

# Generating Random Graphs with Large Girth

Mohsen Bayati\*

Andrea Montanari†

Amin Saberi‡

## Abstract

We present a simple and efficient algorithm for randomly generating simple graphs without small cycles. These graphs can be used to design high performance Low-Density Parity-Check (LDPC) codes. For any constant  $k$ ,  $\alpha \leq 1/2k(k+3)$  and  $m = O(n^{1+\alpha})$ , our algorithm generates an asymptotically uniform random graph with  $n$  vertices,  $m$  edges, and girth larger than  $k$  in polynomial time. To the best of our knowledge this is the first polynomial algorithm for the problem.

Our algorithm generates a graph by sequentially adding  $m$  edges to an empty graph with  $n$  vertices. Recently, this type of *sequential* process has been very successful for efficiently counting and generating random graphs [35, 18, 11, 7, 5, 6].

## 1 Introduction

We present efficient algorithms for generating random simple graphs with cycles of size larger than a constant  $k$ . The main motivation for this work comes from the design of high performance Low-Density Parity-Check (LDPC) codes [31].

For positive integers  $m, n, k$ , our algorithm generates a random graph with  $n$  vertices and  $m$  edges that has no cycles with length less than or equal to  $k$  using  $O(n^2m)$  operations. The algorithm starts with an empty graph and sequentially adds  $m$  edges between pairs of non-adjacent vertices. In every step of the procedure, an edge can be added between two distinct vertices  $i$  and  $j$  that are of distance at least  $k$ . The probability of adding an edge between  $i$  and  $j$ , denoted by  $p_{ij}$ , changes in every step of the algorithm. In order to get a uniform sampling,  $p_{ij}$  should be proportional to the number of extensions of the current graph to graphs with  $m$  edges that contain  $(ij)$  and have no small cycles. The algorithm estimates the number of such valid extensions by computing the expected number of small cycles if the rest of edges are added uniformly at random (see Sections 3 and 4 for more details).

We show that our algorithm produces an asymptotically uniform sample when  $k$  is a constant and  $m = O(n^{1+\alpha})$  with  $\alpha < 1/2k(k+3)$ . To the best of

our knowledge this is the first polynomial algorithm for the problem. The analysis uses Janson's inequality for obtaining a close approximation of the number valid extensions of a partially constructed graph. This method goes beyond the Poisson approximations that is typically used when the number of the edges is linear (see [2] for more details).

From a theoretical perspective, our work is related to the following problem. Given a graph property  $P$  that is preserved by removal of any edge from the graph. A *random maximal  $P$ -graph* is obtained from  $n$  isolated vertices by randomly adding those edges (at each stage choosing uniformly among edges whose inclusion would not destroy property  $P$ ) until no further edges can be added. The question of finding the number of edges of a random maximal  $P$ -graph for several properties  $P$  is well studied [33, 16, 38, 8, 27]. In particular, when  $P$  is the property that the graph has girth greater than  $k$ , [27] shows that the above process of sequentially growing the graph leads to graphs with  $m = O(n^{1+\frac{1}{k-1}} \log n)$  edges.

Unfortunately, these random maximal  $P$ -graphs may have distribution that are far from uniform. In fact it has been shown (e.g. [36]) that when  $P$  is the property of having no triangle, the maximal triangle free graphs are close to bipartite. We show that our new algorithm guarantees asymptotically uniform distribution at the expense of reducing the number of edges to  $m = O(n^{1+\frac{1}{2k(k+3)}})$ .

Recently, sequential algorithms have been shown, empirically ([11, 7]) and theoretically ([35, 18, 6, 5]), to be very successful for designing fast algorithms for counting and generating random graphs with given degrees. The current paper builds on this line of research and develops mainly two new ideas. For the design, instead of starting from a biased heuristic and calculating and canceling its bias we use Poisson approximations to obtain the correct  $p_{ij}$ 's. For the analysis, we use convex functions and Janson's inequality for controlling the accumulated error and obtaining a tighter bound than the concentration results used in [5, 18].

### 1.1 Application in designing LDPC Codes

It has been shown that LDPC codes can approach Shan-

\*Microsoft Research New England; mohsenb@microsoft.com.

†Department of Electrical Engineering, Stanford University; montanar@stanford.edu.

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

non capacity asymptotically for large size codes, when their associated graph representations (Tanner graphs) are selected uniformly at random from the set of bipartite graphs with a properly optimized degree sequences [12, 21]. However, in practice, the maximum graph size is between  $10^3$  and  $10^5$  (depending on the delay sensitivity and on the hardware constraints). In this range, it is well known that the existence of a small number of subgraphs with a certain structure (in particular, small cycles) spoil the code performances [28, 30, 20].

Different approaches have been developed within the coding theory community to deal with this problem. For example, deterministic constructions of graph sequences with large girth [26, 32] have been studied. However, numerical studies have shown that known deterministic constructions can have poor performances [22]. From a theoretical point of view, no deterministic graph sequence is known that asymptotically outperforms random graphs.

One can also stick to random constructions and grow the graph by adding random edges sequentially while avoiding short cycles. This method has been very popular in practice and is known by the name of *progressive edge growth* (PEG) algorithm [14]. We will describe the main intuition behind PEG and show its limitations with respect to two standard performance measures for the codes: (i) bit error rate or expected fraction of wrong bits; and (ii) block error rate or probability that at least one bit in the message was received incorrectly.

Let  $C_{\text{iter}}$  be the maximum rate achievable by random LDPC codes (empirically  $C_{\text{iter}}$  is indistinguishable from the channel capacity). It is known that uniformly random graphs contain a random number (of order  $O_n(1)$ ) of cycles of size  $k$  or smaller. These cycles are responsible for non-vanishing block error probability that is bounded away from 0 at small noise. The main goal of PEG is to reduce this error, to a value that vanishes with  $k$ , by removing the cycles of length up to  $k$ . But the final distribution of PEG is not necessarily uniform which may affect the other performance measure (bit error rate). In fact preliminary simulations suggest that our new algorithm produces codes with lower bit error rate.

In this paper we define the first code generation algorithm that overcomes both problems. We show that there exists a graph sequence that (1) can be generated efficiently; (2) has vanishing bit error rate at any rate below  $C_{\text{iter}}$  (this follows by the standard density evolution analysis [31] using optimized degree sequences [12, 21]); and (3) has girth larger than  $k$  (therefore has low block error rate probability).

For the sake of simplicity we will present the rele-

vant calculations only for the problem of generating random (not necessarily bipartite) graphs that have large girth. Then for LDPC design, we will define the algorithm for generating bipartite graphs with given degree sequences. Generalizing proofs to this case is cumbersome but is conceptually straightforward.

**1.2 Organization of the Paper** The rest of the paper is organized as follows. In Section 2, we provide the setup and necessary definitions. In Section 3, we describe the new algorithm for randomly generating large-girth graphs, and state the main result. In Section 4 we explain the intuition behind the algorithm, and present its application to LDPC codes in Section 5. Finally the analysis of the algorithm and its running time are discussed in Section 6. To the interest of space some of the proofs are presented in the longer version of the paper [4].

## 2 Definitions and Problem Statement

The *girth* of a graph  $G$  is defined to be the length of its shortest cycle. Let  $\mathbb{G}_{n,m}$  denote the set of all simple graphs with  $m$  edges over  $n$  vertices and let  $\mathbb{G}_{n,m,k} \subset \mathbb{G}_{n,m}$  be the set of graphs with girth greater than  $k$ . In this paper we assume that  $k$  is a constant. We want to sample a uniformly random graph  $G$  from  $\mathbb{G}_{n,m,k}$ . From a theoretical point of view, this is a relatively easy problem for  $m = O(n)$  and in this paper our focus is on the difficult case of  $m = O(n^{1+\alpha})$  for some small  $\alpha > 0$ .

A naïve approach would be to start with an empty graph  $G_0$  with no edges over the vertex set  $V = \{1, 2, \dots, n\}$  and sequentially add pairs  $(ij)$  chosen uniformly at random (without replacement) among all pairs  $(ij)$  that are not yet selected. Repeating the edge addition  $m$  times leads to a uniformly random member of  $\mathbb{G}_{n,m}$  that has girth larger than  $k$ , with probability bounded away from 0. Re-running the whole  $m$  steps a sufficiently large number of times, yields a polynomial algorithm for uniformly randomly generating graphs in  $\mathbb{G}_{n,m,k}$  when  $m = O(n)$ . This approach does not work for the case of  $m = O(n^{1+\alpha})$  since the success probability is of the order  $e^{-n^\alpha}$ . For the applications to coding theory, even the case  $m = O(n)$  is very challenging in practice since  $n$  is very large.

