A Sequential Algorithm for Generating Random Graphs

Mohsen Bayati* Jeong Han Kim† Amin Saberi ‡

Abstract

The goal of this article is to develop a fast algorithm for generating simple random graphs with a given degree sequence. For degree sequence (d_i) with maximum degree $d_{\max} = O(m^{1/4-\epsilon})$, our algorithm generates an asymptotically uniform random graph with that degree sequence in almost linear time, where $m = \frac{1}{2}\sum_i d_i$ is the number of edges in the graph and ϵ is any positive constant. Our method also gives an independent proof of McKay's estimate [35] for number of such graphs. Additionally, for d-regular graphs on n vertices we show the above result holds in the larger region $d_{\max} = O(m^{1/3-\epsilon})$, or equivalently $d = O(n^{1/2-\epsilon})$.

*Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510; bayati@stanford.edu

†Department of Mathematics, Yonsei University, South Korea; jehkim@yonsei.ac.kr

‡Departments of Management Science and Engineering, Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305; saberi@stanford.edu.

1 Introduction

Random graph models and algorithms are one of the central topics in graph theory and theoretical computer science. Recently they have also attracted a lot of attention for modeling real world networks such as the Internet, World Wide Web and biological and social networks [14, 21]. In particular, random graph generation is an important tool for detecting certain motifs in biological networks [39] and for simulating a wide range of networking protocols on the Internet topology [43, 22, 34, 16, 4]. In these applications we are usually interested in networks with simple underlying graphs; i.e. when no self edge loops or multiple edges are allowed.

Unfortunately, there is a big gap between theory and practice for this problem. When the degree sequence is non-regular, there is no efficient algorithms known for random graph generation. The existing polynomial algorithms [24, 36] have a large running time, which makes them very difficult to use for real-world networks which have tens of thousands of nodes (for example, see [23]). On the other hand, there is no rigorous analysis of the simple heuristics used in practice. In fact, it is known that some of these heuristics are incorrect [38, 39].

In this paper, we give a practical algorithm for sampling uniformly from the set of all simple graphs with given degrees $d_1, \ldots, d_n$. Suppose $\sum_{i=1}^{n} d_i = 2m$. Our algorithm is as follows: Start with an empty graph and sequentially add edges between pairs of non-adjacent vertices. In every step of the algorithm, the probability that an edge is added between two distinct vertices i and j is proportional to $\hat{d}_i\hat{d}_j(1-d_id_j/4m)$ where $\hat{d}_i$ and $\hat{d}_j$ denote the remaining degrees of vertices i and j.

We show that for any $\epsilon > 0$ when the maximum degree $d_{\max}$ is of $O(m^{1/4-\epsilon})$, our algorithm finds an asymptotically uniform sample with a running time of $O(md_{\max})$. Moreover for $d = O(n^{1/2-\epsilon})$, our algorithm generates an asymptotically uniform d-regular graph. Our results are improving the bounds of Kim and Vu [31] and Steger and Wormald [42] for regular graphs and extending them to more general degree sequences.

In order to explain the intuition behind our algorithm and proof, let us define the configurational (pairing) model for graphs with general degree sequences [13, 2, 40]. In the configurational model, we represent every vertex i of degree d_i with d_i mini-vertices. A perfect matching between the mini-vertices would correspond to a multi-graph with the degree sequence $d_1, d_2, \cdots, d_n$. It is easy to see that even when the degrees are growing as $\log n$, the probability that a uniformly chosen perfect matching corresponds to a simple graph is very small.

A natural way ([42]) for increasing the probability of success is to generate the matching sequentially by adding the new pairs uniformly at random avoiding the invalid choices that create loops or multiple edges in the graph. We will show that for non-regular degree sequences, the above algorithm has an exponential bias. More precisely, the probability of generating a graph G by this algorithm is proportional to $\exp\left(\sum_{i\sim_G j} d_id_j/4m\right)$. This is mainly because pairing mini-vertices belonging to high degree vertices reduces the number of valid choices in the future steps and therefore creates a bias towards graphs with more of such edges. Our algorithm balances this effect by distributing the bias term into the probabilities of the pairs.

The main part of our proof is deriving very close estimates of the probability of each valid choice in every step. This is necessary because the final error in the output distribution of our algorithm will be the product of these errors. Using Kim-Vu's polynomial concentration inequality ([30, 45] see also [3]) is critical for our analysis. Furthermore, for d-regular graphs when $d = \Omega(n^{1/3})$ we use a simple martingale exposure inequality where polynomial tail inequality seems to be not applicable. This is because approximating the random variable of interest with a polynomial of independent random variables causes the variance to grow exponentially.

We also prove for general degree sequences with $d_{\max} = O(m^{1/4-\epsilon})$ the probability that one trial of our algorithm produces a simple graph is very close to 1. We note that our proof does not rely on existing McKay and Wormald's estimates on the number of simple graphs with a given degree sequence [35, 37].

In fact our analysis yields a different (and may be simpler) proof for McKay's formula [35].

Our algorithm and its analysis provide more insight into the modern random graph models such as the configurational model or random graphs with a given expected degree sequence [19]. These models are considered to approximate random graphs with a given degree sequence and more amenable to estimate graph invariants such as diameter or the sizes of connected components. It is easy to see that, in these models, the probability of having an edge between vertices i and j of the graph is proportional to $d_i d_j$. However, one can use our analysis or McKay's formula [35] to see that in a random simple graph this probability is proportional to $d_i d_j(1 - d_i d_j/2m)$. Simple examples show that this difference can carry over to other quantities such as the expected number of neighbors of a vertex that have a fixed degree. On the other hand, we expect that, by adding the correction term and using the concentration result of this paper, it is possible to achieve a better approximation for which sandwiching theorems similar to [32] can be applied.

We have also used similar ideas to generate random graphs with large girth [8]. These graphs are used to design high performance Low-Density Parity-Check (LDPC) codes. One of the methods for constructing these codes is to generate a random bipartite graph with an optimized degree sequence [5]. The performance of the corresponding code in the low noise regime is directly related to the size of the girth of the graph. Two common heuristics for finding good codes are as follows: (i) generate many random copies and pick the one with the highest girth; (ii) grow progressively the graph while avoiding short loops. While the former implies an exponential slowing down in the target girth, the latter induces a systematic and uncontrolled bias in the graph distribution. Using the same principles, we can design a more efficient algorithm that sequentially adds the edges avoiding small cycles. Similar to our analysis here, we can calculate bias of the suggested algorithm and distribute it accordingly to achieve uniform generation as well as small probability of failure.

The rest of the paper has the following structure. The algorithm and main results are stated in Section 2. Some definitions and the main idea is the subject of Section 3. The analysis is presented in Section 4.

1.1 Related work

Generating random graphs with a given degree sequence has been studied extensively. For a detailed survey of related results see [44, 12]. Probably the most studied and the most successful approach for this problem has been the Markov Chain Monte Carlo (MCMC) method. [25, 18, 20, 28, 10]. In [25], Jerrum and Sinclair gave a fully polynomial randomized approximation scheme for generating graphs with "stable" degree sequence [25, 26]. In a different paper [24] and using McKay's estimate [35], they introduced a different Markov chain with a better running time of $O(m^2 n^2 \log m)$ for degree sequences with maximum degree of $o(m^{1/4})$. For generating random bipartite graphs with a general degree distribution, Bezáková et al. [10] improved the best known running time and achieved the running time of $O(n^4 m^3 d_{\max} \log^5(2n))$.

