Uniform generation of random graphs with power-law degree sequences

Pu Gao*
School of Mathematics
Monash University
jane.gao@monash.edu

Nicholas Wormald†
School of Mathematics
Monash University
nick.wormald@monash.edu

Abstract

We give a linear-time algorithm that approximately uniformly generates a random simple graph with a power-law degree sequence whose exponent is at least 2.8811. While sampling graphs with power-law degree sequence of exponent at least 3 is fairly easy, and many samplers work efficiently in this case, the problem becomes dramatically more difficult when the exponent drops below 3; ours is the first provably practicable sampler for this case. We also show that with an appropriate rejection scheme, our algorithm can be tuned into an exact uniform sampler. The running time of the exact sampler is $O(n^{2.107})$ with high probability, and $O(n^{4.081})$ in expectation.

1 Introduction

We consider the following problem. Given a sequence of nonnegative integers $\mathbf{d}$ with even sum, how can we generate uniformly at random a graph with degree sequence $\mathbf{d}$? Motivation to do this can come from testing algorithms, to test null hypotheses in statistics (see Blitzstein and Diaconis [2] for example), or to simulate networks. If the degree sequence comes from a 'real-world' network then it will often follow a power law (defined precisely below), and often the 'exponent' of this power law is between 2 and 3. We give a linear-time algorithm for generating such graphs approximately uniformly at random in the sense that the total variation distance between the output a distribution from the uniform one is $o(1)$ as the number of vertices goes to infinity. There has been no such approximate sampler before with good guaranteed performance, i.e. small error in distribution and low time complexity, for power-law degree sequences with γ below 3. We will also give an exact uniform sampler, for γ less than but sufficiently close to 3, which has a reasonably low time complexity: $n^{2.107}$ with high probability and $n^{4.081}$ in expectation.

Research on uniform generation of graphs with prescribed degrees has a long history. Tinhofer [21] studied the case of bounded regular degrees and described an algorithm without bounding how far the output is from the uniform distribution. Soon afterwards, a uniform sampler using a simple rejection scheme (described in Section 2) arose as an immediate consequence of several enumeration methods [4, 5]. This algorithm works efficiently only when the maximum degree is $O(\sqrt{\log n})$, where n is the number of vertices. A major breakthrough that significantly relaxed the constraint on the maximum degree was by McKay and Wormald [18]. Their algorithm efficiently generates random graphs whose degree sequence satisfies $\Delta = O(m^{1/4})$, where Δ denotes the maximum degree and m is the number of edges in the graph. The expected running time of this algorithm is $O(\Delta^4 n^2)$. Very recently, the authors of the current paper developed a new algorithm which successfully generates random d-regular graphs for $d = o(\sqrt{n})$, with expected running time $O(nd^3)$. This new algorithm is related to the McKay-Wormald algorithm but contains significant new features which make it possible to cope with larger degrees without essentially increasing the runtime.

While the uniform generation of graphs whose degrees are regular, or close to regular, has already been found challenging, generating graphs whose degrees are very far from being regular seems much more difficult. However, a desire to generate such graphs materialised in recent years. Because of the important roles of the internet and social networks in modern society, much attention has been paid to graphs with real-world network properties. One of the most prominent traits of many real-world networks is that their degree distribution follows the so-called power law, usually with parameter γ between 2 and 3 (i.e. the number of vertices with degree i is roughly proportional to $i^{-\gamma}$). Graphs with such degree distributions are sparse but have vertices with very large degrees. There is more than one definition of a power law degree sequence examined in the literature, and we consider the common one resulting when the degrees are independently distributed with a power

*Research supported by NSERC and ARC DP160100835.
†Research supported by ARC DP160100835.

law with parameter γ. For this, the maximum degree is roughly $n^{1/(\gamma-1)}$.

Generating random graphs with power-law degrees for $\gamma > 3$ is quite easy using the above-mentioned simple rejection scheme or the McKay-Wormald algorithm. However, a radical change occurs when the power-law parameter γ drops from above 3 to below. A notable difference is that the second moment of the degree distribution changes from finite to infinite. As a result, the running time of the simple rejection scheme changes from linear to super-polynomial time. The running time required for the McKay-Wormald algorithm also becomes super-polynomial when $\gamma < 3$. Another major difference is the appearance of vertices with degree well above $\sqrt{m}$. To date, no algorithm has been proved to uniformly generate such graphs in time that is useful in practice, with any reasonable performance guarantee on the distribution.

Some known approximation algorithms have polynomial running times without being provably practical. Jerrum and Sinclair [15] gave an approximate sampler using the Markov Chain Monte Carlo method for generating graphs with so-called P-stable degree sequences. Sufficient conditions for P-stability and non-P-stability are given by Jerrum, McKay and Sinclair [14], but none of these conditions are met by power-law degree sequences with $2 < \gamma < 3$. On the other hand, the authors found enumeration formulae for power-law degree sequences [8] that imply P-stability when $\gamma > 1 + \sqrt{3}$, and one can show using a related but simpler argument that P-stability holds for all $\gamma > 2$. However the degree of the polynomial bound for the running time of this algorithm is too high to make it suitable for any practical use with a provable approximation bound. Recently Greenhill [12] used a different Markov chain, an extension of the one used for the regular case in [7], to approximately generated non-regular graphs. The mixing time of that chain is bounded by $\Delta^{14}m^{10}$, and the algorithm only works for degree sequences that are not too far from regular. In particular, it is not applicable to power-law degree sequences of the type we consider that have γ below 3. In [2] it is explained how to use sequential importance sampling to estimate probabilities in the uniform random graph with given degrees, using statistics from non-uniform samplers. However, there are no useful theoretical bounds on the number of samples required to achieve desirable accuracy of such estimates. As stated in [2, Section 8], "Obtaining adequate bounds on the variance for importance sampling algorithms is an open research problem in most cases of interest." Using a uniform sampler completely nullifies this problem.

In this paper, we present a linear-time approximate sampler PLD* for graphs with power-law degree sequences, whose output has a distribution that differs from uniform by $o(1)$ in total variation distance. This approximate sampler, like others such as in [20, 17, 3, 23], has weaker properties than the ones from MCMC-based algorithms. Its approximation error depends on n and cannot be improved by running the algorithm longer. However, it has the big advantage of a linear running time. None of the previously proposed asymptotic approximate samplers [20, 17, 3, 23] is effective in the non-regular case when Δ is above $m^{1/3}$, whereas the type of power-law degree sequences we consider can handle Δ well above $m^{1/2}$. So PLD* greatly extends the family of degree sequences for which we can generate random graphs asymptotically uniformly and fast.

Using an appropriate rejection scheme, we tune PLD* into an exact uniform sampler PLD which, for $\gamma > 2.8811$, has a running time that is $O(n^{4.081})$ in expectation, and $O(n^{2.107})$ asymptotically almost surely (a.a.s.). Most of the running time of PLD is spent computing certain probabilities in the rejection scheme. The reason that the expected running time is much greater than our bound on the likely running time is that our algorithm has a component that computes a certain graph function. This particular component is unlikely to be needed on any given run, but it is so time-costly that, even so, it contributes immensely to the expected running time.

We next formally define power law sequences. There have been different versions in the literature. The definition by Chung and Lu [6] requires the number of vertices with degree i to be $O(i^{-\gamma}n)$, uniformly for all i. Hence the maximum degree is $O(n^{1/\gamma})$, which is well below $\sqrt{n}$. However, it has been pointed out [19] that this definition leads to misleading results, as in many real networks there are vertices with significantly higher degrees. A more realistic model [19, 22] is to consider degrees composed of independent power-law variables. In this model the maximum degree is of order $n^{1/(\gamma-1)}$, which matches the statistics in the classical preferential attachment model by Barabási and Albert [1] for modelling complex networks. In this paper, we will use a more relaxed definition [8] which includes the latter version of power-law sequences. Let X be a random variable with power law distribution with parameter $\gamma > 1$, i.e. $\mathbb{P}(X = i) = ci^{-\gamma}$, where $c^{-1} = \sum_{i=1}^{\infty} i^{-\gamma}$, then for $\gamma > 1$ we have $\mathbb{P}(X \geq x) = O(i^{1-\gamma})$. As vertices with degree 0 can be ignored in a generation algorithm, we assume that the minimum degree is at least 1.

DEFINITION 1.1. *A sequence* $\mathbf{d} = (d_1, \ldots, d_n)$ *is power-law distribution-bounded (*plib *for short) with parameter* $\gamma > 1$ *if the minimum component in* $\mathbf{d}$ *is at least* 1 *and* *there is a constant $K > 0$ independent of n such that the number of components that are at least i is at most $Kni^{1-\gamma}$ for all $i \geq 1$.*

In *plib* sequences, the maximum element can reach as high as $n^{1/(\gamma-1)}$, making both enumeration and generation of graphs with such degree sequences more challenging than the Chung-Lu version mentioned above. It is easy to see that if $\mathbf{d}$ is composed of independent power-law variables with exponent γ then $\mathbf{d}$ is a *plib* sequence with parameter γ' for any $\gamma' < \gamma$.

Our main result for the approximate sampler PLD*, defined in Sections 2 and 3.4, is as follows.

THEOREM 1.1. *Assume $\mathbf{d} = (d_1, \ldots, d_n)$ is a* plib *sequence with parameter $\gamma > 21/10 + \sqrt{61}/10 \approx 2.881024968$ and $\sum_{1\leq i\leq n} d_i$ is even. Then PLD* runs in time $O(n)$ in expectation and generates a graph with degree sequence $\mathbf{d}$ whose distribution differs from uniform by $o(1)$ in total variation distance.*

A general description of our uniform sampler, which we call PLD, will be given in Section 2, and its formal definition in Sections 3 and 6. Our result for the exact uniform sampler is the following.

THEOREM 1.2. *Assume $\mathbf{d} = (d_1, \ldots, d_n)$ is a* plib *sequence with parameter $\gamma > 21/10 + \sqrt{61}/10 \approx 2.881024968$ and $\sum_{1\leq i\leq n} d_i$ is even. Then PLD uniformly generates a random simple graph with degree sequence $\mathbf{d}$, with running time $O(n^{4.081})$ in expectation, and $O(n^{2.107})$ with high probability.*

We remark that the same statements are immediately implied for the Chung-Lu version of power law sequences (which we called power-law density-bounded in [8]), however for that version one can easily modify our methods to reduce the bound on γ somewhat. This is because the maximum degree in the Chung-Lu version is much smaller (well below $\sqrt{n}$), and moreover, all multiple edges are likely to have bounded multiplicity. Hence, the major difficult issues we face in this paper do not appear in the Chung-Lu version of power law sequences.

