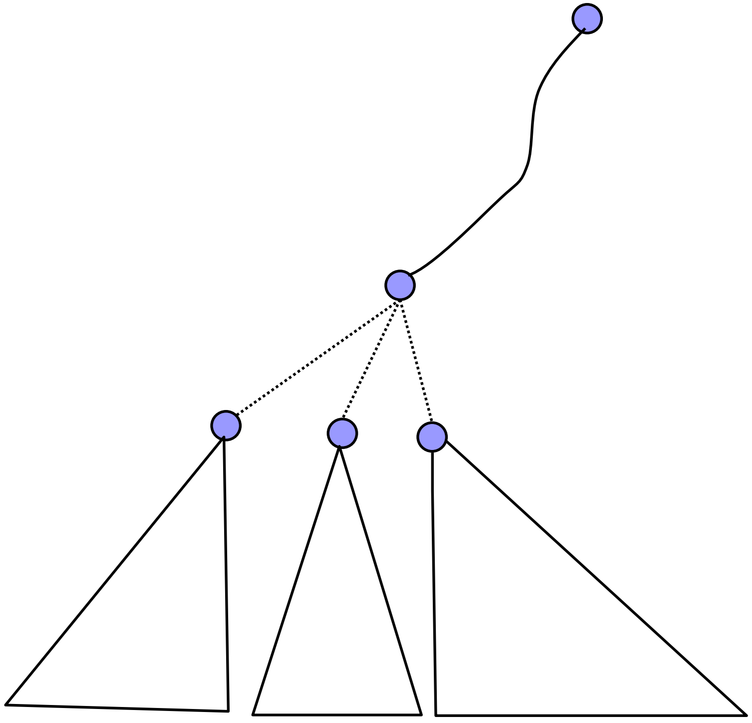
$$P \propto S \odot \sum_{\ell=1}^{k} \left(A + \frac{m-t}{\binom{n}{2} - t} A^c \right)^{\ell}$$

Coordinate-wise multiplication

- P can be calculated quickly (e.g. with MATLAB)

Where does h come from?

The execution tree:



Problem: estimate the number of valid leaves of each subtree

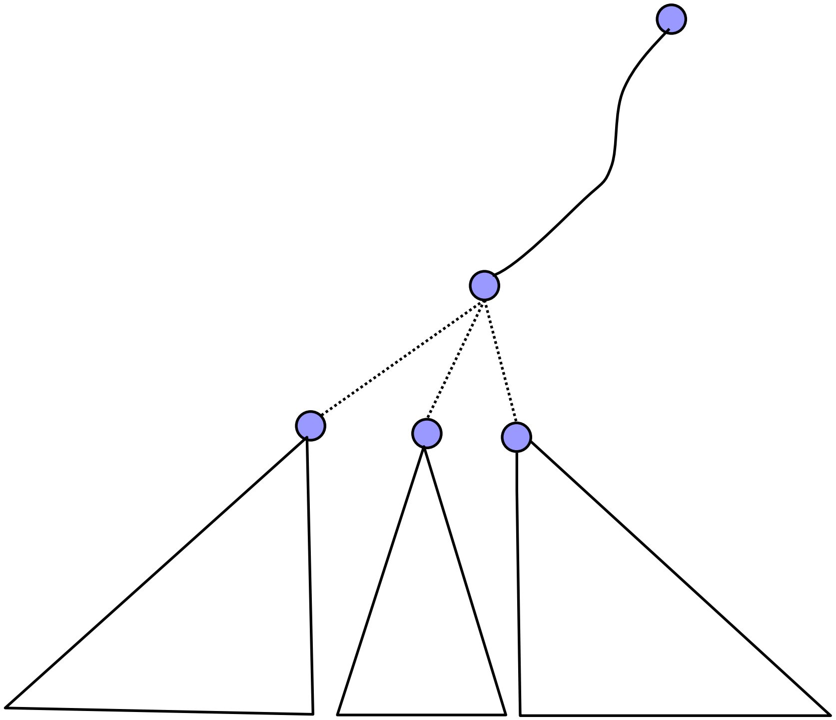
In other words

estimate the number of extensions of the partial graph that do not have short cycles

Where does h come from?

add the remaining m - k edges uniformly at random, and compute the expected number of small cycles Y.

Assuming the distribution of small cycles is Poisson, the probability of having no small cycles is e^{-Y}



Summary

- Random simple graphs with given degrees
- Random bipartite graphs with given degrees and large girth
- More extensive analysis of SIS?