The configurational model is also studied for regular graphs [13, 9] and more recently for general degree sequences [2, 41, 1]. McKay and Wormald [36] considered a modified version of this model in which the loops and multiple edges can be removed by a sequence of switches. But in order to obtain uniform samples at the end they used an accept/reject scheme within the switching phase. If $M = \sum_{i=1}^n d_i$ and $M_2 = \sum_{i=1}^n d_i(d_i - 1)$ then they showed for $d_{\max}^3 = O(M^2/M_2)$ and $d_{\max}^3 = o(M + M_2)$ the average running time of the algorithm is $O(M + M_2^2)$. This leads to $O(n^2 d^4)$ average running time for d-regular graphs with $d = o(n^{1/3})$.

Sequential algorithms and in particular sequential importance sampling (SIS) methods have been widely used in practice for this and other similar problems [33, 6]. Chen et al [17] used the SIS method to generate bipartite graphs with a given degree sequence. Later Blitzstein and Diaconis [12] used a similar approach to generate general graphs. Bezáková et al [11] studied the algorithm of Chen et al and showed that their procedure may generate contingency tables with probability far from uniform distribution for certain row and column sums. However the simplicity of these algorithms and their great performance in some instances, suggest further study of the SIS method is necessary.

In fact our algorithm is inspired by the algorithm of Blitzstein and Diaconis [12] that uses Sequential Importance Sampling. We propose a small modification to their algorithm that improves its asymptotic behavior substantially.

2 Our Algorithm

Assume we are given a sequence of n nonnegative integers $d_1, d_2, \ldots d_n$ with $\sum_{i=1}^n d_i = 2m$ and a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$. Assume the sequence of given degrees $d_1, \ldots, d_n$ is *graphical.* That is there exists at least one simple graph with those degrees. We suggest the following algorithm for sampling an element of the set $\mathcal{L}(\bar{d})$ of all labeled simple graphs G with $V(G) = V$ and degree sequence $\bar{d}$ $=(d_1, d_2, \cdots, d_n)$.

Algorithm A

---

(1) Start with an empty set E of edges. Let also $\hat{d} = (\hat{d}_1, \ldots, \hat{d}_n)$ be an n-tuple of integers and initialize it by $\hat{d} = \bar{d}$.

(2) Choose two vertices $v_i, v_j \in V$ with probability proportion to $\hat{d}_i \hat{d}_j (1 - \frac{d_i d_j}{4m})$ among all pairs i, j with $i \neq j$ and $(v_i, v_j) \notin E$. Add (v_i, v_j) to E and reduce each of $\hat{d}_i$, $\hat{d}_j$ by 1.

(3) Repeat step (2) until no more edge can be added to E.

(4) If $|E| < 2m$ report *failure* and restart from step (1), otherwise output $G = (V, E)$.

Note that for regular graphs the factors $1 - d_i d_j / 4m$ are redundant and Algorithm A is the same as Steger-Wormald's [42] algorithm. Throughout this paper m refers to total number of edges in the graph, $d_{\max} = \max_{i=1}^n \{d_i\}$ and d is the degree in regular case; i.e. when $d_i = d$ for all $i = 1, \ldots, n$. The following theorem summarizes the main result of this paper.

Theorem 1 *Let* $\epsilon > 0$*:*

(a) For any graphical degree sequence $\bar{d}$ *with maximum degree of* $O(m^{1/4-\epsilon})$*, Algorithm* A *terminates successfully in one round with probability* $1 - o(1)$ *and generates an asymptotically uniform random graph with that degree sequence. Expected running time of Algorithm* A *is at most* $O(md_{\max})$*.*

(b) For d-regular graphs the result of part (a) holds in the larger region $d = O(n^{1/2-\epsilon})$*.*

3 Definitions and the Main Idea

Before explaining our approach let us quickly review the configurational model. Let $W = \cup_{i=1}^n W_i$ be a set of $2m = \sum_{i=1}^n d_i$ labeled mini-vertices with $|W_i| = d_i$. If one picks two distinct mini-vertices uniformly at random and pairs them together then repeats that for remaining mini-vertices, after m steps a perfect matching $\mathcal{M}$ on the vertices of W is generated. Such matching is also called a *configuration* on W. One can see that number of all distinct configurations is equal to $(1/m!) \prod_{r=0}^{m-1} \binom{2m-2r}{2}$. The same equation shows that this sequential procedure generates all configurations with equal probability. Given a configuration $\mathcal{M}$, combining all mini-vertices of each W_i to form a vertex v_i, a graph $G_{\mathcal{M}}$ is generated whose degree sequence is $\bar{d}$.

Note that the graph $G_{\mathcal{M}}$ might have self edge loops or multiple edges. In fact McKay and Wormald's estimate [37] show that this happens with very high probability except when $d_{\max} = O(\log^{1/2} m)$. In order to fix this problem [42] at any step one can only look at those pairs of mini-vertices that lead to simple graphs (denote these by *suitable pairs*) and pick one uniformly at random. For regular graphs with degrees growing as small power of m Steger-Wormald [42] and Kim-Vu [31] have shown this approach asymptotically samples regular graphs with uniform distribution.

Unfortunately for general degree sequences some graphs can have probabilities that are far from uniform. In this paper we will show that for general degree sequences suitable pairs should be picked non-uniformly. In fact *Algorithm* A *is a weighted configurational model where at any step a suitable pair* $(u, v) \in W_i \times W_j$ *is picked with probability proportional to* $1 - d_i d_j / 4m$.

Here is a rough intuition behind Algorithm A. Define the *execution tree* T of the configurational model as follows. Consider a rooted tree where the root (the vertex in level zero) corresponds to the empty matching in the beginning of the model and level r vertices correspond to all partial matchings that can be constructed after r steps. There is an edge in T between a partial matching $\mathcal{M}_r$ from level r to a partial matching $\mathcal{M}_{r+1}$ from level $r+1$ if $\mathcal{M}_r \subset \mathcal{M}_{r+1}$. Any path from the root to a leaf of T corresponds to one possible way of generating a random configuration.

Let us denote those partial matchings $\mathcal{M}_r$ whose corresponding partial graph $G_{\mathcal{M}_r}$ is simple by "valid" matchings. Denote the number of invalid children of $\mathcal{M}_r$ by $\Delta(\mathcal{M}_r)$. Our goal is to sample valid leaves of the tree T uniformly at random. A simple improvement to configurational model is to restrict the algorithm at step r to the valid children of $\mathcal{M}_r$ and picking one uniformly at random.

For regular graphs this approach leads to almost uniform generation since the number of valid children for all partial matchings at level r of the T, is almost equal. However it is crucial to note that for non-regular degree sequences if the $(r+1)^{th}$ edge matches two elements belonging to vertices with larger degrees, the number of valid children for $\mathcal{M}_{r+1}$ will be smaller. Thus there will be a bias towards graphs that have more of such edges.