We remark that our argument easily gives slightly improved time complexity bounds for any particular γ in the range given in Theorem 1.2; to simplify the presentation we have taken a uniform bound on a complicated function of γ.

We describe the general framework of both PLD and PLD* in Section 2. The formal definition of PLD is given throughout Sections 3 and 6, together with key lemmas used for bounding the running time. We focus on the design of the algorithm including its definition, and the specification of various parameters involved in the algorithm definition, leaving many of the straightforward proofs to the Appendix of [11], particularly those similar to the material in [9]. We prove Theorems 1.2 and 1.1 in Sections 4 and 5.

2 General description

As in [18] and [9], we will use the pairing model. Here each vertex i is represented as a bin containing exactly d_i points. Let Φ be the set of perfect matchings of these $\sum_{1\leq i\leq n} d_i$ points. Each element in Φ is called a *pairing*, and each set of two matched points is called a *pair*. Given a pairing P, let $G(P)$ denote the graph obtained from P by contracting each bin into a vertex and represent each pair in P as an edge. Typically, $G(P)$ is a multigraph, since it can have edges in parallel (multiple edges) joining the same vertices. An edge that is not multiple we call a *single* edge. The pairs of P are often referred to as edges, as in $G(P)$. A loop is an edge with both end points contained in the same vertex. The graph is simple if it has no loops or multiple edges.

The simple rejection scheme repeatedly generates a random $P \in \Phi$ until $G(P)$ is simple. Let $\mathcal{G}_{\mathbf{d}}$ denote the set of simple graphs with degree sequence $\mathbf{d}$. An easy calculation shows that every simple graph in $\mathcal{G}_{\mathbf{d}}$ corresponds to exactly $\prod_{i=1}^{n} d_i!$ pairings in Φ. Hence, the output of the rejection scheme is uniformly distributed over $\mathcal{G}_{\mathbf{d}}$. The problem is that, when the degrees become very non-regular, it takes too many iterations for the rejection scheme to find a simple $G(P)$. With power-law degree sequences as in Theorems 1.1 and 1.2, the rejection scheme is doomed to run in super-polynomial time.

The McKay-Wormald algorithm [18] does not reject P if $G(P)$ is a multigraph. Instead, it uses switching operations (e.g. see Figures 1 and 2) to switch away the multiple edges. These switchings slightly distort the distribution of the pairings away from uniform, and so some rejection scheme is used to correct the distribution. The algorithm in [9] included major rearrangements in set-up and analysis in order to incorporate new features that reduce the probability of having a rejection, and thereby extended the family of degree sequences manageable. Both [18] and [9] only deal with multiplicities at most three. However, it is easy to see that for a *plib* degree sequence with parameter $\gamma < 3$, there are multi-edges of multiplicity as high as some power of n. Previous exact generation algorithms have never reached the density at which unbounded multiplicity occurs in the pairing model, and new considerations for the design of PLD* and PLD stem from this. One may think that switchings for low multiplicities such as in Figures 1 and 2 work for any multiplicity. However, these switchings would cause difficulties for high multiplicities because (a) the error arising in the analysis is too hard to control, and (b) incorporating unbounded multiplicities into the counting scheme would result in super-polynomial running time.

One new feature in the present paper is that we treat vertices of large degree differently. We will specify a set $\mathcal{H}$ of such vertices, which we call *heavy*. Other vertices are called *light*. An edge in P is *heavy* if its both end vertices are heavy. In particular, a loop is heavy if it is at a heavy vertex. Non-heavy edges are called *light*. The algorithm PLD* contains two stages. In the first stage, it uses some switchings, defined in Section 3.2, to turn every heavy multiple edge ij into either a non-edge, or a single edge, and to switch away all heavy loops. It can be shown that with a non-vanishing probability, the final pairing is in the following set

(2.1)

$$\mathcal{A}_0 = \Big\{ P \subseteq \Phi : \ G_{[\mathcal{H}]}(P) \text{ is simple, } \\ L(P) \le B_L;\ D(P) \le B_D;\ T(P) \le B_T; \\ G(P) \text{ contains no other types of multiple edges} \Big\},$$

where $G_{[\mathcal{H}]}(P)$ denotes the subgraph of $G(P)$ induced by $\mathcal{H}$, $L(P)$, $D(P)$ and $T(P)$ denote the number of light simple loop, light double edges, and light triple edges in P, and B_L, B_D and B_T are prescribed parameters specified in (6.19) in Section 6. PLD* restarts until the final pairing P is in $\mathcal{A}_0$, and this is the first stage of PLD*. We will show that the distribution of the output of the first stage of PLD* is almost uniform over $\mathcal{A}_0$. Algorithm PLD is PLD* accompanied with a rejection scheme so that the output of Stage 1 is uniformly distributed over $\mathcal{A}_0$. After Stage 1, PLD* enters the second stage, in which it repeatedly removes light loops, and then light triple edges using random "valid" switchings in Figures 1 and 2, which were also used in [9]. Valid switchings mean that the performance of the switching does not destroy more than one multiple edge (or loop), or create new multiple edges (or loops).

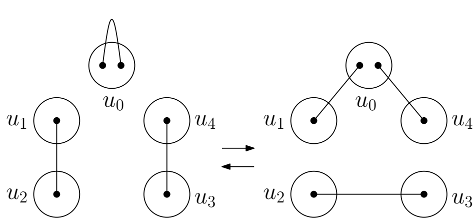


Figure 1: *Switching for a light loop*

To remove the double edges, we will use parameters $\rho_{III}(i)$, specified in Section 6.2. In each iteration, if the current pairing P contains i double edges, then with probability $\rho_{III}(i)$, it performs a random valid switching as in Figure 3, and with probability $1-\rho_{III}(i)$, it performs a random valid switching as in Figure 4. Note that the switching in Figure 4 is the typical type (called type I) used to switch away a double edge. The switching in Figure 3 is a new type (called type III) to remedy the distribution distortion caused by merely performing type I switchings.

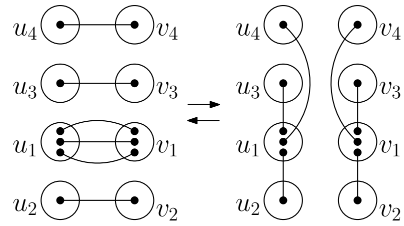


Figure 2: *Switching for a light triple edge*

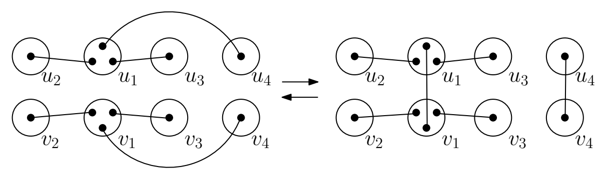


Figure 3: *Type III*

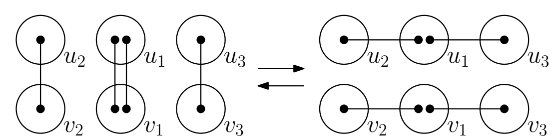


Figure 4: *Type I*

We will prove:

(a) all parameters $\rho_{III}(i)$ can be computed in time $o(n)$;

(b) the number of iterations in the second stage is $o(n)$ in expectation and with high probability;

(c) starting from a uniformly random $P \in \mathcal{A}_0$, the second stage of PLD* outputs an asymptotically uniform random graph with target degree sequence $\mathbf{d}$.

Even though the type III switching is new, and some other new considerations will be brought in to cope with various complications, the main ideas involved in the second stage of PLD* are similar to those in [9]. Indeed, we will tune PLD* into PLD with introduction of more types of switchings, and with various sorts of rejections.

The output of PLD is a uniformly random graph with the target degree sequence (see Lemma 6.14). We can show that the probability that PLD ever performs a rejection or other types of switchings is $o(1)$. This allows us to couple PLD and PLD* and deduce that the output distributions of PLD* differs from PLD, which is uniform, by $o(1)$, and thus derive our main theorem. The full details of PLD and PLD* for the second stage will be presented in Section 6.

Our major innovative ideas lie in the first stage of PLD and PLD*. We will later specify a set of pairings $\Phi_0 \subseteq \Phi$ (in Section 3.1) such that with a non-vanishing probability a random pairing in Φ is in Φ_0. Roughly speaking, Φ_0 excludes pairings in which some vertex is incident with too many heavy multiple edges or heavy loops. The algorithm starts by repeatedly generating a random pairing $P \in \Phi$ until $P \in \Phi_0$. Then, PLD* enters two phases (both in Stage one) sequentially where heavy multiple edges and then heavy loops are eliminated using switchings.

Since our analysis is tailored for the power-law degree sequence, we start by stating some properties of power-law degree sequences. In the rest of this paper, we will assume that $\mathbf{d}$ is such that $\sum_i d_i$ is even and Definition 1.1 is satisfied for some fixed $K > 0$ and $\gamma > 2.5$, that is

$$\sum_{j \geq i} N_j \leq Kni^{1-\gamma}, \quad \text{for all } 1 \leq i \leq \Delta \tag{2.2}$$

where $N_j = |t : d_t = j|$. A stronger lower bound on γ will be imposed later. We will also assume without loss of generality in the proof that $\gamma < 3$, since a *plib* sequence with parameter $\gamma \geq 3$ is also a *plib* sequence with parameter γ' for every $\gamma' < \gamma$.

Recall that Δ denotes the maximum degree. Without loss of generality we may assume that $\Delta = d_1 \geq \cdots \geq d_n \geq 1$. For every $1 \leq i \leq n$, the number of vertices with degree at least d_i is at least i, so (2.2) implies that $i \leq Kn(d_i)^{1-\gamma}$. This gives

$$d_i \leq (Kn/i)^{1/(\gamma-1)} \quad \text{for all } 1 \leq i \leq n. \tag{2.3}$$

For a positive integer k, define $M_k = \sum_{1 \leq i \leq n} [d_i]_k$ where $[x]_k$ denotes $\prod_{i=0}^{k-1}(x-i)$. It follows immediately from (2.3) using $\gamma > 2$ that

$$\begin{aligned} &\Delta \leq (Kn)^{1/(\gamma-1)} \\ &M_1 = \Theta(n) \\ &M_k = O(\Delta^k) = O(n^{k/(\gamma-1)}), \quad \forall k \geq 2. \end{aligned} \tag{2.4}$$

Let $0 < \delta < 1/2$ be a constant to be specified later. Define

$$h = n^{1-\delta(\gamma-1)}, \tag{2.5}$$

$$\mathcal{H} = \{i : 1 \leq i \leq h\}, \quad \mathcal{L} = [n] \setminus \mathcal{H}. \tag{2.6}$$

Immediately from (2.3),

$$d_h \leq K^{1/(\gamma-1)} n^{\delta}, \tag{2.7}$$

and so every vertex in $\mathcal{L}$, the set of light vertices, has degree $O(n^{\delta})$.