## 3 Algorithm and Main Result

Our sequential algorithm to sample from  $\mathbb{G}_{n,m,k}$  works as follows.

---

### Algorithm S

---

- (1) Set  $G_0$  to be the graph over vertex set  $V =$

$\{1, 2, \dots, n\}$  and with no edges. For  $t = 0, \dots, m - 1$ , repeat the following steps:

- If there is no edge  $(ij)$  such  $G_t \cup (ij) \in \mathbb{G}_{n,t+1,k}$  (adding any edge to  $G_t$  creates a small cycle), stop and return FAIL.
- Otherwise, consider the probability distribution  $p(ij|G_t)$  given by equation (3.1) below on the set of all edges  $(ij)$ . Sample an edge  $(ij)$  with distribution  $p(ij|G_t)$  and set  $G_{t+1} = G_t \cup (ij)$ .

- (2) If the algorithm does not FAIL before  $t = m - 1$ , return  $G_m$ .

Let  $M_t$  and  $M_t^c$  be the adjacency matrix of the partially constructed graph  $G_t$  and its complement  $G_t^c$  respectively. Let also  $S_t$  be the adjacency matrix of all edges  $(ij)$  such that  $G_t \cup (ij) \in \mathbb{G}_{n,t+1,k}$ . Define a probability distribution on the set of all pairs  $(ij)$  and represent it by a symmetric matrix  $P_{G_t} = [p(ij|G_t)]$  defined by

$$(3.1) \quad P_{G_t} = \frac{1}{Z(G_t)} S_t \odot \widehat{\exp} \left[ - \sum_{a=2}^{k-1} \left( M_t + \frac{m-t}{\binom{n}{2} - t} M_t^c \right)^a \right].$$

Here  $Z(G_t)$  is a normalization constant. Symbols  $\odot$  and  $\widehat{\exp}$  represent the coordinate-wise multiplication and exponentiation of square matrices. i.e. for  $n \times n$  matrices  $A, B, C$  the expression  $A = B \odot C$  ( $A = \widehat{\exp}(B)$ ) means that for all  $1 \leq i, j \leq n$  we have  $a_{ij} = b_{ij} c_{ij}$  ( $a_{ij} = e^{b_{ij}}$ ).

The main intuition for defining the probabilities  $p(ij|G_t)$  through matrix multiplication is explained in Section 4.

By construction  $G_m \in \mathbb{G}_{n,m,k}$ . Let  $\mathbb{P}_S(G)$  denote the probability that algorithm S does not FAIL and returns a graph  $G \in \mathbb{G}_{n,m,k}$ . Let also  $\mathbb{P}_U$  denote the uniform probability on the set  $\mathbb{G}_{n,m,k}$ ; i.e.  $\mathbb{P}_U(G) = |\mathbb{G}_{n,m,k}|^{-1}$ . Our goal is to show that  $\mathbb{P}_S(G)$  and  $\mathbb{P}_U(G)$  are very close to each other and for that we will use the total variation metric.

**Definition** The *total variation distance* between two (not necessarily normalized) measures  $\mathbb{P}$  and  $\mathbb{Q}$  on a set  $S$  is defined by:

$$d_{TV}(\mathbb{P}, \mathbb{Q}) \equiv \sup \{ |\mathbb{P}(A) - \mathbb{Q}(A)| : A \subset S \}$$

Our main result is the following:

**THEOREM 3.1.** For  $m = n^{1+\alpha}$  and constant  $k \geq 3$  such that  $\alpha \leq [2k(k+3)]^{-1}$ , the failure probability of algorithm S is  $o(1)$ . Moreover, the algorithm generates all

but  $o(|\mathbb{G}_{n,m,k}|)$  graphs in  $\mathbb{G}_{n,m,k}$  with equal probability asymptotically. In particular we will show that:

$$\lim_{n \rightarrow \infty} d_{TV}(\mathbb{P}_S, \mathbb{P}_U) = 0.$$

We will prove Theorem 3.1 in Section 6. We will also discuss an implementation of algorithm S in Subsection 6.2 that has an expected running time of  $O(n^2m)$ .

#### 4 The Intuition Behind Algorithm S

Define the *execution tree*  $T$  of the naïve algorithm described in Section 2 as follows. Consider a rooted  $m$ -level tree where the root (the vertex in level zero) corresponds to the empty graph in the beginning of the algorithm and level  $r$  vertices correspond to all couples  $(G_r, \pi_r)$  where  $G_r$  is a partial graph that can be constructed after  $r$  steps, and  $\pi_r$  is an ordering of its  $r$  edges. There is a link in  $T$  between a partial graph  $(G_r, \pi_r)$  from level  $r$  to a partial graph  $(G_{r+1}, \pi_{r+1})$  from level  $r + 1$  if  $G_r \subset G_{r+1}$  and  $\pi_r, \pi_{r+1}$  coincide on the first  $r$  positions. Any path from the root to a leaf at level  $m$  of  $T$  corresponds to one possible way of generating a random graph in  $\mathbb{G}_{n,m}$ .

Let us denote those partial graphs  $G_r$  that have girth greater than  $k$  by “valid” graphs. Our goal is to reach a valid leaf in  $T$ , uniformly at random, by starting from the root and going down the tree. It is known that [34] in order to achieve this goal, at any step  $r$ , one needs to choose  $G_{r+1} = G_r \cup \{(ij)\}$  with probability proportional to the number of valid leaves of  $T$  that are descendant of  $(G_{r+1}, \pi_{r+1})$  (see [5] for a similar analysis in more details). Denote this probability by  $p(G_{r+1}, \pi_{r+1})$ .

The main technical contribution of this paper is deriving a new approximation  $\hat{p}(G_{r+1}, \pi_{r+1})$  for the true probabilities  $p(G_{r+1}, \pi_{r+1})$ , selecting  $(G_{r+1}, \pi_{r+1})$  with probability  $\hat{p}(G_{r+1}, \pi_{r+1})$ , and controlling the accumulated error  $\prod_{r=0}^{m-1} [\hat{p}(G_{r+1}, \pi_{r+1})/p(G_{r+1}, \pi_{r+1})]$ .

Consider a random variable  $n_k(G_{r+1}, \pi_{r+1})$  that is the number of cycles of length at most  $k$  in a leaf chosen uniformly at random from the descendants of  $(G_{r+1}, \pi_{r+1})$  in  $T$ . First we use Janson’s inequality [2] to show that the distribution of  $n_k(G_{r+1}, \pi_{r+1})$  behaves like Poisson. That is the probability of  $n_k(G_{r+1}, \pi_{r+1}) = 0$  (i.e. reaching a valid leaf) is approximately  $\exp(-\mathbb{E}[n_k(G_{r+1}, \pi_{r+1})])$ . That explains how  $\hat{p}(G_{r+1}, \pi_{r+1})$  is calculated.

Furthermore, instead of calculating the error  $[\hat{p}(G_{r+1}, \pi_{r+1})/p(G_{r+1}, \pi_{r+1})]$  for each  $r$  and then bounding the accumulated error, we look at the final product, simplify it, and then find a bound for it. For this problem, the standard method of bounding each term separately, leads to much larger error terms and

is not sufficient for deriving our final result. In order to make the implementation easier, it is not hard to see that the terms  $\mathbb{E}[n_k(G_{r+1}, \pi_{r+1})]$  can be approximated by matrix multiplication (more precisely by entries of the matrix  $\sum_{\ell=1}^k \left( M_t + \frac{m-t}{\binom{2}{2}-t} M_t^c \right)^\ell$  used in equation (3.1)). These claims will be rigorously proven in Section 6. We believe that this idea is applicable for the analysis of similar random generation algorithms beyond their existing correctness ranges.

## 5 Application to LDPC codes

The ideas described above can be used to generate random bipartite graphs with given degree sequences. Such graphs define the standard ensemble for irregular LDPC codes. Here we will show how to modify the algorithm for this application without repeating its cumbersome, but conceptually simple, analysis.