In order to find a rough estimate of the bias fix a graph G with degree sequence $\bar{d}$. Let $p_r = r/m$ and $q_r = 1 - p_r$. Let $\mathrm{M}(G)$ be the set of all leaves $\mathcal{M}$ of the tree T that lead to graph G; i.e. $G_{\mathcal{M}} = G$. It is easy to see that $|\mathrm{M}(G)| = m! \prod_{i=1}^n d_i!$. Moreover for exactly $p_r|\mathrm{M}(G)|$ of these leaves a fixed edge (i, j) of G appears in the first r edges of the path leading to them; i.e. $(i, j) \in \mathcal{M}_r$. Furthermore we can show that for a typical such leaf after step r, number of unmatched mini-vertices in each W_i is roughly $d_i q_r$. Therefore expected number of un-suitable pairs $(u, v) \in W_i \times W_j$ is about $p_r q_r^2 \sum_{i \sim_G j} d_i d_j$. Similarly expected number of un-suitable pairs corresponding to self edge loops is approximately $\sum_{i=1}^n \binom{d_i q_r}{2} \approx 2mq_r^2 \lambda(\bar{d})$ where $\lambda(\bar{d}) = \sum_{i=1}^n \binom{d_i}{2} / (\sum_{i=1}^n d_i)$. Therefore defining $\gamma_G = \sum_{i \sim_G j} d_i d_j / 4m$ and using $\binom{2m-2r}{2} \approx 2m^2 q_r^2$ we can write

$$\mathbb{P}(G) \approx m! \prod_{i=1}^n d_i! \prod_{r=0}^{m-1} \frac{1}{2m^2 q_r^2 - 2mq_r^2 \lambda(\bar{d}) - 4mp_r q_r^2 \gamma_G} \approx e^{\lambda(\bar{d}) + \gamma_G} \; m! \prod_{i=1}^n d_i! \prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2}} \propto e^{\gamma_G}$$

Hence adding the edge (i, j) roughly creates an $\exp(d_i d_j / 4m)$ bias. To cancel that effect we need to reduce probability of picking (i, j) by $\exp(-d_i d_j / 4m) \approx 1 - d_i d_j / 4m$. We will rigorously prove the above argument in the next section.

4 Analysis

Let us fix a simple graph G with degree sequence $\bar{d}$. Denote the set of all edge-wise different perfect matchings on the set of mini-vertices W that lead to graph G by $\mathrm{R}(G)$. Any two elements of $\mathrm{R}(G)$ can be obtained from one another by permuting labels of the mini-vertices in any W_i. We will find probability of generating a fixed element $\mathcal{M}$ of the set $\mathrm{R}(G)$. There are $m!$ different orders for generating edges of $\mathcal{M}$ sequentially and different orderings could have different probabilities. Denote the set of these orderings by $\mathrm{S}(\mathcal{M})$. Thus

$$\mathbb{P}_{\mathrm{A}}(G) = \sum_{\mathcal{M} \in \mathrm{R}(G)} \sum_{\mathcal{N} \in \mathrm{S}(\mathcal{M})} \mathbb{P}_{\mathrm{A}}(\mathcal{N}) = \prod_{i=1}^n d_i! \sum_{\mathcal{N} \in \mathrm{S}(\mathcal{M})} \mathbb{P}_{\mathrm{A}}(\mathcal{N}).$$

For any ordering $\mathcal{N} = \{e_1, \ldots, e_m\} \in \mathrm{S}(\mathcal{M})$ and $0 \le r \le m-1$ let $\mathbb{P}(e_{r+1})$ denote the probability of picking the $(r+1)^{th}$ edge. Hence $\mathbb{P}_{\mathrm{A}}(\mathcal{N}) = \prod_{r=0}^{m-1} \mathbb{P}(e_{r+1})$ and each term $\mathbb{P}(e_{r+1})$ is equal to

$$\mathbb{P}\left(e_{r+1} = (i,j)\right) = \frac{(1 - d_i d_j/4m)}{\sum_{(u,v)\in E_r} d_u^{(r)} d_v^{(r)} (1 - d_u d_v/4m)}$$

where $d_i^{(r)}$ denotes the residual degree of vertex i after r steps that is equal to the number of unmatched mini-vertices of W_i at step r. The set E_r consists of all possible edges after picking $e_1, \ldots, e_r$. Denominator of the above fraction for $\mathbb{P}(e_{r+1})$ can be written as $\binom{2m-2r}{2} - \Psi(\mathcal{N}_r)$ where $\Psi(\mathcal{N}_r) = \Delta(\mathcal{N}_r) + \sum_{(i,j)\in E_r} d_i^{(r)} d_j^{(r)} d_i d_j/4m$. This is because $\sum_{(u,v)\in E_r} d_u^{(r)} d_v^{(r)}$ counts number of suitable pairs in step r and is equal to $\binom{2m-2r}{2} - \Delta(\mathcal{N}_r)$. The quantity $\Psi(\mathcal{N}_r)$ can be also viewed as sum of the weights of the un-suitable pairs. Now using $1 - x = e^{-x+O(x^2)}$ for $0 \le x \le 1$ we can write $\mathbb{P}_{\mathrm{A}}(G)$ in the range $d_{\max} = O(m^{1/4-\epsilon})$ as following

$$\begin{aligned}
\mathbb{P}_{\mathrm{A}}(G) &= \prod_{i=1}^{n} d_i! \prod_{i\sim_G j} (1 - \frac{d_i d_j}{4m}) \sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2} - \Psi(\mathcal{N}_r)} \\
&= \prod_{i=1}^{n} d_i! \; e^{-\gamma_G + o(1)} \sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2} - \Psi(\mathcal{N}_r)}
\end{aligned}$$

The next step is to show that with very high probability $\Psi(\mathcal{N}_r)$ is close to a number $\psi_r(G)$ independent of the ordering $\mathcal{N}$. More specifically for $\psi_r(G) = (2m-2r)^2 \left(\frac{\lambda(\bar{d})}{2m} + \frac{r \sum_{i\sim_G j} (d_i - 1)(d_j - 1)}{4m^3} + \frac{(\sum_{i=1}^n d_i^2)^2}{32m^3} \right)$ the following is true

$$\sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2} - \Psi(\mathcal{N}_r)} = (1 + o(1))\, m! \prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2} - \psi_r(G)}.$$

The proof of this concentration uses Kim-Vu's polynomial [30] and is quite technical. It generalizes Kim-Vu's [31] calculations to general degree sequences. We have shown this cumbersome analysis in [7] and to the interest of space will not cover them here. But in section 4.1 we will prove the concentration for regular graphs in a larger family of graphs using a new technique. As stated in Lemma 2 below we extend previous bound $d = O(n^{1/3-\epsilon})$ proved in [31] to $d = O(n^{1/2-\epsilon})$ via a martingale approach.

Next step is to use simple algebra to show that for $d_{\max} = O(m^{1/4-\epsilon})$

$$\prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2} - \psi_r(G)} = \prod_{r=0}^{m-1} \frac{1}{\binom{2m-2r}{2}} e^{\lambda(\bar{d}) + \lambda^2(\bar{d}) + \gamma_G + o(1)}. \tag{1}$$

Proof of the above equation is shown in the appendix.

The following lemma summarizes the above analysis.

Lemma 1 *For $d_{\max} = O(m^{1/4-\epsilon})$ Algorithm* A *generates all graphs with degree sequence $\bar{d}$ with asymptotically equal probability. More specifically*

$$\sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \mathbb{P}(\mathcal{N}) = \frac{m!}{\prod_{r=0}^{m} \binom{2m-2r}{2}} e^{\lambda(\bar{d}) + \lambda^2(\bar{d}) + o(1)}$$

The following lemma extends the result of Lemma 2 for d-regular graphs to the larger family $d = O(n^{1/2-\epsilon})$.

Lemma 2 *For $d = O(n^{1/2-\epsilon})$ Algorithm* A *generates all d-regular graphs on n vertices with asymptotically equal probability. More specifically*

$$\sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \mathbb{P}(\mathcal{N}) = \frac{m!}{\prod_{r=0}^{m} \binom{2m-2r}{2}} e^{\frac{d^2-1}{4} + o(1)}$$

Our proof for Lemma 2 is based on a variation of the Azuma's concentration inequality for martingales developed in [29] and is presented in section 4.1.