Finally, for $k \geq 1$ define $H_k = \sum_{v \in \mathcal{H}} [d_v]_k$ and $L_k = M_k - H_k$.

3 First stage of PLD and PLD*: heavy multiple edge reductions

Recall that PLD (PLD*) first generates a uniformly random pairing $P_0 \in \Phi_0$, where $\Phi_0 \subseteq \Phi$ is the set of pairings satisfying some initial conditions. Our next step is to specify Φ_0.

Suppose firstly that $M_2 < M_1$. In this case, we define Φ_0 to be the set of simple pairings in Φ. Janson [13] showed that under this condition, there is a probability bounded away from 0, uniformly for all n, that $P_0 \in \Phi_0$. So if P_0 is not simple, PLD (PLD*) rejects it, and otherwise, the remaining phases are defined to pass P_0 directly to the output. In this case, PLD (PLD*) is equivalent to the simple rejection scheme. Henceforth in the paper, we assume

$$M_2 \geq M_1. \tag{3.8}$$

3.1 Specifying Φ_0 Given $i < j$, let $W_{i,j}(P)$ denote the number of points in vertex i that belong to heavy loops or heavy multiple edges in P with one end in i and the other end *not* in j. Let $W_i(P)$ denote the number of pairs that are in a heavy non-loop multiple edge, with one end in i. For any $i \leq j$, let $m_{i,j} = m_{i,j}(P)$ denote the number of pairs between i and j in P. Define $\eta = \sqrt{M_2^2 H_1 / M_1^3}$ and

$$\Phi_0 = \left\{ P \in \Phi : \begin{array}{l} m_{i,j} I_{m_{i,j} \geq 2} W_{i,j} \leq \eta d_i, \quad \forall 1 \leq i, j \leq h \text{ with } i \neq j; \\ m_{i,i} W_i \leq \eta d_i, \quad \forall 1 \leq i \leq h; \\ \sum_{1 \leq i < j \leq h} m_{i,j} I_{m_{i,j} \geq 2} \leq 4M_2^2/M_1^2; \\ \sum_{1 \leq i \leq h} m_{i,i} \leq 4M_2/M_1. \end{array} \right\} \tag{3.9}$$

The following lemma ensures that the probability of an initial rejection is at most $1/4 + o(1)$.

Lemma 3.1. *Assume that $\eta = o(1)$, i.e. $M_2^2 H_1 / M_1^3 = o(1)$. If P is a random pairing in Φ then the probability that $P \in \Phi_0$ is at least $1/4 + o(1)$.*

Next we define the switchings used in Stage 1.

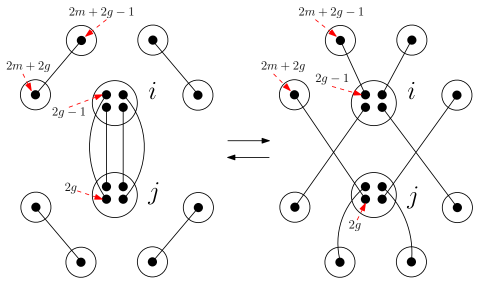


Figure 5: Switching for heavy multiple edge

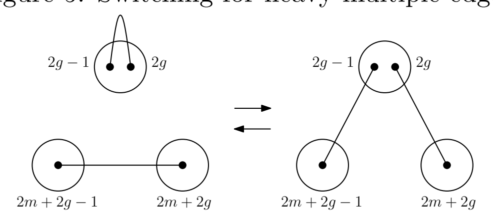


Figure 6: Switching for heavy multiple loop

3.2 Heavy switchings Given a pairing P, let $m(P,i,j)$ denote the number of pairs between i and j in P, and set $m(P,i) = m(P,i,i)$, the number of loops at i.

We first define the switchings we are going to use to switch away heavy multiple edges between a fixed pair of heavy vertices (i,j), $i < j$. The switching that we are going to define will change the multiplicity of ij from some $m \geq 1$ to 0 (i.e. no pair between i and j). We will only use this switching directly if $m(P,i,j) \geq 2$, and we will also use an inverse version with a certain probability to change the multiplicity from 0 to 1. The inverse switching is needed for the present problem because i and j may be quite likely to have an edge between them.

Definition 3.1. *(Heavy m-way switching at (i,j) on P.)*

This is defined for an ordered pair (i,j) of vertices with $m(P,i,j) = m \geq 1$. Label the endpoints of the m pairs between i and j as $2g-1$ and $2g$, $1 \leq g \leq m$, where points $1,3,\ldots,2m-1$ are contained in vertex i. Pick m distinct light pairs $x_1,\ldots,x_m$. Label the endpoints of x_g as $2m+2g-1$ and $2m+2g$. The switching operation replaces pairs $\{2g-1,2g\}$ and $\{2m+2g-1,2m+2g\}$ by $\{2g-1,2m+2g-1\}$ and $\{2g,2m+2g\}$. This is a valid switching if no new heavy multiple edges or heavy loops are created. See Figure 5.

If S is a heavy multiple edge switching creating P' from P, then the inverse of S is an inverse heavy multiple edge switching creating P from P'.

Definition 3.2. *(Heavy m-way loop switching at i.)*

This is defined if $m(P,i) = m \geq 1$. Label the endpoints of the m loops at i as $2g-1$ and $2g$, $1 \leq g \leq m$. Pick m distinct light pairs in P, labelling the endpoints of the g-th pair $2m+2g-1$ and $2m+2g$. The switching replaces pairs $\{2g-1,2g\}$ and $\{2m+2g-1,2m+2g\}$ by $\{2g-1,2m+2g-1\}$ and $\{2g,2m+2g\}$. This switching is valid if no new heavy multiple edges or heavy loops are simultaneously created. See Figure 6.

Note from the above definitions that heavy multiple edge and heavy loop switchings are permitted to create or destroy light multiple edges or light loops. This was not the case for the switchings introduced in [8] for enumeration purposes, and is another feature of our arguments not present in [18] and [9]. A heavy multiple edge switching on the pair (i,j) still does not affect other heavy multiple edges.

3.3 PLD: Phases 1 and 2 Now we are ready to define the first two phases of PLD. We define PLD* in Section 3.4.

3.3.1 Phase 1: heavy multiple edge reduction

Given a pairing P, we define an $h \times h$ symmetric matrix $\mathbf{M}(P)$ as follows. For each $1 \leq i \leq j \leq h$, if ij is not a multiple edge or loop in $G(P)$, then the ij and ji entries in $\mathbf{M}(P)$ are set as $\clubsuit$; otherwise, these entries are set as the multiplicity of the edge ij. Therefore, in this matrix, all the diagonal entries take values in $\{\clubsuit\} \cup \mathbb{N}_{\geq 1}$ and all off-diagonal entries take values in $\{\clubsuit\} \cup \mathbb{N}_{\geq 2}$ where $\mathbb{N}_{\geq k}$ denotes the set of integers that are at least k. We call $\mathbf{M}(P)$ the *signature* of P. Given such a matrix $\mathbf{M}$, we let $\mathcal{C}(\mathbf{M})$ denote the set of pairings that have $\mathbf{M}$ as their signature. Phase 1 of PLD uses the heavy m-way switchings of Definition 3.1 to switch away heavy multiple edges. Each switching step converts a pairing uniformly distributed in $\mathcal{C}(\mathbf{M})$ for some $\mathbf{M}$ to a pairing uniformly distributed in $\mathcal{C}(\mathbf{M}')$ where $\mathbf{M}'$ is obtained from $\mathbf{M}$ by setting a symmetric pair of its off-diagonal entries to $\clubsuit$. By the end of Phase 1, if no rejection occurs, the resulting pairing has a signature with all off-diagonal entries being $\clubsuit$, and hence has no heavy multiple edges.

Recall Φ_0 in (3.9). When Phase 1 starts, P_0 is uniformly distributed in Φ_0. Let $\mathbf{M}_0 = \mathbf{M}(P_0)$.

Recall that $\mathcal{H}$ denotes the set of heavy vertices. Let $\mathcal{J}$ be a list of all pairs (i,j) with $i < j$ and both $i,j \in \mathcal{H}$; i.e. $\mathcal{J}$ is an enumeration of all pairs of heavy vertices. This defines an ordering of elements in $\mathcal{J}$: $(i',j') \preceq (i,j)$ if (i',j') appears before (i,j) in the list $\mathcal{J}$. For each $k \geq 1$, let $\mathbf{M}_k$ denote the matrix obtained from $\mathbf{M}_0$ by setting the ij and ji entries of $\mathbf{M}_0$ to $\clubsuit$, for each ij among the first k entries in the list $\mathcal{J}$.

Let $k \geq 1$ and suppose (i,j) be the k-th element of

$\mathcal{J}$. For any non-negative integer m, let $\mathcal{C}(\mathbf{M}_k, i, j, m)$ denote the set of pairings with ij having multiplicity m, and all other edges between heavy vertices satisfying the signature $\mathbf{M}_k$ (ignoring its ij entry). Given $m \ge 1$ and a pairing $P \in \mathcal{C}(\mathbf{M}_k, i, j, m)$, let $f_{i,j}(P)$ denote the number of heavy m-way switchings at (i, j) where m is the multiplicity of ij in P. Given $m \ge 1$ and $P' \in \mathcal{C}(\mathbf{M}_k, i, j, 0)$, let $b_{i,j}(P', m)$ denote the number of inverse heavy m-way switchings applicable to P'; these convert P' to some $P \in \mathcal{C}(\mathbf{M}_k, i, j, m)$.