Consider two sequences of positive integers  $\bar{r} = r_1, \dots, r_n$  and  $\bar{c} = c_1, \dots, c_m$  for degrees of the vertices such that  $e = \sum_{i=1}^n r_i = \sum_{j=1}^m c_j$ . We would like to generate a random bipartite graph  $G(V_1, V_2)$  with degree sequence  $(\bar{r}, \bar{c})$ , (i.e. for  $V_1 = \{u_1, \dots, u_n\}$ ,  $V_2 = \{v_1, \dots, v_m\}$  we need  $\deg(u_i) = r_i$  and  $\deg(v_j) = c_j$ ) that has also girth greater than  $k$  (assume  $k$  is an even number). We denote the set of all such graphs by  $\mathbb{G}_{\bar{r}, \bar{c}, k}$ . The algorithm is a natural generalization of Algorithm S where the probabilities  $p(ij|G_t)$  are adjusted to obtain the elements of set  $\mathbb{G}_{\bar{r}, \bar{c}, k}$ .

---

### Algorithm Bip-S

---

- (1) Set  $G_0$  to be a graph over vertex sets  $V_1 = \{u_1, \dots, u_n\}$ ,  $V_2 = \{v_1, \dots, v_m\}$  and with no edges. Let also  $\hat{r} = \{\hat{r}_1, \dots, \hat{r}_n\}$  and  $\hat{c} = \{\hat{c}_1, \dots, \hat{c}_m\}$  be two ordered set of integers that are initialized by  $\hat{r} = \bar{r}$  and  $\hat{c} = \bar{c}$ . For  $t = 0, \dots, e - 1$ , repeat the following steps:
    - If there is no *suitable* edges, i.e. For any edge  $(u_i v_j)$ , the graph  $G_t \cup (u_i v_j)$  has a cycle of length at most  $k$ , stop and return FAIL.
    - Consider the probability distribution  $q(u_i v_j | G_t)$  given by equation (5.2) below on the set of all edges  $(u_i v_j)$ . Sample an edge  $(u_i v_j)$  with distribution  $q(u_i v_j | G_t)$  and set  $G_{t+1} = G_t \cup (u_i v_j)$ .
  - (2) If the algorithm does not halt before  $t = e - 1$ , return  $G_e$ .
- 

Here each probability  $q(u_i v_j | G_t)$  is an approximation to the probability that a uniformly random extension of

graph  $G_t$  has girth larger than  $k$ . In order to find this approximation, we will consider a configuration model representation for the graphs with degree sequence  $(\bar{r}, \bar{c})$  (see [9] for the definition of configuration model). Then we use the argument of Section 4, to find the following Poisson approximation for  $q(u_i v_j | G_t)$ :

$$(5.2) \quad q(u_i v_j | G_t) = \frac{\hat{r}_i \hat{c}_j e^{-E_k(G_t, u_i v_j)}}{Z(G_t)}$$

where  $Z(G_t)$  is a normalization constant, and  $\hat{r}_i, \hat{c}_j$  denote the remaining degrees of  $i$  and  $j$ . Furthermore,  $E_k(G_t, u_i, v_j) = \sum_{r=1}^{k/2} \sum_{\gamma \in \mathbb{C}_{2r}} \mathbb{I}_{\{(u_i v_j) \in \gamma\}} a(\gamma, G_t, u_i v_j)$  where  $\mathbb{C}_{2r}$  is the set of all simple cycles of length  $2r$  in the complete bipartite graph on vertices of  $V_1$  and  $V_2$ , and  $a(\gamma, G_t, u_i v_j)$  is roughly the probability that  $\gamma$  is in a random extension of  $G_t$ . More precisely  $a(\gamma, G_t, u_i v_j)$  is a product of the three terms  $\frac{(e-t-2r+|\gamma \cap G_t|)!}{(e-t-1)!}$ ,  $\prod_{u_\ell \in \gamma} b(u_\ell, \gamma, G_t, u_i v_j)$ , and  $\prod_{v_\ell \in \gamma} b(v_\ell, \gamma, G_t, u_i v_j)$  where  $b(u_\ell, \gamma, G_t, u_i v_j)$  is equal to  $1, \hat{r}_\ell$ , or  $\hat{r}_\ell(\hat{r}_\ell - 1)$  depending on whether both, one, or none of the adjacent edges to  $u_\ell$  in  $\gamma$  are in  $G_t \cup (u_i v_j)$  respectively.  $b(v_s, \gamma, G_t, u_i v_j)$  is defined similarly.

We defer a more complete discussion of the codes generated by this algorithm to a complete version of the paper. Here we limit ourselves to a few remarks: (1) Several definitions have been proposed for the substructures responsible for the decoding errors at high signal-to-noise ratio. Our algorithm can be adapted to exclude any of these substructures (instead of cycles) as well. (2) In any of these definitions, the cycles play a dominant role. Therefore the above algorithm should be a good starting point. (3) In practical code design it can be preferable to partially structure the ensemble for facilitating the layout (as, for instance, in protograph codes [31]). Our graph generation procedure can be adapted to partially structured ensembles as well.

## 6 Analysis

The aim of this section is to prove Theorem 3.1. The most challenging part of the proof is to show that  $\mathbb{P}_S(G)$ , probability of generating a graph  $G$  by Algorithm S, is not less than  $\mathbb{P}_U(G)$ , the uniform probability, for almost all graphs  $G$  in  $\mathbb{G}_{n,m,k}$ . In fact, the two parts of the proof of Theorem 3.1 are given below and Subsection 6.1 is completely dedicated to the first one. There is also a brief analysis of the running time of the algorithm which is discussed in Subsection 6.2.

**Lower bound for  $\mathbb{P}_S$ .** The probability of generating a typical graph in  $\mathbb{G}_{n,m,k}$  is at least a constant fraction of the uniform probability. More precisely we will show that for all but  $o(|\mathbb{G}_{n,m,k}|)$  graphs  $G \in \mathbb{G}_{n,m,k}$ :  $\mathbb{P}_S(G) \geq (1 - o(1))\mathbb{P}_U(G)$  where the term

$o(1)$  goes to zero quickly as  $n$  goes to infinity. Proof of the lower bound covers almost all of this section and as we can see in the next paragraph the main theorem follows easily from it.

**Proof of  $d_{TV}(\mathbb{P}_S, \mathbb{P}_U) = o(1)$ .** Once the above lower bound is given to us it is not hard to show that  $\mathbb{P}_S$  and  $\mathbb{P}_G$  are close to each other in total variation metric. This part is a straightforward algebraic calculation. We just need to apply the triangle inequality to the definition of  $d_{TV}(\mathbb{P}_S, \mathbb{P}_U)$  and obtain  $d_{TV}(\mathbb{P}_S, \mathbb{P}_U) \leq \sum_{G \in \mathbb{G}_{n,m,k}} |\mathbb{P}_S(G) - \mathbb{P}_U(G)|$ . Then depending on whether  $\mathbb{P}_S(G) \geq \mathbb{P}_U(G)$  or  $\mathbb{P}_S(G) < (1 - o(1))\mathbb{P}_U(G)$  we bound the term  $|\mathbb{P}_S(G) - \mathbb{P}_U(G)|$  differently. Let  $D \subset \mathbb{G}_{n,m,k}$  be the set of graphs  $G$  where the  $\mathbb{P}_S(G) < (1 - o(1))\mathbb{P}_U(G)$  and let  $B \subset \mathbb{G}_{n,m,k}$  be the set of all graphs  $G$  with  $\mathbb{P}_S(G) \leq \mathbb{P}_U(G)$ . Now assuming the lower bound on  $\mathbb{P}_S$  given above the following facts hold: (i)  $|D| = o(|\mathbb{G}_{n,m,k}|)$ . (ii) For  $G \in B \setminus D$ :  $|\mathbb{P}_S(G) - \mathbb{P}_U(G)| = \mathbb{P}_S(G) - \mathbb{P}_U(G) \leq o(1)\mathbb{P}_U(G)$ .