Finally in order to show that $\mathbb{P}_{\mathrm{A}}(G)$ is uniform we need an important piece. The analysis so far has shown that $\mathbb{P}_{\mathrm{A}}(G)$ is independent of G but this probability might be still far from uniform. In other words the algorithm A might fail with very high probability. In section 4.2 we will prove the following lemma which shows this does not happen.

Lemma 3 *For $d_{\max} = O(m^{1/4-\epsilon})$ the probability of failure in one trial of Algorithm* A *is asymptotically zero; i.e.* $\mathbb{P}_{\mathrm{A}}(\mathit{fail}) = o(1)$ *where* $\mathbb{P}_{\mathrm{A}}(\mathit{fail})$ *denotes the probability that one trial of Algorithm* A *fails.*

Therefore all graphs G are generated asymptotically with uniform probability. Note that this will also give an independent proof of McKay's formula [35] for number of graphs.

The running time of Algorithm A for regular graphs has been shown by Steger-Wormald [42] to be $\mathrm{O}(nd^2)$ per trial. The same argument shows for general degrees this running time is equal to $\mathrm{O}(md_{\max})$. Thus the expected running time of the Algorithm A is of $\mathrm{O}(md_{\max})$ since probability of reporting failure is $o(1)$. This completes proof of Theorem 1(a). ■

Remark 1 Without using any of the known formulas for $|\mathcal{L}(\bar{d})|$ Lemma 1 gives an upper bound for number of simple graphs with degree sequence $\bar{d}$. If $d_{\max} = O(m^{1/4-\epsilon})$ then

$$|\mathcal{L}(\bar{d})| \le e^{-\lambda(\bar{d})-\lambda^2(\bar{d})+o(1)} \frac{\prod_{r=0}^{m}\binom{2m-2r}{2}}{m!\prod_{i=1}^{n} d_i!}$$

Later in section 4.2 we will show the above inequality is in fact an equality.

4.1 Concentration inequality for random regular graphs

Let $\mathcal{L}(n,d)$ denotes the set of all simple d-regular graphs with $m = nd/2$ edges. Same as before let G be a fixed element of $\mathcal{L}(n,d)$ and $\mathcal{M}$ be a fixed matching on W with $G_{\mathcal{M}} = G$. The main goal is to show that for $d = o(n^{1/2-\epsilon})$ probability of generating G is at least $(1-o(1))$ of uniform; i.e. $\mathbb{P}_{\mathrm{A}}(G) = (d!)^n \sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \mathbb{P}(\mathcal{N}) \ge (1-o(1))/|\mathcal{L}(n,d)|$. Our proof builds upon the steps in Kim and Vu [32] and extends their bound via martingale approach. The following summarizes the analysis of Kim and Vu [32] for $d = O(n^{1/3-\epsilon})$. Let $m_1 = \frac{m}{d^2\omega}$ where ω goes to infinity very slowly; e.g. $\mathrm{O}(\log^{\delta} n)$ for small $\delta > 0$ then;

$$\begin{aligned}
|\mathcal{L}(n,d)|(d!)^n \sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \mathbb{P}(\mathcal{N}) &\overset{(a)}{=} \frac{1-o(1)}{m!} \sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \prod_{r=0}^{m-1} \frac{\binom{2m-2r}{2}-\mu_r}{\binom{2m-2r}{2}-\Delta(\mathcal{N}_r)} \\
&\overset{(b)}{\ge} \frac{1-o(1)}{m!} \sum_{\mathcal{N}\in \mathrm{S}(\mathcal{M})} \prod_{r=0}^{m_1} \left(1+\frac{\Delta(\mathcal{N}_r)-\mu_r}{\binom{2m-2r}{2}-\Delta(\mathcal{N}_r)}\right) \\
&\overset{(c)}{\ge} (1-o(1)) \prod_{r=0}^{m_1} \left(1-3\frac{T_r^1+T_r^2}{(2m-2r)^2}\right) \\
&\overset{(d)}{\ge} (1-o(1)) \exp\left(-3e\sum_{r=0}^{m_1} \frac{T_r^1}{(2m-2r)^2} - 3e\sum_{r=0}^{m_1} \frac{T_r^2}{(2m-2r)^2}\right) \qquad (2)
\end{aligned}$$

Here we explain these steps in more details. First define $\mu_r = \mu_r^1 + \mu_r^2$ where $\mu_r^1 = (2m-2r)^2(d-1)/4m$ and $\mu_r^2 = (2m-2r)^2(d-1)^2 r/4m^2$. The step (a) easily follows from equation (3.5) of [32] and McKay-Wormald's estimate [37] for regular graphs. Also algebraic calculations in page 10 of [32] justify (b) easily.

The main step is (c) which uses large deviations. For simplicity write Δ_r instead of $\Delta(\mathcal{N}_r)$ and let $\Delta_r = \Delta_r^1 + \Delta_r^2$ where Δ_r^1 and Δ_r^2 show the number of un-suitable pairs in step r corresponding to self edge loops and double edges respectively. For $p_r = r/m$, $q_r = 1-p_r$ Kim and Vu [32] used their polynomial inequality [30] to derive bounds T_r^1, T_2^r and to show with very high probability $|\Delta_r^1 - \mu_r^1| < T_r^1$ and $|\Delta_r^2 - \mu_r^2| < T_r^2$. More precisely for some constants c_1, c_2

$$T_r^1 = c_1 \log^2 n\sqrt{nd^2q_r^2(2dq_r+1)} \quad , \quad T_r^2 = c_2 \log^3 n\sqrt{nd^3q_r^2(d^2q_r+1)}$$

Now it can be shown easily that $\mu_r^i = o\left((2m-2r)^2\right)$ and $T_r^i = o\left((2m-2r)^2\right)$, $i = 1, 2$ which validates the step (c). The step (d) is straightforward using $1 - x \geq e^{-ex}$ for $1 \geq x \geq 0$.

Kim and Vu show that for $d = O(n^{1/3-\epsilon})$ the exponent in equation (2) is of $o(1)$. Using similar calculations as equation (3.13) in [32] it can be shown that in the larger region $d = O(n^{1/2-\epsilon})$ for $m_2 = (m \log^3 n)/d$:

$$\sum_{r=0}^{m_1} \frac{T_r^1}{(2m-2r)^2} = o(1) \quad , \quad \sum_{r=m_2}^{m_1} \frac{T_r^2}{(2m-2r)^2} = o(1)$$

But unfortunately the remaining $\sum_{r=0}^{m_2} \frac{T_r^2}{(2m-2r)^2}$ is of order $\Omega(d^3/n)$. In fact it turns out the random variable of Δ_r^2 has large variance in this range.

Let us explain the main difficulty for moving from $d = O(n^{1/3-\epsilon})$ to $d = O(n^{1/2-\epsilon})$. Note that Δ_r^2 is defined on a random subgraphs $G_{\mathcal{N}_r}$ of the graph G which has exactly r edges. Both Steger-Wormald [42] and Kim-Vu [31, 32] have approximated the $G_{\mathcal{N}_r}$ with G_{p_r} in which each edge of G appears independently with probability $p_r = r/m$. Our analysis shows that when $d = O(n^{1/2-\epsilon})$ this approximation causes the variance of Δ_r^2 to blow up.

In order to fix this problem we modify Δ_r^2 before moving to G_{p_r}. It can be shown via simple algebraic calculations that: $\Delta_r^2 - \mu_r^2 = X_r + Y_r$ where

$$X_r = \sum_{u \sim_{G_{\mathcal{N}_r}} v} [d_u^{(r)} - q_r(d-1)][d_v^{(r)} - q_r(d-1)] \quad , \quad Y_r = q_r(d-1)\sum_u \left[(d_u^{(r)} - q_r d)^2 - dp_r q_r\right].$$

The above modification is critical since the equality $\Delta_r^2 - \mu_r^2 = X_r + Y_r$ does not hold in G_{p_r}.