Note that $W_{i,j}(P)$ is determined by the signature of P; hence we may write $W_{i,j}(\mathbf{M}(P))$ for $W_{i,j}(P)$. For any $m \ge 1$, define

$$\overline{m}^f_{i,j}(\mathbf{M}_k, m) = m!M_1^m; \tag{3.10}$$

$$\underline{m}^f_{i,j}(\mathbf{M}_k, m) = m!(M_1 - H_1 - 2m)^m; \tag{3.11}$$

$$\overline{m}^b_{i,j}(\mathbf{M}_k, m) = [d_i - W_{i,j}]_m[d_j - W_{j,i}]_m; \tag{3.12}$$

$$\begin{aligned}\underline{m}^b_{i,j}(\mathbf{M}_k, m) = {} & [d_i - W_{i,j}]_m[d_j - W_{j,i}]_m \\ & - mh^2[d_i - W_{i,j}]_{m-1}[d_j - W_{j,i}]_{m-1},\end{aligned} \tag{3.13}$$

where $W_{i,j} = W_{i,j}(\mathbf{M}_k)$ and $W_{j,i} = W_{j,i}(\mathbf{M}_k)$.

By simple counting arguments (see [11, Lemma 28]) we have that for every $m \ge 1$ and for every $P \in \mathcal{C}(\mathbf{M}_k, i, j, m)$ and $P' \in \mathcal{C}(\mathbf{M}_k, i, j, 0)$,

$$\underline{m}^f_{i,j}(\mathbf{M}_k, m) \le f_{i,j}(P) \le \overline{m}^f_{i,j}(\mathbf{M}_k, m),$$
$$\underline{m}^b_{i,j}(\mathbf{M}_k, m) \le b_{i,j}(P', m) \le \overline{m}^b_{i,j}(\mathbf{M}_k, m).$$

Phase 1 of PLD is defined inductively as follows. For each $1 \le k \le |\mathcal{J}|$, let P_k denote the pairing obtained after the k-th step of Phase 1. To define this step, if $m(P_{k-1}, i, j) \le 1$, then $P_k = P_{k-1}$; otherwise, let $m = m(P_{k-1}, i, j)$ and do the following sub-steps:

(i) Choose a random heavy m-way switching at (i, j) on P. Let P' be the pairing created by S;

(ii) Perform an f-rejection with probability $1 - f_{i,j}(P)/\overline{m}^f_{i,j}(\mathbf{M}_k, m)$; then perform a b-rejection with probability $1 - \underline{m}^b_{i,j}(\mathbf{M}_k, m)/b_{i,j}(P', m)$;

(iii) If no f or b-rejection took place, set $P^*_k = P'$;

(iv) Choose a random *inverse* heavy 1-way switching S' at (i, j) on P'; let P'' denote the pairing created by S';

(v) With probability $1/(1 + \overline{m}^b_{i,j}(\mathbf{M}_k, 1)/\underline{m}^f_{i,j}(\mathbf{M}_k, 1))$, set $P_k = P^*_k = P'$; with the remaining probability, perform an f-rejection (w.r.t. S') with probability $1 - b_{i,j}(P')/\overline{m}^b_{i,j}(\mathbf{M}_k, 1)$ and perform a b-rejection with probability $1 - \underline{m}^f_{i,j}(\mathbf{M}_k, 1)/f_{i,j}(P'')$. If no f- or b-rejection occurred, set $P_k = P''$.

LEMMA 3.2. *The probability of an f-rejection or b-rejection during Phase 1 is $o(1)$, if $\delta > (3-\gamma)/(\gamma-2)$, $\delta > 1/(2\gamma - 3)$, $\delta < 1/2$ and $1 - \delta > 1/(\gamma - 1)$.*

3.3.2 Phase 2: heavy loop reduction The heavy loop reduction is similar to the heavy multiple edge reduction but simpler. Let $\mathcal{I}$ be an ordering of elements in $\mathcal{H}$. At step k of Phase 2, for $k \ge 1$, the algorithm switches away all loops at the k-th vertex in $\mathcal{I}$ using the heavy loop switching, if there are any. In each step, sub-steps (i) and (ii) are the same as in heavy multiple edge reduction, except that $f_{ij}(P)$ and $b_{ij}(P', m)$ are replaced by $f_i(P)$ and $b_i(P', m)$, and $\overline{m}^f_{ij}(\mathbf{M}_k, m)$ and $\underline{m}^b_{ij}(\mathbf{M}_k, m)$ are replaced by $\overline{m}^f_i(\mathbf{M}_k, m)$ and $\underline{m}^b_i(\mathbf{M}_k, m)$, defined as below:

$$\overline{m}^f_i(\mathbf{M}_k, m) = 2^m m! M_1^m; \tag{3.14}$$

$$\underline{m}^f_i(\mathbf{M}_k, m) = 2^m m!(M_1 - H_1 - 2m)^m; \tag{3.15}$$

$$\overline{m}^b_i(\mathbf{M}_k, m) = [d_i - W_i]_{2m}; \tag{3.16}$$

$$\begin{aligned}\underline{m}^b_i(\mathbf{M}_k, m) = {} & [d_i - W_i]_{2m} \\ & - mh^2[d_i - W_i]_{2m-2}.\end{aligned} \tag{3.17}$$

Moreover, P_k is set to P' as long as no f or b-rejection happend. There are no sub-steps (iii)–(v).

It is easy to show that for any $P \in \mathcal{C}(\mathbf{M}_k, i, i, m)$, $\underline{m}^f_i(\mathbf{M}_k, m) \le f_i(P) \le \overline{m}^f_i(\mathbf{M}_k, m)$; and for any $P \in \mathcal{C}(\mathbf{M}_k)$, $\underline{m}^b_i(\mathbf{M}_k, m) \le b_i(P, m) \le \overline{m}^b_i(\mathbf{M}_k, m)$.

LEMMA 3.3. *If $2/(\gamma - 1) < 1 + \delta(\gamma - 2)$ and $\delta(\gamma - 1) > 2/3$, then the probability that a rejection happens during Phase 2 is $o(1)$.*

3.3.3 Uniformity Recall

$$\Phi_2 = \{P \subseteq \Phi_0 : \ G_{[\mathcal{H}]}(P) \text{ is simple}\}.$$

We have the following lemma confirming the uniformity of the output of the first two phases.

LEMMA 3.4. *Let P_0 be a random pairing in Φ_0 and P' be the output after the first two phases, assuming that no rejection occurred. Then P' has the uniform distribution on Φ_2.*

3.4 Stage 1 of PLD*

The first stage of PLD* is just PLD without rejection. I.e. Phase 1 of PLD* consists of substeps (i), (iv) and (v) without f-rejection or b-rejection. Phase 2 of PLD* repeatedly switches away heavy loops using random heavy loop switchings.

4 Proof of Theorem 1.2

Uniformity

By Lemma 3.4, PLD generates a uniformly random pairing $P \in \Phi_2$ after the first stage. In Section 6 we show that the second stage of PLD outputs a uniformly random graphs with a plib degree sequence with parameter γ; see Lemma 6.14.

Running time

The initial generation of $P \in \Phi$ takes $O(n)$ time, as there are only $O(n)$ points in the pairing model.

If (γ, δ) satisfies all conditions in Lemmas 3.1, 3.2 and 3.3 then PLD only restarts $O(1)$ times in expectation in Stage 1. In Section 6.4 we show that if $\gamma > 21/10 + \sqrt{61}/10 \approx 2.881024968$ then there exists δ satisfying

(4.18)

$$\frac{1}{2\gamma - 3} < \delta < \min\left\{1 - \frac{1}{\gamma - 1}, \frac{2 - \frac{2}{\gamma-1} - \frac{2\gamma-3}{(\gamma-1)^2}}{3 - \gamma}\right\},$$

and that with any such choice of δ, the probability of any rejection occurring in the second stage is $o(1)$. It only remains to bound the time complexity of PLD assuming no rejection occurs.

We first bound the running time in the first two phases. Each step of the algorithm involves computing $f_{i,j}(P)$ and $b_{i,j}(P, m)$ in Phase 1, and $f_i(P)$ and $b_i(P, m)$ in Phase 2. In the next lemma we bound the computation time for such functions. The proof is in the Appendix of [11].

Lemma 4.1. *(a) The running time for computing $f_{i,j}(P)$ and $f_i(P)$ in Phases 1 and 2, given P, is $O(H_1 + |\mathcal{H}|m(P, i, j))$ and $O(H_1 + |\mathcal{H}|m(P, i, i))$ respectively.*

(b) The running time for computing $b_{i,j}(P, m)$ and $b_i(P, m)$ in Phases 1 and 2, given P, is $O(\Delta + m)$.

Next, we bound the time complexity of Phases 1 and 2.

Lemma 4.2. *The expected running time for the first two phases is $o(n)$.*

Proof. By the definition of Φ_0 in (3.9),

$$\sum_{i<j\le h} \mathbb{E}(m_{i,j} I_{m_{i,j}\ge 2}) = O(M_2^2/M_1^2)$$

for every $P \in \Phi_0$. Let $P_0 \in \Phi_0$ be the pairing obtained before entering Phase 1. Recall that $\mathbf{M}(P_0)$ is the signature of P_0 and let $m_{i,j}$ denote the ij entry of $\mathbf{M}(P_0)$ for $(i, j) \in \mathcal{J}$. Note that the heavy multiple edge switchings are applied only for $(i, j) \in \mathcal{J}$ such that $m_{i,j} \ge 2$. By Lemma 4.1, for each such (i, j), the running time is $O(m_{i,j})$ for sub-step (i), $O(H_1 + |\mathcal{H}|m_{i,j})$ for (ii), $O(1)$ for (iii) and (iv), and $O(H_1 + |\mathcal{H}|)$ for (v). Hence, the total time required for switching away each heavy multiple edge is $O(H_1 + |\mathcal{H}|m_{i,j})$ and hence, the time complexity for Phase 1 is

$$\sum_{1\le i<j\le h} O((H_1 + |\mathcal{H}|m_{i,j})I_{m_{i,j}\ge 2})$$
$$= O(H_1) \sum_{1\le i<j\le h} \mathbb{E}(m_{i,j} I_{m_{i,j}\ge 2}) = O(H_1 M_2^2/M_1^2),$$

as $|\mathcal{H}| = O(H_1)$. The above is bounded by $o(n)$ for any γ in the range of Theorem 1.2 and any choice of δ satisfying (4.18). Similarly, the time complexity for Phase 2 is $o(n)$. ∎

The analysis for the running time of Phases 3–5 is similar to that of [9]. In Section 6.2 we show that parameters $\rho_\tau(i)$ such as $\rho_{III}(i)$ can be computed inductively in at most $B_D = o(n)$ steps. Each switching step involves counting the number of certain structures (involving a bounded number of edges) in a pairing (e.g. the number of 6-cycles). Using direct brute-force counting methods yields the time complexity claimed in Theorem 1.2. The details are given in the Appendix of [11] under the heading "Running time for Phases 3–5".