$$\begin{aligned}
 (6.3) \quad & \sum_{G \in \mathbb{G}_{n,m,k}} |\mathbb{P}_S(G) - \mathbb{P}_U(G)| = \\
 & \sum_{G \in \mathbb{G}_{n,m,k}} (\mathbb{P}_S(G) - \mathbb{P}_U(G)) + 2 \sum_{G \in B} |\mathbb{P}_S(G) - \mathbb{P}_U(G)| = \\
 & \sum_{G \in \mathbb{G}_{n,m,k}} (\mathbb{P}_S(G) - \mathbb{P}_U(G)) + 2 \sum_{G \in B \setminus D} |\mathbb{P}_S(G) - \mathbb{P}_U(G)| \\
 & \quad + 2 \sum_{G \in B \cap D} |\mathbb{P}_S(G) - \mathbb{P}_U(G)| \\
 & \leq \sum_{G \in \mathbb{G}_{n,m,k}} \mathbb{P}_S(G) - \sum_{G \in \mathbb{G}_{n,m,k}} \mathbb{P}_U(G) \\
 & \quad + o(1) \sum_{G \in B} \mathbb{P}_U(G) + 2 \sum_{G \in D} 1 \\
 & \leq 1 - \mathbb{P}_S(\text{FAIL}) - 1 + o(1) + o(1) \\
 & \leq o(1) - \mathbb{P}_S(\text{FAIL}) \leq o(1).
 \end{aligned}$$

Moreover, since the left hand side of the equation (6.3) is non-negative then the probability of failure of the algorithm S, denoted by  $\mathbb{P}_S(\text{FAIL})$ , is  $o(1)$ . This finishes proof of Theorem 3.1. ■

**6.1 Lower Bound For  $\mathbb{P}_S(G)$**  This proof contains five main steps and the most important ones are the Steps 4 and 5. Before entering the details we give a high level description of the analysis. Since the probabilities of selecting the edges (denoted by  $p(ij|G_t)$ ) are calculated by a Poisson approximation, they are of the form  $\frac{e^{-E_t(ij)}}{\sum_{rs} e^{-E_t(rs)}}$ . Where each random variable  $E_t(ij)$  or  $E_t(rs)$  refers to the expected number of elements in a family of cycles (see Step 2 for the details). We partition the union of these families of cycles for all steps of the algorithm into two subsets depending

on whether they are from a numerator term  $E_t(ij)$  or a denominator term  $E_t(rs)$ . We also consider a third set of cycles corresponding to non-suitable pairs (referred to by forbidden pairs in Step 3 below). The probability  $\mathbb{P}_S(G)$  can be shown to be proportional to  $e^{S_1 + S_2 + S_3}$  where each of  $S_1$ ,  $S_2$  and  $S_3$  is a function of the cycles in a unique family among the three families discussed above. We will show a combinatorial 1 to 1 correspondence between the cycles appearing in  $S_1$ ,  $S_2$ , and the ones appearing in  $S_3$  with negative signs. This simplifies the summation  $S_1 + S_2 + S_3$  and produces the desired lower bound independent of the graph  $G$ .

**Step 1: Using Jensen's Inequality.** The first step is to write an expression for the probability that the algorithm S does not fail and returns a *fixed* graph  $G \in \mathbb{G}_{n,m,k}$ . Note that algorithm S sequentially adds edges to an empty graph to produce a graph with  $m$  edges. Hence for the fixed graph  $G$  there are  $m!$  permutations of the edges of  $G$  that can be generated by algorithm S and each permutation can have a different probability. Let  $\pi$  be any permutation of the edges of  $G$  (i.e. a one-to-one mapping from  $\{1, \dots, m\}$  to the edges of  $G$ ), and let  $G_t^\pi$  be the graph having  $V$  as vertex set and  $\{\pi(1), \dots, \pi(t)\}$  as edge set. This is the partial graph that is generated after  $t$  steps of the algorithm S conditioned on having  $\pi$  as output. Now we can write  $\mathbb{P}_S(G) = \sum_{\pi} \prod_{t=0}^{m-1} p(\pi(t+1)|G_t^\pi)$ . Assuming that  $\pi$  is a uniformly random permutation then the term  $\sum_{\pi}$  can be replaced by  $m! \mathbb{E}_{\pi}$  where  $\mathbb{E}_{\pi}$  is expectation with respect to the uniformly random permutations. Thus we have

$$\begin{aligned}
 (6.4) \quad & \mathbb{P}_S(G) = m! \mathbb{E}_{\pi} \left\{ \prod_{t=0}^{m-1} p(\pi(t+1)|G_t^\pi) \right\} \\
 & = m! \mathbb{E}_{\pi} \exp \left\{ \sum_{t=0}^{m-1} \log p(\pi(t+1)|G_t^\pi) \right\} \\
 & \geq m! \exp \left\{ \sum_{t=0}^{m-1} \mathbb{E}_{\pi} \log p(\pi(t+1)|G_t^\pi) \right\}
 \end{aligned}$$

where the inequality is by Jensen's inequality for the convex function  $e^x$ .

**Step 2: Estimate for the probabilities  $p(\pi(t+1)|G_t^\pi)$ .** Recall from Section 4, that in algorithm S, after  $t$  steps, the probability of adding an edge  $(ij)$  to a partially constructed graph  $G_t$  should be proportional to the number of uniformly random extensions of  $G_t \cup (ij)$  to a graph in  $\mathbb{G}_{n,m,k}$ . Using the Poisson approximation this number is roughly proportional to  $e^{-E_k(G_t, ij)}$  where  $E_k(G_t, ij)$  is defined to be the expected number

of simple cycles<sup>1</sup> of length at most  $k$  if we add  $m - t - 1$  random edges to  $G_t \cup (ij)$ . Define  $N = \binom{n}{2}$ . It is easy to see that  $E_k(G_t, ij) = \sum_{r=3}^k \sum_{\ell=0}^{r-2} N_{r,\ell}^{G_t,(ij)} q_t^{r-1-\ell}$  where  $q_t = \frac{m-t}{N-t}$  and  $N_{r,\ell}^{G_t,(ij)}$  is the number of simple cycles in the complete graph on  $V$  that have length  $r$ , and include  $(ij)$  and exactly  $\ell$  other edges of  $G_t$ .

In fact the rigorous statement is given by the following lemma and its proof is given in [4].

**LEMMA 6.1.** *For any non-zero probability term  $p(ij|G_t)$  we have  $p(ij|G_t) \geq \frac{1}{Z(G_t)} e^{-E_k(G_t,ij) - O(n^{k(k+3)\alpha-2})}$  where  $Z(G_t) = \sum_{rs} e^{-E_k(G_t,rs)}$  is a summation over all suitable pairs  $(rs)$  at step  $t$ .*

**Note 1.** In the definition of  $p(ij|G_t)$  in Section 3 we used matrix multiplication to count number of cycles which counts non-simple cycles as well. This is because it makes the implementation of the algorithm easier. For the analysis it is convenient to work with simple cycles and the above lemma provides the comparison that we need between these two methods of counting cycles.

**Step 3: Algebraic modifications.** Now we use the estimates given by Lemma 6.1 in equation (6.4) to obtain:

$$\mathbb{P}_S(G) \geq m! \exp \left[ - \sum_{t=0}^{m-1} \mathbb{E}_\pi E_k(G_t^\pi, \pi(t+1)) - \sum_{t=0}^{m-1} \mathbb{E}_\pi \log Z(G_t^\pi) - O(n^{(k(k+3)+1)\alpha-1}) \right].$$

By condition  $\alpha < [(k(k+3)+1)]^{-1}$  the third term of the exponent is  $o(1)$ . Now we are going to simplify the second term of the exponent. Let us define  $Z_0(G_t^\pi) = N - t - \sum_{r=3}^k F_r(G_t^\pi)$  where  $F_r(G_t^\pi)$  is number of all forbidden (unsuitable) pairs at step  $t$ . Now we drop the reference to  $G_t^\pi$  for simplicity and obtain:

$$\begin{aligned} \log Z &= \log Z_0 + \log(Z/Z_0) \\ &= \log \left[ (N-t) \left( 1 - \frac{\sum_{r=3}^k F_r}{N-t} \right) \right] + \log(Z/Z_0) \\ &\stackrel{(a)}{\leq} \log(N-t) - \frac{\sum_{r=3}^k F_r}{N-t} + \log(Z/Z_0) \\ &\stackrel{(b)}{\leq} \log(N-t) - \frac{\sum_{r=3}^k F_r}{N} + \log(Z/Z_0) \end{aligned}$$

where (a) uses  $\log(1-x) \leq -x$  for  $x \in [0, 1]$  and (b) uses  $N-t \leq N$ . Using the above we will the following