Next task is to find a new bound $\hat{T}_r^2$ such that $|X_r + Y_r| < \hat{T}_r^2$ with very high probability and

$$\sum_{r=0}^{m_2} \frac{\hat{T}_r^2}{(2m-2r)^2} = o(1).$$

It is easy to see that in G_{p_r} both X_r and Y_r have zero expected value. At this time we will move to G_{p_r} and show that X_r and Y_r are concentrated around zero. In the following we will show the concentration of X_r in details. For Y_r it can be done in exact same way.

Now look at the edge exposure martingale (page 94 of [3]) for the edges that are picked up to step r. That is for any $0 \leq \ell \leq r$ define $Z_\ell^r = \mathbb{E}(X_r \mid e_1, \ldots, e_\ell)$. For simplicity of notation let us drop the index r from $Z_\ell^r, d_u^{(r)}, p_r$ and q_r.

Next step is to bound the martingale difference $|Z_i - Z_{i-1}|$. Note that for $e_i = (u, v)$:

$$|Z_i - Z_{i-1}| \leq \left|\big(d_u - (d-1)q\big)\big(d_v - (d-1)q\big)\right| + \left|\sum_{u' \sim_{G_p} u} \big(d_{u'} - (d-1)q\big)\right| + \left|\sum_{v' \sim_{G_p} v} \big(d_{v'} - (d-1)q\big)\right|$$

Bounding the above difference should be done very carefully since the standard worst case bounds are very weak for our purpose. Note that the following observation is very crucial. For a typical ordering $\mathcal{N}$ the residual degree of the vertices are roughly $dq \pm \sqrt{dq}$. We will make this more precise. For any vertex $u \in G$ consider the following $L_u = \{|d_u - dq| \leq c \log^{1/2} n (dq)^{1/2}\}$ where $c > 0$ is a large constant. Then the following lemma holds:

Lemma 4 *For all $0 \leq r \leq m_2$ the following is true:* $\mathbb{P}(L_u^c) = o(\frac{1}{m^4})$.

Proof of Lemma 4 uses generalization of the Chernoff inequality from [3] and is given in the appendix.

To finish bounding the martingale difference we now look at the other terms in equation (3). For any vertex u consider the following event:

$$K_u = \left\{ \Big| \sum_{u' \sim_{G_p} u} \big(d_{u'} - (d-1)q\big) \Big| \leq c\left((dq)^{3/2} + qd + dq^{1/2}\right)\log n \right\}$$

where $c > 0$ is a large constant. We will use the following lemma to bound the martingale difference.

Lemma 5 *For all $0 \leq r \leq m_2$ the followings hold $\mathbb{P}(K_u^c) = o(\frac{1}{m^4})$.*

Proof For any vertex u let $N_G(u) \subset V(G)$ denotes neighbors of u in G. Set $A_G(u),\ B_G(u),\ C_G(u) \subset E(G)$ where $A_G(u)$ is the edges that are connected to u, $B_G(u)$ is those edges with both endpoints in $N_G(u)$ and $C_G(u)$ consists of the edges with exactly one endpoint in $N_G(u)$. For any edge e of G let $t_e = 1_{\{e \notin G_p\}}$. Then we can write

$$\begin{aligned}
\sum_{u' \sim_{G_p} u} \big(d_{u'} - (d-1)q\big) &= \sum_{u' \in N_{G_p}(u)} \ \sum_{e \in A_G(u') \setminus A_G(u)} (t_e - q) \\
&= \sum_{u' \in N_G(u)} \ \sum_{e \in A_G(u') \setminus A_G(u)} (t_e - q) - \sum_{u' \in N_G(u) \setminus N_{G_p}(u)} \ \sum_{e \in A_G(u') \setminus A_G(u)} (t_e - q) \\
&= \underbrace{\sum_{e \in C_G(u)} (t_e - q)}_{(i)} + 2\underbrace{\sum_{e \in B_G(u)} (t_e - q)}_{(ii)} - \underbrace{\sum_{u' \in N_G(u) \setminus N_G(u')} \big(d'_u - 1 - q(d-1)\big)}_{(iii)}
\end{aligned}$$

each of (i) and (ii) are sum of $\mathrm{O}(d^2)$ i.i.d. Bernoulli(q) random variables minus their expectations and similar to Lemma 4 can be shown to be less than $\mathrm{O}(\sqrt{12qd^2 \log n})$ with probability at least $1 - o(1/m^4)$. For (iii) we can say

$$\sum_{u' \in N_G(u) \setminus N_G(u')} \big(d'_u - 1 - q(d-1)\big) \leq d_u \max_{u' \in N_G(u) \setminus N_G(u')} \big(|d_{u'} - 1 - q(d-1)|\big)$$

now using Lemma 4 twice we can say (iii) is of $\mathrm{O}\big(\ [dq + \sqrt{12qd\log n}\ ]\sqrt{12qd\log n}\ \big)$ with probability at least $1 - o(1/m^4)$. These finish proof of Lemma 5. ■

Now we are ready to bound the martingale difference. Let $L = \bigcap_{r=0}^{m_2} \bigcap_{u=1}^{n} (L_u \cap K_u)$. Using Lemmas 4, 5 and the union bound $\mathbb{P}(L^c) = o(1/m^2)$. Hence the martingale difference is bounded by $|Z_i - Z_{i-1}| 1_L \leq O(dq + dq^{1/2} + (dq)^{3/2}) \log n$.

Next we state and use the following variation of the Azuma's inequality.

Proposition 1 (Kim [29]) *Consider a martingale $\{Y_i\}_{i=0}^n$ adaptive to a filtration $\{\mathcal{B}_i\}_{i=0}^n$. If for all k there are $A_{k-1} \in \mathcal{B}_{k-1}$ such that $\mathbf{E}[e^{\omega Y_k} | \mathcal{B}_{k-1}] \leq C_k$ for all $k = 1, 2, \cdots, n$ with $C_k \geq 1$ for all k, then*

$$\mathbf{P}(Y - \mathbf{E}[Y] \geq \lambda) \leq e^{-\lambda\omega} \prod_{k=1}^{n} C_k + \mathbf{P}(\cup_{k=0}^{n-1} A_k)$$

Proof of Theorem 1(b) Applying the above proposition for a large enough constant $c' > 0$ gives:

$$\mathbb{P}\left(|X_r| > c' \sqrt{6r \log^3 n \left(dq + d(q)^{1/2} + (dq)^{3/2}\right)^2} \right) \leq e^{-3\log n} + \mathbb{P}(L^c) = o(\frac{1}{m^2})$$

The same equation as above holds for Y_r since the martingale difference for Y_r is of $O(|dq_r(d_u - q_r d)|) = O((dq)^{3/2} \log^{1/2} n))$ using Lemma 4.