5 Proof of Theorem 1.1

The approximate sampler PLD* is simply PLD without rejections, except for rejections when checking $P \in \Phi_0$ and $P \in \mathcal{A}_0$. As a result, there is no need to compute the number of certain structures, as done in PLD.

Proof of Theorem 1.1. We have already discussed above that it takes PLD, and thus PLD* as well, $O(n)$ time to find $P \in \mathcal{A}_0$ before entering the last three phases. The computation of $\rho_{III}(i)$ takes at most $B_D = o(n)$ units of time. It is easy then to see then that each iteration of PLD* in these three phases takes only $O(1)$ time, as it only involves switching several pairs, and it is shown in Section 6 that the total number of iterations in these three phases is $O(B_L + B_D + B_T) = o(n)$. Hence, PLD* is a linear time algorithm in expectation. It only remains to show that the output of PLD* is close to uniform.

Clearly PLD and PLD* can be coupled so that they produce the same output if no rejection occurs, and no types other than $\tau = I, III$ are chosen in PLD in the last three phases. By Lemma 6.14, any rejection occurs in Phase 3–5 with probability $o(1)$. We also prove (see (6.40) and Lemma 6.4) that the probability of ever performing a switching of type other than I and III is $o(1)$ (the probability is $O(\xi B_D)$ which is guaranteed $o(1)$ by Lemma 6.6). Thus, the probability that PLD and PLD* have different output is $o(1)$. Let A be an arbitrary subset of $\mathcal{G}_{\mathbf{d}}$. The probability that PLD outputs a graph in A whereas PLD* does not is $o(1)$, as that happens only if they have different outputs. Hence,

$$\mathbb{P}_{PLD}(A) - \mathbb{P}_{PLD*}(A) = o(1) \quad \text{for all } A \subseteq \mathcal{G}_{\mathbf{d}}.$$

That is, the distribution of the output of PLD* differs by $o(1)$ in total variation distance from the distribution of the output of PLD, which is uniform. ∎

6 Second stage: phases for light switchings

The second stage of PLD* has been described already in Section 2. In this section, we tune it into PLD with a rejection scheme, as well as introducing more types of switchings.

Recall that $M_k = \sum_{1\le i\le n}[d_i]_k$, $H_k = \sum_{i\in\mathcal{H}}[d_i]_k$ and $L_k = M_k - H_k$. Define

$$B_L = \frac{4L_2}{M_1}; \quad B_D = \frac{4L_2M_2}{M_1^2} \quad B_T = \frac{2L_3M_3}{M_1^3}. \tag{6.19}$$

and recall from (2.1) that

$$\mathcal{A}_0 = \Big\{P \in \Phi_2 : \begin{array}{l} L(P) \le B_L;\ D(P) \le B_D;\ T(P) \le B_T; \\ G(P) \text{ contains no other types of multiple edges} \end{array}\Big\}.$$

Roughly speaking, the only possible multiple edges in a pairing in $\mathcal{A}_0$ are loops, double edges and triple edges, and there are not too many of them. Lemma 3.4 guarantees that the resulting pairing P after the first two phases is uniformly distributed in Φ_2, and the following lemma guarantees that PLD restarts $O(1)$ times in expectation until entering the second stage, i.e. the last three phases. The proof of the lemma is presented in the Appendix of [11].

LEMMA 6.1. *Assume $L_4 = o(M_1^2)$ and $L_4M_4 = o(M_1^4)$. With probability at least $1/2$, a random $P \in \Phi_2$ is in $\mathcal{A}_0$.*

The second stage consists of Phases 3, 4 and 5 during which light multiple edges (loops, triple edges and double edges) are eliminated in turn. These phases are similar to those in [9] for regular graphs, except that here we use more types of switchings to boost several types of structures due to the non-regular degree sequence. In each phase, the number of a certain type of light multiple edges will be reduced until there are no multiple edges of that type. There will be f and b-rejections designed so that after each phase, the output is uniform conditioned on the event that the number of multiple edges of a certain type is zero. In each phase, the set of pairings is partitioned into classes $\cup_{i=0}^{i_1}\mathcal{S}_i$, where $\mathcal{S}_i$ is the set of pairings with exactly i multiple edges of the type that is treated in the phase; $i_1 \in \{B_L, B_D, B_T\}$ is a pre-determined integer whose validity is guaranteed by the definition of $\mathcal{A}_0$. Let $\mathcal{A}_i$ be the output after each phase for $i = 3, 4, 5$. In Phase 3 (and 4), the number of loops (triple edges) will be reduced by one in each step. Hence, $\mathcal{A}_4$ will be the set of pairings in $\mathcal{A}_0$ containing no loops or triple edges. The analysis for Phases 3 and 4 is quite simple, almost the same as that in [18] and [9]. We postpone this until Sections 6.3. In Phase 5 the number of light double edges is reduced. For this, we will use the boosting technique introduced in [9], but we use more booster switchings to cope with non-regular degrees. We briefly describe the framework of Phase 5 here, pointing out the major differences compared with [9], with the detailed design, definitions and analysis given in the rest of this section. Phase 5 consists of a sequence of switching steps, where each step typically eliminates one double edge using a type I switching operation. This is the same operation used in [18]. However, in some occasional switching steps, some other types of switchings are performed. The algorithm first determines certain parameters $\rho_\tau(i)$, $0 \le i \le B_D$. In each switching step, PLD chooses a switching type τ, e.g. $\tau = I$, from a given set of switching types, with probability $\rho_\tau(i)$, where i is the number of double edges in the current pairing. After choosing type τ, PLD performs a random switching of type τ. Rejections can happen in each switching step with small probabilities, and these probabilities are carefully designed to maintain the property that the expected number of times a pairing is reached, throughout the course of the algorithm, depends only on the number of double edges the pairing has. (This feature was introduced in [9] and permits the use of switchings that occasionally increase the number of double edges, and is a marked departure from the method in [18].) As a consequence, the output after Phase 5 has uniform distribution. It follows that the output of PLD is uniform over simple graphs with degree sequence $\mathbf{d}$.

The parameters $\rho_\tau(i)$ are designed to guarantee the uniform distribution of the output of PLD, and meanwhile to have a small probability of any rejection during Phase 5. They are determined as the solution of a system of equations for which we give an efficient ($o(n)$ time) computation scheme. The analysis is a little different from [9] due to the different types of switchings being used. Switchings in [9] are classified into different classes (class A and class B), whereas the switchings in the present paper can all be classified as class A under that paradigm. However, the analysis of b-rejections is more complicated than in [9], and as a convenience for the argument we need to introduce “pre-states” of a pairing, and “pre-b-rejections”. These concepts will be discussed in detail in Section 6.1.1.

As in [9], under various conditions on M_k, H_k and L_k for several small values of k, we bound the probability of any rejection during Phases 3–5 by $o(1)$ (see Lemmas 6.3, 6.6, 6.7, 6.8, 6.9, 6.11, and 6.13). These conditions set constraints on (γ, δ). Eventually we will show that if $\gamma > 21/10 + \sqrt{61}/10 \approx 2.881024968$ then there exists $\delta > 0$ satisfying all conditions required in these lemmas; see Lemma 6.14.

As mentioned before, we postpone the discussions of Phases 3 and 4 till Section 6.3 and we start our discussions of Phase 5 by assuming that the algorithm enters Phase 5 with a uniform random pairing $P \in \mathcal{A}_4$, where $\mathcal{A}_4$ is the set of pairings in $\mathcal{A}_0$ containing no loops or triple edges. By the definition of $\mathcal{A}_0$, the number of double edges in P is at most $i_1 = B_D$, and they are all light. In Phase 5, $\mathcal{S}_i$ denotes the set of pairings in $\mathcal{A}_4$ containing exactly i double edges. Then, $\mathcal{S}_0$ is the set of pairings in Φ that produce simple graphs.

6.1 Phase 5: double edge reduction

We will use five different types of switchings in the phase for the double edge reduction. We introduce one of them now, and the rest after some discussion. The most commonly used type is I, which was shown in less detail in Figure 4. We formally define it as follows.

Type I switching. Take a double edge z; note that one end of z must be a light vertex. Label the end points of the two pairs in z by $\{1, 2\}$ and $\{3, 4\}$ so that points 1 and 3 are in the same vertex and this vertex is light. Choose another two pairs x and y and label their ends by $\{5, 6\}$ and $\{7, 8\}$ respectively. Label the vertices that containing points $1 \le i \le 8$ as shown in Figure 7 (same as Figure 4 but with points and vertices labelled as in the definition). If (a) u_i, v_i, $1 \le i \le 3$ are six distinct vertices; (b) both x and y are single edges; (c) there are no edges between u_1 and u_i for $i \in \{2, 3\}$ and no edges between v_1 and v_i for $i \in \{2, 3\}$, then replace the four pairs $\{2i-1, 2i\}$, $1 \le i \le 4$ by $\{1, 5\}$, $\{3, 7\}$, $\{2, 6\}$ and $\{4, 8\}$ as shown in Figure 7.

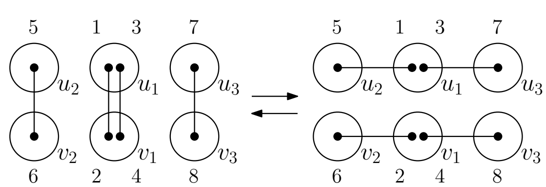


Figure 7: type I

The inverse of the above operation is called an inverse type I switching. Hence, to perform an inverse switching, we will choose a light 2-star and label the points in the two pairs by $\{5, 1\}$ and $\{3, 7\}$ so that 1 and 3 are in the same light vertex; choose another 2-star (not necessarily light) and label the points in the two pairs by $\{6, 2\}$ and $\{4, 8\}$ so that 2 and 4 are in the same vertex. Label all involved vertices as shown in Figure 7. If (a) all the six vertices are distinct; (b) both 2-stars chosen are simple; (c) there is no edge between u_i and v_i for all $1 \le i \le 3$, then replace all four pairs $\{1, 5\}$, $\{3, 7\}$, $\{2, 6\}$ and $\{4, 8\}$ by $\{2i-1, 2i\}$, $1 \le i \le 4$.

Note that from the definition of the type I switching, there can be multiple switchings of type I that switch a pairing P to another pairing P', caused by different ways of labelling the points in the switched pairs.