<sup>1</sup>Cycles that do not repeat a vertex or edge.

modified lower bound for  $\mathbb{P}_S(G)$ :

$$(6.5) \quad \mathbb{P}_S(G) \geq \frac{1}{\binom{N}{m}} \exp \left[ - \sum_{t=0}^{m-1} \mathbb{E}_\pi E_k(G_t^\pi, \pi(t+1)) + \frac{1}{N} \sum_{r=3}^k \sum_{t=0}^{m-1} \mathbb{E}_\pi F_r(G_t^\pi) - \sum_{t=0}^{m-1} \mathbb{E}_\pi \log \frac{Z(G_t^\pi)}{Z_0(G_t^\pi)} + o(1) \right].$$

To simplify the notation, we will denote the three terms in the exponent by  $S_1(G), S_2(G), S_3(G)$  respectively. In particular:  $S_1(G) = - \sum_{t=0}^{m-1} \mathbb{E}_\pi E_k(G_t^\pi, \pi(t+1))$ ,  $S_2(G) = \frac{1}{N} \sum_{r=3}^k \sum_{t=0}^{m-1} \mathbb{E}_\pi F_r(G_t^\pi)$  and  $S_3(G) = - \sum_{t=0}^{m-1} \mathbb{E}_\pi \log \frac{Z(G_t^\pi)}{Z_0(G_t^\pi)}$ .

Next two steps are the most important parts of our effort in proving the inequality  $\mathbb{P}_S(G) \geq (1 - o(1)) \mathbb{P}_U(G)$ .

**Step 4: Simplifying the exponent  $S_1(G) + S_2(G) + S_3(G)$ .** This step shows the main benefit of leaving the calculation of approximation errors for  $p(ij|G_t^\pi)$  to final steps. We will show that even though the terms  $S_i(G)$  for  $i = 1, 2, 3$  can be large and dependent on graph  $G$ , their sum can be simplified to an expression which is independent of graph  $G$ . In fact we will show that  $S_1$  and  $S_2$  will be completely canceled by the graph dependent parts of  $S_3$ .

First we are going to find a lower bound for the  $S_i$ 's. These lower bound are given with the following lemma.

**LEMMA 6.2.** *Let  $m = n^{1+\alpha}$  and constant  $k \geq 3$  be such that  $\alpha \leq [2k(k+3)]^{-1}$  then for all but  $o(|\mathbb{G}_{n,m,k}|)$  number of graphs  $G \in \mathbb{G}_{n,m,k}$  the followings hold:*

- (a)  $S_1(G) \geq o(1) - \sum_{r=3}^k \sum_{\ell=1}^{r-1} |\mathbb{C}_{r,\ell}(G)| \left(\frac{m}{N}\right)^{r-\ell} \ell \int_0^1 \theta^{\ell-1} (1-\theta)^{r-\ell} d\theta.$
- (b)  $S_2(G) \geq o(1) + \sum_{r=3}^k |\mathbb{C}_{r,r-1}(G)| \frac{m}{N} \int_0^1 \theta^{r-1} d\theta.$
- (c)  $S_3(G) \geq o(1) + \sum_{r=3}^k \sum_{\ell=0}^{r-2} |\mathbb{C}_{r,\ell}(G)| \left(\frac{m}{N}\right)^{r-\ell} (r-\ell) \int_0^1 \theta^\ell (1-\theta)^{r-\ell-1} d\theta.$

where  $\mathbb{C}_{r,\ell}(G)$  is set of all simple cycles of length  $r$  in complete graph on  $V$  that include exactly  $\ell$  edges of  $G$ .

Let us start by providing a high level over view of the proof of this lemma and for now we just focus on  $S_1(G)$ . By definition  $S_1(G) = - \sum_{t=0}^{m-1} E_k(G_t^\pi, \pi(t+1))$ . The first approximation we use is to change the randomness given by  $\pi$ . The partial graph  $G_t^\pi$  is a uniformly random subgraph of  $G$  that has exactly  $t$  edges. Instead we look at  $G_\theta$  which is a random subgraph of  $G$  that has each edge of  $G$  independently with probability  $\theta = t/m$ . The subgraph  $G_\theta$  has  $t$  edges in expectation which makes it a good approximation for  $G_t^\pi$ . With this substitution, one can see that  $- \sum_{t=0}^{m-1} \mathbb{E}_\pi E_k(G_t^\pi, \pi(t+1))$

1)) is roughly equal to  $-m\mathbb{E}_\theta \int_0^1 d\theta E_k(G_\theta, (ij))$  where  $(ij)$  is a uniformly random edge of  $G$ .

The next steps focus on expanding the  $E_k$ .  $E_k$  is summation of the terms  $q_t^{r-\ell-1}$  for any couples  $(\gamma, ij)$  where  $\gamma$  is a simple cycle of length  $r$  on complete graph on  $V$  that has  $\ell$  edges of  $G_\theta$  and  $(ij)$  is an edge of  $G \setminus G_\theta$  that is in  $\gamma$  as well. For any fixed  $r, \ell$  it can be shown that the expected number of such couples is dominated by the cases where  $|\gamma \cap G_\theta| = |\gamma \cap G| - 1 = \ell$  that is when  $(ij)$  is the only edge of  $G \cap \gamma$  that is not in  $G_\theta$ . Moreover  $q_t$  is approximately equal to  $(1 - \theta)m/N$ . Therefore the contribution of the pairs  $(\gamma, ij)$  in  $S_1(G)$  for any fixed  $r, \ell$  (i.e. for all  $\gamma \in \mathbb{C}_{r,\ell}$ ) is the same. Thus we obtain the right hand side of the Lemma 6.2(a).

The above discussion is proven rigorously in [4]. Now we will show how  $S_1 + S_2 + S_3$  is simplified using the above lower bounds.

**Step 5: Finishing the proof of  $\mathbb{P}_S \geq (1 - o(1))\mathbb{P}_U$ .** First we provide an estimate for number of graphs in the set  $\mathbb{G}_{n,m,k}$ . Using Janson's inequality [2] one can obtain close approximation for number of graphs in the set  $\mathbb{G}_{n,m,k}$ . In fact Janson's inequality shows the number of cycle of constant length in  $\mathbb{G}_{n,m}$  converges to a Poisson random variable. The following Lemma formalizes this statement.

**LEMMA 6.3.** *Let  $m = n^{1+\alpha}$  and constant  $k \geq 3$  be such that  $\alpha < (2k - 1)^{-1}$ . Then  $|\mathbb{G}_{n,m,k}|$  is equal to  $\binom{N}{m} \exp \left[ -\sum_{r=3}^k |\mathbb{C}_r| \left(\frac{m}{N}\right)^r + o(1) \right]$ .*

Where  $\mathbb{C}_r$  is the set of all simple cycles of length  $r$  in the complete graph on vertices of  $V$ . Proof of Lemma 6.3 is given in Appendix A.

Next we will show that how different terms in the lower bounds for  $S_i$ 's in Lemma 6.2 cancel each other. The main key in relating the terms in those lower bounds of  $S_i$ 's is the following equation which is obtained using integration by parts for  $r - 1 \geq \ell > 1$ :

$$(6.6) \quad \ell \int_0^1 \theta^{\ell-1} (1 - \theta)^{r-\ell} d\theta = (r - \ell) \int_0^1 \theta^\ell (1 - \theta)^{r-\ell-1} d\theta$$

Therefore, combining equation (6.5) and Lemma 6.2 for all graphs  $G$  in  $\mathbb{G}_{n,m,k}$  that satisfy Lemma 6.2 (all but  $o(1)$  fraction of the elements in  $\mathbb{G}_{n,m,k}$ ) we obtain:

$$(6.7) \quad \begin{aligned} \mathbb{P}_S(G) &\geq \frac{e^{S_1(G)+S_2(G)+S_3(G)+o(1)}}{\binom{N}{m}} \\ &\geq \frac{e^{o(1)+\sum_{r=3}^k |\mathbb{C}_{r,0}(G)| \left(\frac{m}{N}\right)^r} \int_0^1 (1-\theta)^{r-1} d\theta}{\binom{N}{m}} \\ &= \frac{1}{\binom{N}{m} e^{o(1)-\sum_{r=3}^k |\mathbb{C}_{r,0}(G)| \left(\frac{m}{N}\right)^r}} \end{aligned}$$

where the second inequality is the result of equation (6.6) and some algebraic cancelations as follows. All terms in the lower bound for  $S_1$  with  $1 \leq \ell \leq r - 2$  are canceled with the corresponding terms in the lower bound for  $S_3$ , and the  $\ell = r - 1$  term in the lower bound of  $S_1$  is canceled with the lower bound of  $S_2$ . Therefore the uncanceled terms in  $S_1(G) + S_2(G) + S_3(G)$  are  $\ell = 0$  terms from the lower bound of  $S_3$ .