Therefore defining $\hat{T}_r^2 = c'(dq + d(q)^{1/2} + (dq)^{3/2})\sqrt{6r\log^3 n}$ we only need to show the following is $o(1)$.

$$\sum_{r=0}^{m_2} \frac{(dq + d(q)^{1/2} + (dq)^{3/2})\sqrt{6r\log^3 n}}{(2m-2r)^2}$$

But using $ndq = 2m - 2r$:

$$\begin{aligned}\sum_{r=0}^{m_2} \frac{\left(dq + dq^{1/2} + (dq)^{3/2}\right)\sqrt{6r\log^3 n}}{n^2d^2q^2} &= \sum_{r=0}^{m_2} O\left(\frac{d^{1/2}}{n^{1/2}(2m-2r)} + \frac{d}{(2m-2r)^{3/2}} + \frac{d^{1/2}}{n(2m-2r)^{1/2}}\right)\log^{1.5} n \\ &= O\Big(\frac{d^{1/2}\log(nd)}{n^{1/2}} + \frac{d}{(n\log^3 n)^{1/2}} + \frac{d}{n^{1/2}}\Big)\log^{1.5} n\end{aligned}$$

which is $o(1)$ for $d = O(n^{1/2-\epsilon})$. ■

4.2 Probability of failure

In this section we will show that probability of failure in one round of Algorithm A is very small. First we will characterize the degree sequence of the partial graph generated up to the time of failure. Then we apply the upper bound of Remark 1 from the beginning of Section 4 to derive an upper bound on probability of failure and show that it is $o(1)$.

Lemma 6 *If Algorithm* A *fails in step s then $2m - 2s \le d_{\max}^2 + 1$.*

Proof Algorithm A fails when there is no suitable pair left to choose. If the failure occurs in step s then number of unsuitable edges is equal to total number of possible pairs $\binom{2m-2s}{2}$. On the other it can be easily shown that the number of unsuitable edges at step s is at most $(2m-2s)/2d_{\max}^2$ (see Corollary 3.1 in [42] for more details). Therefore $2m - 2s \le d_{\max}^2 + 1$. ■

Failure in step s means there are some W_i's which have unmatched mini-vertices ($d_i^{(s)} \neq 0$). Let us call them "unfinished" W_i's. Since the algorithm fails, any two unfinished W_i's should be already connected. Hence there are at most $d_{\max}$ of them due to $\forall i: \ |W_i| = d_i \le d_{\max}$. The main idea is to show this is a very rare event. Without loss of generality assume $W_1, W_2, \ldots, W_k$ are all the unfinished sets. Above argument shows $k \le d_{\max}$ and by construction $k \le 2m - 2s$. The algorithm up to this step has created a partial matching $\mathcal{M}_s$ where graph $G_{\mathcal{M}_s}$ is simple with degree sequence $\bar{d}^{(s)} = (d_1 - d_1^{(s)}, \ldots, d_k - d_k^{(s)}, d_{k+1}, \ldots, d_n)$. Let $A_{d_1^{(s)},\ldots,d_k^{(s)}}$ denotes the above event of failure. Hence

$$\mathbb{P}(\text{fail}) = \sum_{2m-2s=2}^{d_{\max}^2+1} \sum_{k=1}^{\max(d_{\max},2m-2s)} \sum_{i_1,\ldots,i_k=1}^{n} \mathbb{P}_{\mathrm{A}}(A_{d_1^{(s)},\ldots,d_k^{(s)}}) \tag{3}$$

The following lemma is the central part of the proof.

Lemma 7 *Probability of the event that Algorithm* A *fails in step s and vertices $v_1, \ldots, v_k$ are the only unfinished vertices; i.e. $d_i^{(s)} \neq 0$ $i = 1, \cdots, k$ is at most*

$$(1 + o(1)) \frac{d_{\max}^{k(k-1)} \prod_{i=1}^k d_i^{d_i^{(s)}}}{m^{\binom{k}{2}}(2m)^{2m-2s}} \binom{2m-2s}{d_1^{(s)}, \ldots, d_k^{(s)}}$$

Proof By our notation the above event is denoted by $A_{d_1^{(s)},\ldots,d_k^{(s)}}$. Note that the above discussion shows the graph $G_{\mathcal{M}_s}$ should have a clique of size k on $v_1, \ldots, v_k$. Therefore number of such graphs should be less than $|\mathcal{L}(\bar{d}_k^{(s)})|$ where $\bar{d}_k^{(s)}$=$(d_1 - d_1^{(s)} - (k-1), \ldots, d_k - d_k^{(s)} - (k-1), d_{k+1}, \ldots, d_n)$. Thus $\mathbb{P}_{\mathrm{A}}(A_{d_1^{(s)},\ldots,d_k^{(s)}})$ is at most $|\mathcal{L}(\bar{d}_k^{(s)})|\ \mathbb{P}_{\mathrm{A}}(G_{\mathcal{M}_s})$. On the other hand we can use Remark 1 to derive an upper bound for $|\mathcal{L}(\bar{d}_k^{(s)})|$ because $m-s$ and k are small relative to m and the following can be easily shown $d_{\max} = O([s-\binom{k}{2}]^{1/4-\epsilon})$. The result of these gives

$$\mathbb{P}_{\mathrm{A}}(A_{d_1^{(s)},\ldots,d_k^{(s)}}) \leq \left(\frac{(2s-k(k-1))! \quad \exp\left[-\lambda(\bar{d}_k^{(s)}) - \lambda^2(\bar{d}_k^{(s)}) + o(1)\right]}{[s-\binom{k}{2}]! \quad 2^{s-\binom{k}{2}} \quad \prod_{i=1}^n (d_i^{(s)})!} \right) \mathbb{P}_{\mathrm{A}}(G_{\mathcal{M}_s})$$

The next step is to bound $\mathbb{P}_{\mathrm{A}}(G_{\mathcal{M}_s})$. We can use the same methodology as in the beginning of Section 4 to derive

$$\begin{aligned}\mathbb{P}_{\mathrm{A}}(G_{\mathcal{M}_s}) = \frac{\prod_{i=1}^n d_i!}{\prod_{i=1}^k \left[d_i^{(s)}\right]!} \sum_{\mathcal{N}_s \in S(\mathcal{M}_s)} \mathbb{P}_{\mathrm{A}}(\mathcal{N}_s) \quad &= \quad s!\ \exp\left(-\frac{\sum_{i\sim_{G_s} j} d_i d_j}{4m} + o(1)\right) \prod_{r=0}^{s-1} \frac{1}{\binom{2m-2r}{2} - \psi_r(G_{\mathcal{M}_s})} \\ &= \quad s!\ \exp\left(\frac{m}{s}\lambda(\bar{d}) + \frac{m^2}{s^2}\lambda^2(\bar{d}) + o(1)\right) \prod_{r=0}^{s-1} \frac{1}{\binom{2m-2r}{2}}\end{aligned}$$

Now using the following simple algebraic approximation

$$\frac{m}{s}\lambda(\bar{d}) + \frac{m^2}{s^2}\lambda^2(\bar{d}) - \lambda(\bar{d}_k^{(s)}) - \lambda^2(\bar{d}_k^{(s)}) = O\left(\lambda(\bar{d})\left[\lambda(\bar{d}) - \lambda(\bar{d}_k^{(s)})\right]\right) = O(\frac{d_{\max}^4}{m^2}) = o(1)$$

the following is true

$$\begin{aligned}\mathbb{P}_{\mathrm{A}}(A_{d_1^{(s)},\ldots,d_k^{(s)}}) \quad &\leq \quad e^{o(1)} \frac{[2s-k(k-1)]!\ (2m-2s)!\ s!\ 2^{\binom{k}{2}}\ \prod_{i=1}^k d_i!}{[s-\binom{k}{2}]!\ (2m)!\ \prod_{i=1}^k \left[(d_i^{(s)})!(d_i - k - d_i^{(s)} + 1)!\right]} \\ &\leq \quad e^{o(1)} \frac{\prod_{i=1}^k d_i^{d_i^{(s)}+k-1}}{\prod_{\ell=2s+1}^{2m} \ell\ \prod_{j=1}^{\binom{k}{2}} (2s-2j+1)} \binom{2m-2s}{d_1^{(s)}, \ldots, d_k^{(s)}} \qquad (4)\end{aligned}$$

Next using $m-s = O(d_{\max}^2)$ and $k = O(d_{\max})$ we can show $\prod_{j=1}^{\binom{k}{2}}(2s-2j+1) \geq m^{\binom{k}{2}}$ and $(1/m^{2m-2s}) \prod_{\ell=2s+1}^{2m} \ell \geq e^{-O(d_{\max}^4/m)}$. These two added to equation (4) finish proof of Lemma 7. ∎

Now we are ready to prove the main result of this section.