When a type I switching is performed, the number of double edges will decrease by one. Given a pairing P, let $\mathcal{F}_1(P)$ denote the set of mappings from the numbers 5, 1, 3, 7 to the points of P such that $\{5, 1\}$ and $\{3, 7\}$ are pairs of P and 1 and 3 are in the same vertex, which is light. We call $(\{5, 1\}, \{3, 7\})$ a *light ordered 2-star*. Define $\mathcal{F}_2(P)$ similarly from the numbers 6, 2, 4, 8 without the restriction to light vertices. Let $\mathcal{F}(P) = \mathcal{F}_1(P) \times \mathcal{F}_2(P)$. The elements of $\mathcal{F}(P)$ are called *doublets*. As $|\mathcal{F}_1(P)| = L_2$ and $|\mathcal{F}_2(P)| = M_2$, we have $|\mathcal{F}(P)| = M_2 L_2$, which is independent of P. If a type I switching switches a pairing P' to a new pairing P, then it creates a new element in $\mathcal{F}(P)$ (but not all elements in $\mathcal{F}(P)$ are created this way).

Note the labels on the points in each type I switching induce, in an obvious way, a unique doublet in the resulting pairing. Now, for a given pairing P, define $\mathcal{Z}_0(P)$ to be the set of doubles in $\mathcal{F}(P)$ that can be created in this way. Then, the number of ways that P can be created via a type I switching is $|\mathcal{Z}_0(P)|$. Since $\mathcal{Z}_0(P) \subseteq \mathcal{F}(P)$, we immediately have $|\mathcal{Z}_0(P)| \le M_2 L_2$. For a power-law distribution-bounded degree sequence with $\gamma < 3$, $|\mathcal{Z}_0(P)|$ varies a lot among pairings $P \in \mathcal{S}_i$. As we discussed before, a big variation of this number will cause a big probability of a b-rejection. It turns out that five main structures cause the undesired variation of this number: H_i, $1 \le i \le 5$ in Figure 8. In order to reduce the probability of a b-rejection, we boost pairings P such that $G(P)$ does not contain sufficient number of copies of H_0, or in other words, $G(P)$ contains many copies of H_i for some $1 \le i \le 5$. These force the use of switchings of type III, IV, V, VI and VII respectively (we leave out type II as there is a type of switching called type II used for the d-regular case which we do not use in this paper; but it is possible that using this type, together with strengthening other parts can lead to a weaker constraint on γ). These switchings will be defined in Section 6.1.3 and are shown in Figures 3 and 9 – 12. Let $\Lambda = \{I, III, IV, V, VI, VII\}$ be the set of switching types to be used. All types of switchings, except for type I, will switch a pairing to another pairing containing the same number of double edges. In other words, a type τ ($\tau \neq I$) switching does not change the number of double edges in a pairing.

The basic method in this part of the algorithm and analysis is similar to that in [9] but there are new ingredients, which include the use of pre-states. We briefly describe how and why they are used.

Given a pairing P, let $\mathcal{Z}(P)$ be the subset of $\mathcal{F}(P)$ that includes just those doublets induced (in the way described above) by applying a switching of any type to some other pairing to create P. Then, each switching (of any type) converting some P' to P induces a doublet $\mathfrak{z} \in \mathcal{Z}(P)$. However, except for type I, other type of switchings create extra pairs along with $\mathfrak{z}$. For instance, a type III switching creates one extra pair whereas a type IV switching creates four extra pairs (see Figures 3 and 9). These extra pairs modify (slightly) the expected number of times that a given $\mathfrak{z} \in \mathcal{Z}(P)$ is created. In order to account for this, we will associate with each pairing P a set of pre-states. These pre-states are basically a copy of P plus a designated $\mathfrak{z} \in \mathcal{Z}(P)$, together with several pairs, depending on how P is reached via a switching. Then, going from a pre-state to P, we may perform a pre-b-rejection. This rejection trims off the fluctuation caused by the extra pairs created together with $\mathfrak{z}$, ensuring that every $\mathfrak{z}$ is created with the same expected number of times.

For every pairing $P \in \mathcal{S}_i$, we want to make sure that the expected number of times that P is created in the phase is independent of P; we do so by equalising the expected number of times that each $\mathfrak{z} \in \mathcal{Z}(P)$ is created, after the pre-b-rejection; denote this quantity by q_i (i.e. this quantity is independent of P). Then the expected number of times that P is created, before the b-rejection, is $q_i|\mathcal{Z}(P)|$. Then, the probability of a b-rejection depends only on the variation of $|\mathcal{Z}(P)|$ over all $P \in \mathcal{S}_i$. The reason to use type III,IV,V,VI and VII aside from type I is to ensure that the variation of $|\mathcal{Z}(P)|$ is sufficiently small in a class $\mathcal{S}_i$. Let $\mathcal{Z}^*(P) = \mathcal{F}(P) \setminus \mathcal{Z}(P)$. We will prove that $|\mathcal{Z}^*(P)|$ is sufficiently small compared with $|\mathcal{Z}(P)|$.

Next, we partition $\mathcal{Z}(P)$ to $\cup_{i=0}^5 \mathcal{Z}(P)$ according to the different restrictions put on to $\mathfrak{z} \in \mathcal{Z}(P)$ when different types of switchings are applied. Let $\mathcal{Z}_i(P)$ ($1 \leq i \leq 5$) denote the set of $\mathfrak{z} \in \mathcal{F}(P)$, as shown in Figure 7, satisfying

(a) all these four pairs {5,1},{3,7},{6,2},{4,8} are contained only in single edges; v_1 is a light vertex; and all six vertices u_i and v_i, $1 \leq i \leq 3$, are distinct;

(b) for each $1 \leq i \leq 5$, $\mathfrak{z}$ corresponds to a copy of H_i in Figure 8 in $G(P)$.

Let $Z_i(P) = |\mathcal{Z}_i(P)|$ and $Z^*(P) = |\mathcal{Z}^*(P)|$. Let $\mathcal{S}_\tau^+(P)$ denote the number of type τ switchings that converts a pairing to P. Then, $\mathcal{S}_I^+(P) = Z_0(P)$. Clearly,

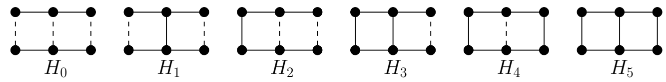


Figure 8: A pair of 2-paths

$$\sum_{i=0}^{5} Z_i(P) + Z^*(P) = |\mathcal{F}(P)| = M_2 L_2. \tag{6.20}$$

We will show later that the variation of $\sum_{i=0}^5 Z_i(P)$ caused by $Z^*(P)$ is sufficiently small for the range of γ we consider.

Note that each type $\tau \in \Lambda$ corresponds to a unique integer t so that the created element $\mathfrak{z} \in \mathcal{Z}(P)$, formed after a type τ switching creating a pairing P, is in $\mathcal{Z}_t(P)$. We define

$$t = t(\tau) \text{ to be this unique integer.}$$

For instance, $t(I) = 0$, and by looking at Figures 3, 9 and 8, we have $t(III) = 1$ and $t(IV) = 2$.

Before defining all types of switchings, we first give a general description of the whole approach.

6.1.1 A brief outline

Note that a pairing can be reached more than once in Phase 5. Let $\sigma(P)$ denote the expected number of times that pairing P is reached. Our goal is to design the algorithm in such a way that $\sigma(P) = \sigma(i)$ for every $P \in \mathcal{S}_i$ and for every $0 \leq i \leq i_1$; this ensures that the output of Phase 5, if no rejection happens, is uniform over $\mathcal{S}_0$.

Choosing a switching type.

As in [9], in Phase 5, the algorithm starts with a uniform distribution on $\mathcal{A}_4$. In each step, the algorithm starts from a pairing $P \in \mathcal{S}_i$ obtained from the previous step, and chooses a switching type according to a pre-determined "type distribution". This distribution only depends on i, i.e. the class where P is in, and is independent of P. We will use $\rho_\tau(i)$ to denote the probability that type τ switching is chosen for a pairing in $\mathcal{S}_i$. We require $\sum_\tau \rho_\tau(i) \leq 1$ for every i. If this inequality is strict, then with the remain probability, the algorithm terminates. Such a rejection is called a t-rejection.

An f-rejection.

If a type τ is chosen, then the algorithm chooses a random type τ switching S that can be performed on $P \in \mathcal{S}_i$. An f-rejection may happen in order to equalise the expected number of times a particular type

τ switching is performed, for every $P \in \mathcal{S}_i$. Let $f_\tau(P)$ denote the number of type τ switchings that can be performed on P, then, we will perform an f-rejection with probability

$$1 - \frac{f_\tau(P)}{\overline{m}_\tau(i)}, \tag{6.21}$$

where $\overline{m}_\tau(i)$ will be specified later as an upper bound for $\max_{P\in\mathcal{S}_i} f_\tau(P)$. Therefore, every switching S is chosen and is not f-rejected with probability $\rho_\tau(i)/\overline{m}_\tau(i)$. Hence, the expected number of times that S is performed during Phase 5 is

$$\sigma(i)\frac{\rho_\tau(i)}{\overline{m}_\tau(i)},$$

assuming that each pairing in $\mathcal{S}_i$ is reached with $\sigma(i)$ expected number of times.

Pre-states and pre-b-rejection.

We first give a formal definition of pre-states of a pairing $P' \in \mathcal{S}_j$. Given pairing $P' \in \mathcal{S}_j$, a switching type τ and an arbitrary $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P')$, a pre-state of P', corresponding to τ and $\mathfrak{z}$ is a copy of P' in which $\mathfrak{z}$ is designated together with a set of pairs X such that an inverse type τ switching can be performed on X and $\mathfrak{z}$. Let $\widehat{B}_\tau(P,\mathfrak{z})$ denote the set of pre-states of P corresponding to switching type τ and $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P)$ and let $\widehat{b}_\tau(P,\mathfrak{z}) = |\widehat{B}_\tau(P,\mathfrak{z})|$. Define $\widehat{b}_\tau(P) = \min_{\mathfrak{z}\in\mathcal{Z}_{t(\tau)}} \widehat{b}_\tau(P,\mathfrak{z})$. We will define later $\underline{\widehat{m}}_\tau(i)$ as a lower bound for $\min_{P\in\mathcal{S}_i} \widehat{b}_\tau(P)$. In particular, we will show that for every $0 \le i \le i_1 - 1$, $\widehat{b}_I(P,\mathfrak{z}) = 1$ for any $P \in \mathcal{S}_i$ and for any $\mathfrak{z} \in \mathcal{Z}_{t(I)}(P) = \mathcal{Z}_0(P)$ (see (6.22)), which immediately implies that

$$\underline{\widehat{m}}_I(i) = 1, \quad \text{for all } 0 \le i \le i_1 - 1. \tag{6.22}$$

Now, let P' be the pairing that S converts P into and let $\mathfrak{z}$ be the new pair of 2-stars created by S. Then, $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P')$. Assume $P' \in \mathcal{S}_j$. Note that S designates a pre-state of P'. We will perform a pre-b-rejection before entering P', to ensure that every $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P')$ are created with equal expected number of times, for every $P' \in \mathcal{S}_j$.