Comparing the right hand side of the equation (6.7) and the expression for  $|\mathbb{G}_{n,m,k}|$  given by Lemma 6.3 we see that the only difference is the use of  $\mathbb{C}_{r,0}(G)$  instead of  $\mathbb{C}_r$ . But note that for a cycle  $C \in \mathbb{C}_r$  the probability that it intersects graphs  $G$  is at most  $r$  times the probability that a fixed edge of  $C$  intersects  $G$  (union bound). Therefore  $\mathbb{P}_{n,m}(C \cap G \neq \emptyset) \leq rm/N = O(n^{\alpha-1})$ . This shows the following:

$$\begin{aligned} \sum_{r=3}^k |\mathbb{C}_{r,0}(G)| \left(\frac{m}{N}\right)^r &\geq \sum_{r=3}^k |\mathbb{C}_r| (1 - O(n^{\alpha-1})) \left(\frac{m}{N}\right)^r \\ &\geq -O(n^{(k+1)\alpha-1}) + \sum_{r=3}^k |\mathbb{C}_r| \left(\frac{m}{N}\right)^r \\ &= o(1) + \sum_{r=3}^k |\mathbb{C}_r| \left(\frac{m}{N}\right)^r. \end{aligned}$$

Thus, we obtain

$$\mathbb{P}_S(G) \geq \frac{1}{\binom{N}{m} e^{o(1)-\sum_{r=3}^k |\mathbb{C}_r| \left(\frac{m}{N}\right)^r}} \stackrel{(h)}{\geq} (1 - o(1)) \mathbb{P}_U(G)$$

where (h) uses Lemma 6.3. ■

**6.2 Running time of the algorithm** The fact that Algorithms S has polynomial running time is clear since the matrix of the probabilities,  $P_{G_t}$ , at any step can be calculated using matrix multiplication. In fact a naïve calculation shows that  $P_{G_t}$  can be calculated with  $O(kn^3) = O(n^3)$  operations. This is because  $A^r$  for any  $r$  takes  $O(rn^3)$  operations to compute. So we obtain the simple bound of  $O(n^3m)$  for the running time of the algorithm S. But one can improve this running time by at least a factor  $n$  with exploiting the structure of the matrices.

Recall that the matrix  $P_{G_t}$  is proportional to  $S_t \odot \widehat{\exp} \left[ -\sum_{a=2}^{k-1} \left( M_t + \frac{m-t}{\binom{n}{2}-t} M_t^c \right)^a \right]$ . Let us denote the matrix  $M_t + \frac{m-t}{\binom{n}{2}-t} M_t^c$  by  $X_t$ . We also denote the adjacency matrix of the partially constructed graph  $G_t$  at step  $t$  by  $A_t$ . Notice that the adjacency matrix of all "suitable pairs" at step  $t$  (denoted by  $S_t$ ) is equal to  $J_n - \widehat{\text{sign}}(\sum_{r=0}^{k-1} A_t^r)$  where  $J_n$  is the  $n$  by  $n$  matrix of all ones and the operation  $\widehat{\text{sign}}(B)$  for any matrix  $B$  means that the "sign" function is applied to each entry of the

matrix  $B$ . This is correct since any non-suitable pair  $(ij)$  corresponds to a path in  $G_t$  of length  $r$  between  $i$  and  $j$  for  $0 \leq r \leq k-1$ . Such path forces the  $ij$  entry of the matrix  $A_t^k$  to be positive.

Now we can store the matrices  $A_t, \dots, A_t^{k-1}$  and  $X_t^2, \dots, X_t^{k-1}$  at the end of each iteration and use them to efficiently calculate  $A_{t+1}, \dots, A_{t+1}^{k-1}$  and  $X_{t+1}^2, \dots, X_{t+1}^{k-1}$ . This is because the differences  $A_{t+1} - A_t$  and  $X_{t+1} - X_t$  are very sparse matrices and hence updating the matrix multiplications can be done with  $O(n^2)$  operations which reduces the overall running time to  $O(n^2m)$ .

Since Theorem 3.1 shows that algorithm S is successful with probability  $1 - o(1)$  then the expected running time of the algorithm S for generating an element of the set  $\mathbb{G}_{n,m,k}$  is  $O(n^2m)$  as well. Similar idea can be extended to the algorithm Bip-S for generating LDPC codes. We leave a detailed analysis of the running time and implementation of these algorithms to the longer version of the paper.

## 7 Acknowledgement

The authors would like to thank Balaji Prabhakar, Joel Spencer and Daniel Spielman for useful discussions. While working on this paper Mohsen Bayati was supported by Microsoft Technical Computing Initiative.

## References

- [1] D. Alderson, J. Doyle, and W. Willinger, "Toward and Optimization-Driven Framework for Designing and Generating Realistic Internet Topologies", *HotNets*, 2002.
- [2] N. Alon and J. Spencer, "The Probabilistic Method", *Wiley, NY*, 1992.
- [3] A. Amraoui, A. Montanari and R. Urbanke, "How to Find Good Finite-Length Codes: From Art Towards Science", *Eur. Trans. Telecomm.* 18 (2007), 491-508
- [4] M. Bayati, A. Montanari and A. Saberi, *Generating Random Graphs with Large Girth*, (2008) arXiv.
- [5] M. Bayati, J. H. Kim and A. Saberi, "A Sequential Algorithm for Generating Random Graphs", *International Workshop on Randomization and Computation*, 2007.
- [6] J. Blanchet, "Efficient Importance Sampling for Binary Contingency Tables", preprint, 2007.
- [7] J. Blitzstein and P. Diaconis, "A sequential importance sampling algorithm for generating random graphs with prescribed degrees", preprint, 2006.
- [8] B. Bollobás, and O. Riordan, "Constrained graph processes", *Electronic Journal of Combinatorics*, 7, 2000.
- [9] B. Bollobás, *Random Graphs*, Cambridge University Press, 2001.
- [10] T. Bu and D. Towsley, "On Distinguishing between Internet Power Law Topology Generator", *INFOCOM*, 2002.
- [11] Y. Chen, P. Diaconis, S. Holmes and J.S. Liu, "Sequential Monte Carlo methods for statistical analysis of tables", *Journal of the American Statistical Association* 100, 109-120, 2005.
- [12] S. Y. Chung, G. D. Forney, Jr., T. Richardson and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Comm. Lett.* 5 (2001), 58-60
- [13] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On Power-law Relationships of the Internet Topology", *SIGCOM*, 1999.
- [14] X. Hu, E. Eleftheriou and D. Arnold, "Progressive Edge-Growth Tanner Graphs", Proc. of GLOBECOM 01, San Antonio, USA, Nov 2001
- [15] X. Hu, E. Eleftheriou and D. Arnold, "Regular and irregular progressive edge-growth Tanner graphs", *IEEE Trans. on Inform. Theory*, 51 (2005), 386-398
- [16] P. Erdős, S. Suen, and P. Winkler, "On the size of a random maximal graph", *Random Structure and Algorithms*, 6, pp 309-318, 1995.
- [17] S. Janson, T. Luczak, and A. Rucinski, "Random Graphs", *Wiley-Interscience*, 2000.
- [18] J. H. Kim, V. H. Vu, "Generating random regular graphs", *ACM Symposium on Theory of Computing (STOC)*, 213-222, 2003.
- [19] D. Knuth, "Mathematics and computer science: coping with finiteness", *Science* 194(4271):1235-1242, 1976.
- [20] R. Koetter and P. Vontobel, "Graph covers and iterative decoding of finite-length codes," *Int. Conf. on Turbo codes and Rel. Topics*, Brest, France, September 2003.
- [21] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman and V. Stemann, "Practical Loss-Resilient Codes," 29th annual ACM Symposium on Theory of Computing (STOC), 1997, pp. 150-159
- [22] D. J. MacKay and M. S. Postol, "Weaknesses of Margulis and Ramanujan-Margulis Low-Density Parity-Check Codes", *Elec. Notes in Theor. Comp. Science* 74 (2003)
- [23] A. Medina, I. Matta and J. Byers, "On the origin of power laws in Internet topologies", *ACM Computer Communication Review*, vol. 30, no. 2, pp. 18-28, 2000.
- [24] R. Milo, S. ShenOrr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon, "Network motifs: Simple building blocks of complex networks", *Science* 298, 824-827, 2002.
- [25] A. Montanari, "The asymptotic error floor of LDPC ensembles under BP decoding," 44rd Allerton Conference on Comm. Control and Comp., Monticello, IL, Sep 2006
- [26] J. M. F. Moura and J. Lu and H. Zhang "Structured low-density parity-check codes", *IEEE Sign. Proc. Mag.*, 21 (2004), 42-55
- [27] D. Osthus, and A. Taraz, "Random maximal H-free graphs", *Random Structures and Algorithms*, 1999.