Proof of Lemma 3 First we show that the event of failure has probability of $o(1)$ for the case of only one unfinished vertex; i.e. $k = 1$. Lemma 7 for $k = 1$ simplifies to $\mathbb{P}_{\mathrm{A}}(A_{d_1^{(s)}}) = O\left((\frac{D}{m})^{2m-2s}\right)$ therefore summing over all possibilities of $k = 1$ gives

$$\sum_{2m-2s=2}^{d_{\max}^2+1} \sum_{i=1}^{n} \mathbb{P}_{\mathrm{A}}(A_{d_i^{(s)}}) = O\left(\sum_{2m-2s=2}^{d_{\max}^2+1} \frac{D^{2m-2s-1}}{m^{2m-2s-1}}\right) = O(\frac{d_{\max}}{m}) = o(1)$$

For $k > 1$ we use Lemma 7 differently. Using $D^{k(k-1)}/m^{\binom{k}{2}} \leq D^2/m$ and equation (3) we have

$$\mathbb{P}(\text{fail}) \leq o(1) + e^{o(1)} \frac{d_{\max}^2}{m} \sum_{2m-2s=2}^{d_{\max}^2+1} \frac{1}{(2m)^{2m-2s}} \underbrace{\sum_{k=2}^{\max(d_{\max}, 2m-2s)} \sum_{i_1,\ldots,i_k=1}^{n} \prod_{i=1}^{k} d_i^{d_i^{(s)}} \binom{2m-2s}{d_1^{(s)}, \ldots, d_k^{(s)}}}_{(a)}$$

Now note that the double sum (a) is at most $(d_1 + \ldots + d_n)^{2m-2s} = (2m)^{2m-2s}$ since $\sum_{i=1}^k d_i^{(s)} = 2m - 2s$. Therefore

$$\mathbb{P}(\text{fail}) \leq o(1) + e^{o(1)} \frac{d_{\max}^2}{m} \sum_{2m-2s=2}^{d_{\max}^2+1} 1 = O(\frac{d_{\max}^4}{m}) = o(1).$$

■

5 Acknowledgement

We would like to thank Joe Blitzstein, Persi Diaconis, Adam Guetz, Milena Mihail, Alistair Sinclair and Eric Vigoda for insightful discussions and useful comments on earlier version of this paper.

References

[1] W. Aiello, F. Chung and L. Lu, Random evolution in massive graphs, (2001), FOCS.

[2] B. Aiello, F. Chung and L. Lu, A random graph model for massive graphs, (2000) STOC.

[3] N. Alon and J. Spencer, The Probabilistic Method, (1992), Wiley, NY.

[4] D. Alderson, J. Doyle, and W. Willinger, Toward and Optimization-Driven Framework for Designing and Generating Realistic Internet Topologies, (2002) HotNets.

[5] A. Amraoui, A. Montanari and R. Urbanke, How to Find Good Finite-Length Codes: From Art Towards Science, (2006) Preprint, arxiv.org/pdf/cs.IT/0607064.

[6] F. Bassetti, P. Diaconis, Examples Comparing Importance Sampling and the Metropolis Algorithm, (2005).

[7] M. Bayati, J.H. Kim and A. Saberi, A sequential algorithm for generated random graphs, Extended abstract, (2006), http://www.stanford.edu/∼bayati/randomgraphextended.pdf.

[8] M. Bayati, A. Montanari, B. Prabhakar and A. Saberi, (2006), Work in progress.

[9] E.A. Bender and E.R. Canfield, The asymptotic number of labeled graphs with given degree sequence, *J. Combinatorial Theory Ser.* (1978), A 24, 3, 296-307.

[10] I. Bezáková, N. Bhatnagar and E. Vigoda Sampling Binary Contingency tables with a Greedy Start, (2006), SODA.

[11] I. Bezáková, A. Sinclair, D. Štefankovič and E. Vigoda Negative Examples for Sequential Importance Sampling of Binary Contingency Tables, (2006).

[12] J. Blitzstein and P. Diaconis, A sequential importance sampling algorithm for generating random graphs with prescribed degrees, (2005), Submitted.

[13] B. Bollobás, A probabilistic proof of an asymptotoic forumula for the number of labelled regular graphs, (1980), *European J. Combin.* 1, 4, 311-316.

[14] B. Bollobás, Random Graphs, (2001), Cambrdige University Press.

[15] T. Britton, M. Deijfen and A. Martin-Löf ,Generating simple random graphs with prescribed degree distribution, (2005), Preprint.

[16] T. Bu and D. Towsley, On Distinguishing between Internet Power Law Topology Generator, (2002) INFOCOM.

[17] Y. Chen, P. Diaconis, S. Holmes and J.S. Liu, Sequential Monte Carlo methods for statistical analysis of tables, (2005), *Journal of the American Statistical Association* 100, 109-120.

[18] C. Cooper, M. Dyer and C. Greenhill, Sampling regular graphs and peer-to-peer network, (2005), *Combinatorics, Probability and Computing* to appear.

[19] F. Chung and L. Lu, Conneted components in random graphs with given expected degree sequence, (2002), *Ann. Comb.* 6, 2, 125-145.

[20] P. Diaconis and A. Gangolli, Rectangular arrays with fixed margins, (1995), *Discrete probability and algorithms (Minneapolis, MN, 1993)* . IMA Vol. Math. Appl., Vol 72. Springer, New York, 15-41.

[21] R. Durret, Random Graph Dynamics, (2006), Cambridge University Press.

[22] M. Faloutsos, P. Faloutsos and C. Faloutsos, On Power-law Relationships of the Internet Topology, (1999) SIGCOM.

[23] C. Gkantsidis, M. Mihail and E. Zegura, The Markov Chain Simulation Method for Generating Connected Power Law Random Graphs, (2003) Alenex.

[24] M. Jerrum and A. Sinclair, Approximate counting, uniform generation and rapidly mixing Markov chains, (1989), *Inform. and Comput.* 82 , no. 1, 93–133.

[25] M. Jerrum and A. Sinclair, Fast uniform generation of regular graphs, (1990), *Theoret. Comput. Sci.* 73, 1, 91-100.

[26] M. Jerrum, A. Sinclair and B. McKay, When is a graphical sequence stable?, (1992) *Random graphs Vol 2 (Poznań 1989)* Wiley-Intersci. Publ. Wiley, New York, 101-115.

[27] M. Jerrum, A. Sinclair and E. Vigoda, A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Non-Negative Entries, (2004) *Journal of the ACM*, 51(4):671-697.

[28] R. Kannan, P. Tetali and S. Vempala, Simple Markov chain algorithms for generating bipartite graphs and tournaments, (1992) *Random Structures and Algorithms* (1999) 14, 293-308.

[29] J.H.Kim , On Brooks' Theorem for Sparse Graphs, *Combi. Prob. & Comp.*, (1995) 4, 97-132.