A pre-b-rejection will be performed in the following way. Note that our algorithm has been designed so that given τ, $P \in \mathcal{S}_i$ and $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P)$, all pre-states of P corresponding to τ and $\mathfrak{z}$ will be reached with equal expected number of times, which is

$$y_i := \sigma(i)\frac{\rho_\tau(i)}{\overline{m}_\tau(i)},$$

which is independent of P, P' and $\mathfrak{z}$. Then, a pre-b-rejection will be performed with probability

$$1 - \frac{\underline{\widehat{m}}_\tau(j)}{\widehat{b}_\tau(P',\mathfrak{z})}.$$

Now, for any $\mathfrak{z} \in \mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P')$, the expected number of times that $\mathfrak{z}$ is created is

$$\sum_{s\in\widehat{B}_\tau(P',\mathfrak{z})} y_i \frac{\underline{\widehat{m}}_\tau(j)}{\widehat{b}_\tau(P',\mathfrak{z})} = y_i\underline{\widehat{m}}_\tau(j) = \sigma(i)\frac{\rho_\tau(i)}{\overline{m}_\tau(i)}\underline{\widehat{m}}_\tau(j), \tag{6.23}$$

which is independent of $\mathfrak{z}$. This ensures that every $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}(P')$ are created with equal expected number of times, as desired. In fact, we will equalise this quantity for every switching type τ as well. This will be done by choosing $\rho_\tau(i)$ properly. We will later let $q(j)$ denote this equalised quantity, which depends only on j, i.e. the class P' is in.

A b-rejection

If S is not f-rejected or pre-b-rejected, then a b-rejection will be performed with probability

$$1 - \frac{\underline{m}(j)}{b(P')},$$

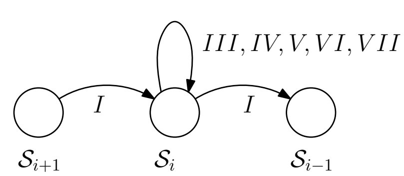
where $b(P') = |\mathcal{Z}(P')|$ and $\underline{m}(j)$ will be a specified lower bound for $\min_{P'\in\mathcal{S}_j} |\mathcal{Z}(P')|$. If no b-rejection occurred, then P is switched to P' and that completes a switching step. The expected number of times that P' is reached via a switching is

$$q(j)\underline{m}(j), \tag{6.24}$$

as the expected number of times a $\mathfrak{z} \in \mathcal{Z}(P')$ is created is equalised to q_j, as we mentioned. This confirms that the expected number of times that P' is reached is dependent only on the state in which P' lies.

The Markov chain.

This defines a Markov chain run on $\mathcal{A}_4$. Each state in the Markov chain is a pairing in $\mathcal{A}_4$ together with $\{\mathcal{O},\mathcal{R}\}$; $\mathcal{O}$ is the state representing an output and $\mathcal{R}$ is the state representing a rejection. A switching step from P to P' is a transition from P to P' in the chain. If a rejection (of any kind) happens in a step, then the chain enters state $\mathcal{R}$ and the algorithm terminates without an output. If $P \in \mathcal{S}_0$, then we interpret a type I switching on P as outputting P. So if the chain reaches a pairing in $\mathcal{S}_0$, then it enters $\mathcal{O}$ in the next step with probability $\rho_I(0)$.



Recall the following quantities that we are going to specify later:

$\overline{m}_\tau(i)$: upper bound for $\max_{P\in\mathcal{S}_i} f_\tau(P)$;

$\underline{\widehat{m}}_\tau(i)$: lower bound for $\min_{P\in\mathcal{S}_i} \widehat{b}_\tau(P) = \min_{P\in\mathcal{S}_i}\min_{\mathfrak{z}\in\mathcal{Z}_{i(\tau)}} \widehat{b}_\tau(P,\mathfrak{z})$;

$\underline{m}(i)$: lower bound for $\min_{P\in\mathcal{S}_i} b(P)$.

In the next section, we will deduce a system involving $\rho_\tau(i)$ and $\sigma(i)$ using the above quantities, and use the solution of that system to specify $\rho_\tau(i)$, the type distribution function to be used in the algorithm.

6.1.2 The system with $\rho_\tau(i)$ and $\sigma(i)$ Note that P_t is the pairing obtained after the t-th step of Phase 5. Initially, each pairing in $\mathcal{A}_4$ can be P_0 with probability $1/|\mathcal{A}_4|$.

Note that while performing switchings, a pairing can be visited more than once. Recall that $\sigma(P)$ denotes the expected number of times that a given pairing P is visited. We wish to ensure that $\sigma(P) = \sigma(i)$ for every $P \in \mathcal{S}_i$.

Given a pairing $P \in \mathcal{S}_i$, we want to equalise the expected number of times that any pair of 2-stars in $\mathcal{Z}(P)$ is created, after the pre-b-rejection. Recall that $q(i)$ denotes this quantity (see the discussion below (6.23)). We first consider switchings of type I. For such a switching S converting P' to P, P' must be in $\mathcal{S}_{i+1}$. By (6.23) (with i in (6.23) being $i+1$ and j in (6.23) being i and $\tau = I$) and (6.22),

(6.25)

$$q(i) = \sigma(i+1)\frac{\rho_I(i+1)}{\overline{m}_I(i+1)}\underline{\widehat{m}}_I(i) = \sigma(i+1)\frac{\rho_I(i+1)}{\overline{m}_I(i+1)}.$$

For any switching type $\tau \neq I$, a switching of type τ will switch a pairing to another pairing in the same class. So, for each $\tau \in \Lambda\setminus\{I\}$, let S be a switching of type τ that converts a pairing $P' \in \mathcal{S}_i$ to P. By (6.23),

$$q(i) = \sigma(i)\frac{\rho_\tau(i)}{\overline{m}_\tau(i)}\underline{\widehat{m}}_\tau(i).$$

Combining this with (6.25), we must have

$$\rho_\tau(i) = \frac{q(i)\overline{m}_\tau(i)}{\sigma(i)\underline{\widehat{m}}_\tau(i)} = \frac{\sigma(i+1)}{\sigma(i)}\frac{\rho_I(i+1)}{\overline{m}_I(i+1)}\frac{\overline{m}_\tau(i)}{\underline{\widehat{m}}_\tau(i)},$$

for each $\tau \in \Lambda\setminus\{I\}$.

Finally, by (6.24) and (6.25),

$$\sigma(i) = \underline{m}(i)q(i) + 1/|\mathcal{A}_4| = \sigma(i+1)\frac{\rho_I(i+1)}{\overline{m}_I(i+1)} + 1/|\mathcal{A}_4|,$$

where $1/|\mathcal{A}_4|$ is the probability that $P_0 = P$. The boundary condition is $q(i_1) = 0$.

Letting $x_i = \sigma(i)|\mathcal{A}_4|$, we have

(6.26)

$$x_{i_1} = 1;$$

(6.27)

$$x_i = x_{i+1}\rho_I(i+1)\frac{\underline{m}(i)}{\overline{m}_I(i+1)} + 1 \ (\forall 0 \le i \le i_1 - 1);$$

(6.28)

$$\rho_\tau(i) = \rho_I(i+1)\frac{x_{i+1}}{x_i}\frac{\overline{m}_\tau(i)}{\overline{m}_I(i+1)\underline{\widehat{m}}_\tau(i)} \quad (\forall \tau\in\Lambda\setminus\{I\}, 1 \le i \le i_1 - 1);$$

(6.29)

$$\rho_\tau(i) \ge 0;\ \rho_\tau(i_1) = 0\ (\tau \ne I);\ \sum_\tau \rho_\tau(i) \le 1.$$

It follows immediately that

$$\frac{x_{i+1}}{x_i} \le \frac{1}{\rho_I(i+1)}\frac{\overline{m}_I(i+1)}{\underline{m}(i)}. \tag{6.30}$$

We will show that there is a solution $\rho^*_\tau(i)$ and x^*_i to the above system with $\rho^*_I(i)$ close to 1 for every i and $\sum_{\tau\in\Lambda}\rho^*_\tau(i) = 1 - o(i_1)$. Then we will set the type probability $\rho_\tau(i)$ in the algorithm to be $\rho^*_\tau(i)$ for every τ and i. A lemma in [9] shows that with this setting, the expected number of times that a pairing in $\mathcal{S}_i$ is reached is $x^*_i/|\mathcal{A}_4|$, independent of $P \in \mathcal{S}_i$. Therefore, for each $P \in \mathcal{S}_0$, the probability that P is outputted is equal to $(x^*_0/|\mathcal{A}_4|)\rho_I(0)$, and thus

(6.31)

the output of PLD is uniformly distributed in $\mathcal{S}_0$.

Before proving the existence of such a solution, we need to define all types of switchings and then specify quantities $\overline{m}_\tau(i)$, $\underline{m}(i)$ and $\underline{\widehat{m}}_\tau(i)$ that have appeared in the system.

6.1.3 Switchings In the definition of a switching (of each type), we will choose a set of pairs and label their end points in a certain way; then we replace this set of pairs by another set, without changing the given degree sequence. The type I switching has been formally defined in detail. The definition of other types is clear by the illustrations in Figures 3, and 9 – 12. All pairs involved in the switchings and the inverse switchings are required to be contained only in single edges.