- [28] C. Di, D. Proietti, I. E. Teletar, T. J. Richardson and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel", *IEEE Trans. Inform. Theory*, 46 (2002) 1570-1579
- [29] A. Ramamoorthy and R. Wesel, "Construction of short block length irregular low-density parity-check codes" *IEEE Intern. Conf. on Comm.*, Paris, June 2004
- [30] T. Richardson, "Error floors of LDPC codes", 41st Allerton Conference on Comm. Control and Comp., Monticello, IL, Sep 2003
- [31] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, Cambridge 2008
- [32] J. Rosenthal and P. O. Vontobel, "Constructions of regular and irregular LDPC codes using Ramanujan graphs and ideas from Margulis", *IEEE Symp. on Inform. Theory*, Washington, June 2001
- [33] A. Rucinski, and N. Wormald, "Random graph processes with degree restrictions", *Combin. Prob. Comput.* 1, pp 169-180, 1992.
- [34] A. Sinclair, "Algorithms for random generation and counting : a Markov chain approach", Birkhäuser, 1993.
- [35] A. Steger and N.C. Wormald, "Generating random regular graphs quickly", *Combinatorics, Probab. and Comput* (8), pp 377-396, 1999.
- [36] A. Steger, "On the Evolution of Triangle-Free Graphs", *Combinatorics, Probability and Computing*, 14, pp 211-224, 2005.
- [37] J. Spencer, "Counting extensions", *J. Combin. Theory Ser. A* 53, pp247-255, 1990.
- [38] J. Spencer, "Maximal triangle-free graphs and Ramsey  $R(3,t)$ ", unpublished manuscript, 1995.
- [39] M. G. Stepanov, V. Y. Chernyak and M. Chertkov, "Diagnosis of weaknesses in modern error correction codes: a physics approach", *Phys Rev. Lett.* 95 (2005) 228701-228704
- [40] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network Topology Generators: Degree based vs. Structural", *ACM SIGCOM*, 2002.
- [41] N. C. Wormald, "Models of random regular graphs", *Surveys in combinatorics (Canterbury)* London Math. Soc. Lecture Notes Ser., Vol 265. Cambridge Univ. Press, Cambridge, 239-298, 1999.
- [42] H. Xiao and A. H. Banihashemi, "Improved progressive-edge-growth (PEG) construction of LDPC codes", *Proc. of GLOBECOM 04*, Dallas, USA, Nov 2004

### A Estimate For $|\mathbb{G}_{n,m,k}|$ and proof of Lemma 6.3

First we approximate the random graph model  $\mathbb{G}_{n,m}$  with the Erdős-Reyni model  $\mathbb{G}_{n,p}$  for  $p = m/N$ . We will denote the probability with respect to the randomness in  $\mathbb{G}_{n,p}$  and  $\mathbb{G}_{n,m}$  by  $\mathbb{P}_{n,p}$  and  $\mathbb{P}_{n,m}$  respectively. Let  $A_k$  be the event that a random graph, selected from  $\mathbb{G}(n,p)$  or  $\mathbb{G}(m,n)$ , has girth greater than  $k$ . First we will calculate the probability  $\mathbb{P}_{n,p}(A_k)$ . Then we

will formalize the following approximation to obtain estimates for  $|\mathbb{G}_{n,m,k}|$ :  $\mathbb{P}_{n,p}(A_k) \approx |\mathbb{G}_{n,m,k}|/|\mathbb{G}_{n,m}|$ .

**A.1 Janson's Inequality** In this section we briefly review Janson inequality. Given a collection of 'bad events'  $\{B_i : i \in \mathbb{I}\}$ , we would like to estimate the probability  $\mathbb{P}(\cap_{i \in \mathbb{I}} B_i^c)$  assuming that the events  $B_i^c, i \in \mathbb{I}$  are "almost independent". More formally, let  $\beta, \gamma$  be real numbers such that  $\beta < 1$  and for all  $i \in \mathbb{I} : \mathbb{P}(B_i) \leq \beta$  and  $\sum_{B_j \sim B_i} \mathbb{P}(B_i \cap B_j) = \gamma$  where  $B_i \sim B_j$  means that the events  $B_i, B_j$  are dependent. Then Janson's inequality is the following:

$$(1.8) \quad \prod_{i \in \mathbb{I}} \mathbb{P}(B_i^c) \leq \mathbb{P}(\cap_{i \in \mathbb{I}} B_i^c) \leq \prod_{i \in \mathbb{I}} \mathbb{P}(B_i^c) \exp\left(\frac{\gamma}{2(1-\beta)}\right).$$

In particular, for  $\gamma = o(1)$  we obtain:  $\mathbb{P}(\cap_{i \in \mathbb{I}} B_i^c) = (1 + o(1)) \prod_{i \in \mathbb{I}} \mathbb{P}(B_i^c)$ .

**A.2 Finding  $\mathbb{P}_{n,p}(A_k)$  via Janson Inequality** By definition,  $\mathbb{P}_{n,p}(A_k)$  is the probability that a random graph  $G$  in  $\mathbb{G}_{n,p}$  has no cycle of length at most  $k$ . Define the set of all cycles of length  $r$  in complete graph on  $V$  by  $\mathbb{C}_r$ . Let  $\mathbb{C} = \cup_{r=3}^k \mathbb{C}_r$  and consider the set of bad events  $B_i, i \in \mathbb{C}$  where  $B_i$  is the event that  $G$  contains the cycle  $i$ . It is not difficult to see that:  $\mathbb{P}(B_i) = O(p^3)$  and  $\gamma = O\left(\sum_{r=3}^k n^{2r-2} p^{2r-1}\right)$ . And since  $p = O(n^{\alpha-1})$  then using Janson's inequality (1.8) we obtain:

$$\prod_{i \in \mathbb{C}} \mathbb{P}(B_i^c) \leq \mathbb{P}_{n,p}(A_k) \leq e^{O(n^{(2k-1)\alpha-1})} \prod_{i \in \mathbb{I}} \mathbb{P}(B_i^c)$$

which gives the following for  $\alpha < 1/(2k-1)$ :

$$(1.9) \quad \begin{aligned} \mathbb{P}_{n,p}(A_k) &= e^{o(1)} \prod_{i \in \mathbb{C}} \mathbb{P}(B_i^c) \\ &= e^{o(1)} \prod_{i \in \mathbb{C}} (1 - p^{\text{length}(i)}) \\ &= \exp\left[o(1) + \sum_{r=3}^k |\mathbb{C}_r| \log(1 - p^r)\right] \\ &= \exp\left[o(1) - \sum_{r=3}^k |\mathbb{C}_r| p^r\right] \end{aligned}$$

where the last equality uses  $|\mathbb{C}_r| p^{2r} = O(n^r n^{2r\alpha-2r}) = o(1)$  since  $\alpha < 1/2$ .