[30] J. H. Kim and V. H. Vu, Concentration of multivariate polynomials and its applications, (2000) *Combinatorica* 20, no 3, 417-434.

[31] J. H. Kim and V. H. Vu, Generating Random Regular Graphs, , STOC 2003 213-222.

[32] J.H.Kim and V. Vu, Sandwiching random graphs, *Advances in Mathematics* (2004) 188, 444-469.

[33] D. Knuth, Mathematics and computer science: coping with finiteness, (1976) *Science* 194(4271):1235-1242.

[34] A. Medina, I. Matta and J. Byers, On the origin of power laws in Internet topologies, *ACM Computer Communication Review*, (2000), vol. 30, no. 2, pp. 18-28.

[35] B. McKay , Asymptotics for symmetric 0-1 matrices with prescribed row sums, (1985), *Ars Combinatorica* 19A:15-25.

[36] B. McKay and N. C. Wormald, Uniform generation of random regular graphs of moderate degree, (1990b), *J. Algorithms* 11, 1, 52-67.

[37] B. McKay and N. C. Wormald, Asymptotic enumeration by degree sequence of graphs with degrees o($n^{1/2}$), (1991), *Combinatorica* 11, 4, 369-382.

[38] R. Milo, N. Kashtan, S. Itzkovitz, M. Newman and U. Alon, On the uniform generation of random graphs with prescribed degree sequences, (2004), http://arxiv.org/PS_cache/cond-mat/pdf/0312/0312028.pdf

[39] R. Milo, S. ShenOrr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon, Network motifs: Simple building blocks of complex networks, (2002), *Science* 298, 824-827

[40] M. Molloy and B. Reed, A critical point for random graphs with a given degree sequence, (1995), *Random Structures and Algorithms* 6, 2-3, 161-179.

[41] M. Molloy and B. Reed, The size of the ginat component of a random graph with a given degree sequence, (1998), *Combin. robab. Comput.* 7, 3, 295-305.

[42] A. Steger and N.C. Wormald, Generating random regular graphs quickly, (English Summary) Random graphs and combinatorial structures (Oberwolfach, 1997), *Combin. Probab. Comput.* 8, no. 4, 377-396.

[43] H.Tangmunarunkit, R.Govindan, S.Jamin, S.Shenker, and W.Willinger, Network Topology Generators: Degree based vs. Structural, (2002), ACM SIGCOM.

[44] N. C. Wormald, Models of random regular graphs, (1999), *Surveys in combinatorics (Canterbury)* London Math. Soc. Lecture Notes Ser., Vol 265. Cambridge Univ. Press, Cambridge, 239-298.

[45] V. H. Vu, Concentration of non-Lipschitz functions and applications, Probabilistic methods in combinatorial optimization, *Random Structures Algorithms* 20 (2002), no. 3, 267-316.

APPENDIX

Generalized Chernoff inequality and Proof of Lemma

Note that in the G_p model the residual degree of a vertex u, d_u, is sum of d independent Bernoulli random variables with mean q. Two generalizations of the Chernoff inequality (Theorems A.1.11, A.1.13 in page 267 of [3]) for state that for $a > 0$ and $X_1, \ldots, X_d$ i.i.d. Bernoulli(q) random variables:

$$\mathbb{P}(X_1 + \cdots + X_d - qd \geq a) < e^{-\frac{a^2}{2qd} + \frac{a^3}{2(qd)^2}} \quad , \quad \mathbb{P}(X_1 + \cdots + X_d - qd < -a) < e^{-\frac{a^2}{2qd}}$$

.

Applying these two for $a = \sqrt{12qd \log n}$ Lemma 4 directly follows. ■

Algebraic proof of the equation (1)

First of all note that it is easy to see ψ_r =O$(\frac{d_{\max}^2(2m-2r)^2}{m})$. Now the following is the proof:

$$\begin{aligned}
\prod_{r=0}^{m-1} \frac{\binom{2m-2r}{2}}{\binom{2m-2r}{2} - \psi_r} &= \prod_{r=0}^{m-1} \Big(1 + \frac{\psi_r}{\binom{2m-2r}{2} - \psi_r}\Big) \\
&= \prod_{r=0}^{m-1} \Big(1 + \frac{\frac{\lambda(\bar{d})}{m} + \frac{r\sum_{i\sim_G j}(d_i-1)(d_j-1)}{2m^3} + \frac{(\sum_i d_i^2)^2}{16m^3} + O(\frac{d_{\max}^4}{m^2} + \frac{d_{\max}^2}{m(2m-2r)})}{1 - \frac{1}{2m-2r} - O(\frac{d_{\max}^2}{m})}\Big) \\
&= \exp\Big[\sum_{r=0}^{m-1} \log\Big(1 + \frac{\frac{\lambda(\bar{d})}{m} + \frac{r\sum_{i\sim_G j}(d_i-1)(d_j-1)}{2m^3} + \frac{(\sum_i d_i^2)^2}{16m^3} + O(\frac{d_{\max}^4}{m^2} + \frac{d_{\max}^2}{m(2m-2r)})}{1 - \frac{1}{2m-2r} - O(\frac{D^2}{m})}\Big)\Big] \\
&= \exp\Big[\sum_{r=0}^{m-1} \log\Big(1 + \frac{\lambda(\bar{d})}{m} + \frac{r\sum_{i\sim_G j}(d_i-1)(d_j-1)}{2m^3} + \frac{(\sum_i d_i^2)^2}{16m^3} + O(\frac{d_{\max}^4}{m^2} + \frac{d_{\max}^2}{m(2m-2r)})\Big)\Big] \\
&= \exp\Big[\sum_{r=0}^{m-1} \Big(\frac{\lambda(\bar{d})}{m} + \frac{r\sum_{i\sim_G j}(d_i-1)(d_j-1)}{2m^3} + \frac{(\sum_i d_i^2)^2}{16m^3} + O(\frac{d_{\max}^4}{m^2} + \frac{d_{\max}^2}{m(2m-2r)})\Big)\Big] \quad (5) \\
&= \exp\Big[\lambda(\bar{d}) + \frac{m(m-1)\sum_{i\sim_G j}(d_i-1)(d_j-1)}{4m^3} + \frac{(\sum_i d_i^2)^2}{16m^2} + O(\frac{d_{\max}^2}{m}\log(2m) + \frac{d_{\max}^4}{m})\Big] \\
&= \exp\Big[\lambda(\bar{d}) + \frac{\sum_{i\sim_G j}(d_i-1)(d_j-1)}{4m} + \frac{(\sum_i d_i^2)^2}{16m^2} + O(\frac{d_{\max}^2}{m} + \frac{d_{\max}^2}{m}\log(2m) + \frac{d_{\max}^4}{m})\Big] \\
&= \exp\Big[\lambda(\bar{d}) + \frac{\sum_{i\sim_G j} d_i d_j}{4m} - \frac{\sum_{i\sim_G j}(d_i + d_j)}{4m} + \frac{1}{4} + \frac{(\sum_i d_i^2)^2}{16m^2} + o(1)\Big] \quad (6) \\
&= \big(1 + o(1)\big)\exp\Big[\lambda(\bar{d}) + \lambda^2(\bar{d}) + \frac{\sum_{i\sim_G j} d_i d_j}{4m}\Big] \quad (7)
\end{aligned}$$

where (5) uses $\log(1+x) = x - O(x^2)$ and (6) uses $d_{\max} = O(m^{1/4-\epsilon})$. We also used $\frac{\psi_r}{(2m-2r)^2} = O(\frac{d_{\max}^2}{m})$ quite often. ■