Finally we are ready to prove Lemma 6.3 which is achieved by approximating  $|\mathbb{G}_{n,m,k}|$  through  $\mathbb{P}_{n,p}(A_k)$ .

**Proof of Lemma 6.3.** Proof uses monotonicity of the events  $A_k$  in the random models  $\mathbb{G}_{n,p}$  and  $\mathbb{G}_{n,m}$  (see Lemma 1.10 in [17]). In particular for  $0 \leq p_1 \leq p_2 \leq 1$  and  $0 \leq M_1 \leq M_2 \leq N$  the following inequalities hold:

$$\mathbb{P}_{n,p_1}(A_k) \geq \mathbb{P}_{n,p_2}(A_k) \text{ and } \mathbb{P}_{n,M_1}(A_k) \geq \mathbb{P}_{n,M_2}(A_k).$$

On the other hand for any  $0 < q < 1$  the random graph model  $\mathbb{G}(n, q)$  conditioning on graphs to have exactly  $m$  edges is equivalent to the random graph model  $\mathbb{G}_{n,m}$ . Thus for a random graph  $G$  the following holds:

$$\begin{aligned}
 (1.10) \quad \mathbb{P}_{n,q}(A_k) &= \mathbb{P}_{n,q}(|E(G)| \geq m) + \mathbb{P}_{n,q}(|E(G)| < m) \\
 &\leq \sum_{\ell=m}^N \mathbb{P}_{n,q}(A_k | |E(G)| = \ell) \mathbb{P}_{n,q}(|E(G)| = \ell) \\
 &\quad + \mathbb{P}_{n,q}(|E(G)| < m) \\
 &\stackrel{(a)}{\leq} \sum_{\ell=m}^N \mathbb{P}_{n,q}(A_k | |E(G)| = m) \mathbb{P}_{n,q}(|E(G)| = \ell) \\
 &\quad + \mathbb{P}_{n,q}(|E(G)| < m) \\
 &\leq \mathbb{P}_{n,m}(A_k) + \mathbb{P}_{n,q}(|E(G)| < m)
 \end{aligned}$$

Similarly, one can obtain the following:

$$\begin{aligned}
 (1.11) \quad \mathbb{P}_{n,q}(A_k) &\geq \mathbb{P}_{n,q}(A_k \cap \{|E(G)| \leq m\}) \\
 &= \sum_{\ell=0}^m \mathbb{P}_{n,q}(A_k | |E(G)| = \ell) \mathbb{P}_{n,q}(|E(G)| = \ell) \\
 &\stackrel{(b)}{\geq} \mathbb{P}_{n,q}(A_k | |E(G)| = m) \sum_{\ell=0}^m \mathbb{P}_{n,q}(|E(G)| = \ell) \\
 &= \mathbb{P}_{n,m}(A_k) \mathbb{P}_{n,q}(|E(G)| \leq m) \\
 &\geq \mathbb{P}_{n,m}(A_k) - \mathbb{P}_{n,q}(|E(G)| > m)
 \end{aligned}$$

where both (a) and (b) use monotonicity. Now for a small constant  $\beta > 0$  let

$$p_1 = \frac{m - m^{\frac{1+\beta}{2}}}{N} \quad \text{and} \quad p_2 = \frac{m + m^{\frac{1+\beta}{2}}}{N}.$$

Using Hoeffding's inequality the followings hold

$$\begin{aligned}
 (1.12) \quad \mathbb{P}_{n,p_1}(|E(G)| > m) &\leq e^{-\frac{m\beta}{8}} \\
 \mathbb{P}_{n,p_2}(|E(G)| < m) &\leq e^{-\frac{m\beta}{8}}
 \end{aligned}$$

This is due to a variation of the Hoeffding's given in [35] that for any  $0 < q < 1$  and  $0 < \delta < \frac{4}{5}$  gives:

$$\mathbb{P}_{n,q} \left( \left| |E(G)| - Nq \right| > \delta Nq \right) \leq e^{-\frac{\delta^2 Nq}{4}}$$

Now one can see that by taking  $\delta = \frac{m^{1+\beta}}{2(m - m^{1+\beta})}$  we obtain:

$$\begin{aligned}
 \mathbb{P}_{n,p_1} \left( |E(G)| > m \right) &\stackrel{(c)}{\leq} \mathbb{P}_{n,p_1} \left( |E(G)| > (1 + \delta)Nq \right) \\
 &\leq e^{-\frac{\delta^2 Nq}{4}} \leq e^{-\frac{m\beta}{8}}
 \end{aligned}$$

where (c) uses  $m > (1 + \delta)Nq$ . Similarly we can prove the second inequality in (1.12). Thus we can use equations (1.9), (1.11) to obtain the following bound for  $\mathbb{P}_{n,m}(A_k)$ :

$$\begin{aligned}
 \mathbb{P}_{n,m}(A_k) &\leq \mathbb{P}_{n,p_2}(A_k) + \mathbb{P}_{n,p_2}(|E(G)| < m) \\
 &\leq \exp \left[ o(1) - \sum_{r=3}^k |\mathbb{C}_r| p_2^r \right] + e^{-\frac{m\beta}{8}}
 \end{aligned}$$

and similarly using (1.10) we get the lower bound:

$$\begin{aligned}
 \mathbb{P}_{n,m}(A_k) &\geq \mathbb{P}_{n,p_1}(A_k) + \mathbb{P}_{n,p_1}(|E(G)| > m) \\
 &\geq \exp \left[ o(1) - \sum_{r=3}^k |\mathbb{C}_r| p_1^r \right] - e^{-\frac{m\beta}{8}}.
 \end{aligned}$$

Let  $H(p) = \sum_{r=3}^k |\mathbb{C}_r| p^r$  then combining the above two inequalities:

$$\begin{aligned}
 (1.13) \quad e^{o(1)+H(p)-H(p_1)} - e^{H(p)-\frac{m\beta}{8}} \\
 \leq \frac{\mathbb{P}_{n,m}(A_k)}{\exp[-H(p)]} \leq \\
 e^{o(1)+H(p)-H(p_2)} + e^{H(p)-\frac{m\beta}{8}}.
 \end{aligned}$$

Now we use  $H(p) = O(n^{k\alpha})$ ,  $m = n^{1+\alpha}$  and apply the mean value theorem to  $H(x)$  to obtain  $|H(p) - H(p_i)| = |p - p_i| |H'(\bar{p})|$  for  $\bar{p}$  between  $p, p_i$ . Since for  $H'(\bar{p})$  we have  $H'(\bar{p}) = O(n^{(k-1)\alpha})$  and  $|p_i - p| = O(n^{\frac{(1+\alpha)(1+\beta)}{2} - 2})$ , one can simplify the equation (1.13) to obtain the following

$$\begin{aligned}
 (1.14) \quad \left| \exp[o(1) + O(n^{\frac{(1+\beta)(1+\alpha)}{2} + (k-1)\alpha - 2})] - \frac{\mathbb{P}_{n,m}(A_k)}{\exp[-H(p)]} \right| \\
 \leq e^{O(n^{k\alpha} - \frac{n^{(1+\alpha)\beta}}{8})}
 \end{aligned}$$

We would like to have  $n^{k\alpha} - \frac{n^{(1+\alpha)\beta}}{8} \rightarrow -\infty$  and  $\frac{(1+\beta)(1+\alpha)}{2} + (k-1)\alpha - 2 \rightarrow -\infty$  which will give us  $\mathbb{P}_{n,m}(A_k) = e^{o(1)-H(p)}$ . These inequalities can be achieved if  $\beta$  satisfies  $\frac{k\alpha}{1+\alpha} < \beta < \frac{3-(2k-1)\alpha}{1+\alpha}$  and such  $\beta$  exist since by assumption  $\alpha < 1/(2k-1)$  that guarantees the upper bound for  $\beta$  is larger than the lower bound. Therefore

$$\begin{aligned}
 |\mathbb{G}_{n,m,k}| &= \binom{N}{m} \mathbb{P}_{n,m}(A_k) \\
 &= \binom{N}{m} \exp[o(1) - H(p)] \\
 &= \binom{N}{m} \exp \left[ o(1) - \sum_{r=3}^k |\mathbb{C}_r| \left( \frac{M}{N} \right)^r \right].
 \end{aligned}$$

This finishes the proof. ■