A type I switching creates only a pair of 2-stars in $\mathcal{Z}_0(P)$ for some P and no other extra pairs. Hence, given any $P \in \mathcal{S}_i$ ($i \le i_1 - 1$) and any $\mathfrak{z} \in \mathcal{Z}_0(P)$, there is only one pre-state of P corresponding to type I and $\mathfrak{z}$. In other words, for every $0 \le i \le i_1 - 1$, $\widehat{b}_I(P,\mathfrak{z}) = 1$ for any $P \in \mathcal{S}_i$ and for any $\mathfrak{z} \in \mathcal{Z}_{t(I)}(P) = \mathcal{Z}_0(P)$. This confirms (6.22), i.e.

$$\widehat{\underline{m}}_I(i) = 1, \quad \text{for all } 0 \le i \le i_1 - 1.$$

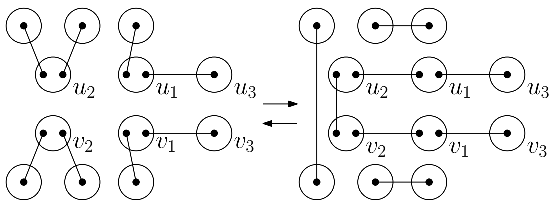


Figure 9: type IV

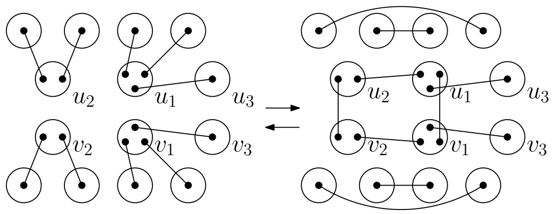


Figure 10: type V

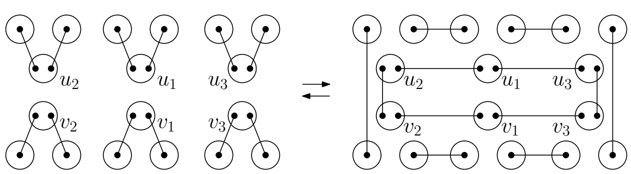


Figure 11: type VI

6.1.4 Specifying $\overline{m}_\tau(i)$, $\underline{m}(i)$ and $\widehat{\underline{m}}_\tau(i)$ Define

(6.32)
$$\overline{m}_I(i) = 4iM_1^2;\ \overline{m}_{III}(i) = M_3L_3;\ \overline{m}_{IV}(i) = 2M_2^3L_2;$$

(6.33)
$$\overline{m}_V(i) = 2M_2^2M_3L_3;\ \overline{m}_{VI}(i) = M_2^5L_2;$$
$$\overline{m}_{VII}(i) = M_2^4M_3L_3.$$

It is easy to verify that for every $P \in \mathcal{S}_i$ and $\tau \in \Lambda$, $f_\tau(P) \le \overline{m}_\tau(i)$ for each $0 \le i \le i_1$. The factor 2 in $\overline{m}_{IV}(i)$ and $\overline{m}_V(i)$ accounts for the two cases that (a) u_2 is adjacent to v_2; (b) u_3 is adjacent to v_3.

Define

$$U_1 = (Kn)^{1/(\gamma-1)}(Kn)^{(\gamma-2)/(\gamma-1)^2} = (Kn)^{(2\gamma-3)/(\gamma-1)^2}; \tag{6.34}$$

$$U_k = \frac{\gamma K^{k/(\gamma-1)}}{k+1-\gamma} n^{1-\delta(\gamma-k-1)} \quad \text{for } k \ge 2. \tag{6.35}$$

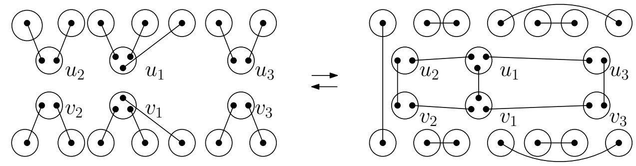


Figure 12: type VII

The U_k $(k \ge 1)$ are upper bounds for the number of various types of structures in $P \in \mathcal{A}_0$. For instance, U_1 is an upper bound for the number of 2-paths with one end at a given vertex (see the Appendix of [11] for details). These U_k will be used to define parameters such as $\underline{m}(i)$ and to bound the probabilities of f-rejections, pre-b-rejections and b-rejections in Phase 5.

Next we specify $\underline{m}(i)$ and $\widehat{\underline{m}}_\tau(i)$. Define

$$\underline{m}(i) = M_2L_2 - 8i(d_hM_2 + d_1L_2) - (2id_1^2d_h^2 + 4iU_1^2 + 8M_2U_1 + L_4). \tag{6.36}$$

and

$$\widehat{\underline{m}}_I(i) = 1;\ \widehat{\underline{m}}_{III}(i) = M_1 - 2U_1 - 4i; \tag{6.37}$$
$$\widehat{\underline{m}}_{IV}(i) = (M_1 - 2U_1 - 4i)^3;$$
$$\widehat{\underline{m}}_V(i) = (M_1 - 2U_1 - 4i)^4 \tag{6.38}$$
$$\widehat{\underline{m}}_{VI}(i) = (M_1 - 2U_1 - 4i)^6;$$
$$\widehat{\underline{m}}_{VII}(i) = (M_1 - 2U_1 - 4i)^7. \tag{6.39}$$

Simple counting arguments yield the following lemma, whose proof is presented in the Appendix of [11].

LEMMA 6.2. *(a) For every $0 \le i \le i_1$, $P \in \mathcal{S}_i$, $b(P) \ge \underline{m}(i)$.*

(b) For every $\tau \in \Lambda$, $0 \le i \le i_1$, $P \in \mathcal{S}_i$ and $\mathfrak{z} \in \mathcal{Z}_{t(\tau)}$, $\widehat{b}_\tau(P, \mathfrak{z}) \ge \widehat{\underline{m}}_\tau(i)$.

6.2 Specifying $\rho_\tau(i)$

Let $\xi = \xi_n > 0$ be a function of n to be specified later. Consider the system (6.26)–(6.29). We will show that there is a unique solution $(x_i^*, \rho_\tau^*(i))$ to the system such that $\rho_I^*(i) + \rho_{III}^*(i) = 1 - \xi$ and $\sum_{\tau \ne I, III} \rho_\tau^*(i) < \xi$ for all i.

We first specify a recursive algorithm to compute $\rho_I^*(i)$ and $\rho_{III}^*(i)$ and x_i^* satisfying (6.26)–(6.28) so that

$$\rho_I^*(i) + \rho_{III}^*(i) + \xi = 1, \quad \text{for all } i.$$

Base case: $\rho_I^*(i_1) = 1 - \xi$ and $\rho_{III}^*(i_1) = 0$. From this we can compute $x_{i_1}^*$.

Inductive step: assume that we have computed $\rho_I^*(i+1)$, $\rho_{III}^*(i+1)$ and x_{i+1}^*. For i: we can compute

$\rho_I^*(i)$, $\rho_{III}^*(i)$ and x_i^* by solving the following system.

$$\begin{aligned} x_i^* &= x_{i+1}^*\rho_I^*(i+1)\frac{\underline{m}(i)}{\overline{m}_I(i+1)}+1; \\ \rho_{III}^*(i) &= \rho_I^*(i+1)\frac{x_{i+1}^*}{x_i^*}\frac{\overline{m}_{III}(i)}{\overline{m}_I(i+1)\underline{\widehat{m}}_{III}(i)}; \\ \rho_I^*(i)+\rho_{III}^*(i) &= 1-\xi. \end{aligned}$$

Putting

$$C_1 = x_{i+1}^*\rho_I^*(i+1)\frac{\underline{m}(i)}{\overline{m}_I(i+1)}+1;$$
$$C_2 = \rho_I^*(i+1)x_{i+1}^*\frac{\overline{m}_{III}(i)}{\overline{m}_I(i+1)\underline{\widehat{m}}_{III}(i)},$$

we have

$$\begin{aligned} x_i^* &= C_1; \\ \rho_I^*(i) &= 1-\xi-\frac{C_2}{C_1}; \\ \rho_{III}(i) &= \frac{C_2}{C_1}. \end{aligned}$$

Hence, all $\rho_I^*(i)$, $\rho_{III}^*(i)$ and x_i^* can be uniquely determined, once we fix ξ. We only have to verify that the solution satisfies (6.29) by choosing ξ appropriately. Define

$$\xi = \frac{32M_2^2}{M_1^3}. \tag{6.40}$$

Lemma 6.3. *Assume $\delta + 1/(\gamma-1) < 1$ and $\xi = o(1)$. Then,*

$$\max_{0\le i\le i_1}\sum_{\tau\in\Lambda\setminus\{I,III\}}\frac{\overline{m}_\tau(i)}{\underline{\widehat{m}}_\tau(i)\underline{m}(i)} \le \xi.$$

Proof. Recall (6.32) – (6.39) and it is easy to see that $\underline{m}(i) \ge M_2L_2/2$ and each $\underline{\widehat{m}}_\tau(i)$ is at least a half of the first term in their expressions. Hence, for every $0 \le i \le i_1$,

$$\sum_{\tau\in\Lambda\setminus\{I,III\}}\frac{\overline{m}_\tau(i)}{\underline{\widehat{m}}_\tau(i)\underline{m}(i)} \le \frac{2}{M_2L_2}\Big(\frac{4M_2^3L_2}{M_1^3}+\frac{4M_2^2M_3L_3}{M_1^4} + \frac{2M_2^5L_2}{M_1^6}+\frac{2M_2^4M_3L_3}{M_1^7}\Big).$$

Using the inequality that $L_3 \le L_2d_h = O(L_2n^\delta)$, and the fact that $M_2 = O(n^{2/(\gamma-1)})$ by (2.4), and our assumptions that $\delta + 1/(\gamma-1) < 1$ and $\xi = o(1)$, the last three terms in the above formula are dominated by the first term. Hence,

$$\sum_{\tau\in\Lambda\setminus\{I,III\}}\frac{\overline{m}_\tau(i)}{\underline{\widehat{m}}_\tau(i)\underline{m}(i)} \le \frac{32}{M_2L_2}\frac{M_2^3L_2}{M_1^3} = \xi.$$

This holds uniformly for all $0 \le i \le i_1$ and this completes the proof of this lemma. ∎

Then we have the following lemma.

Lemma 6.4. *Assume $\xi = o(1)$ and $M_3L_3/M_2L_2M_1 = o(1)$. System (6.26)– (6.29) has unique solution $(x_i^*, \rho_\tau^*(i))$ with $\rho_I^*(i)+\rho_{III}^*(i) = 1-\xi$ for every i. Moreover,*

$$\begin{aligned} \rho_I^*(i) &= 1-O\left(\frac{M_3L_3}{M_2L_2M_1}\right); \quad \rho_{III}^*(i) = O\left(\frac{M_3L_3}{M_2L_2M_1}\right); \\ \rho_\tau^*(i) &= O\left(\frac{\overline{m}_\tau(i)}{\underline{m}(i)\underline{\widehat{m}}_\tau(i)}\right) \quad \textit{for all } \tau\in\Lambda\setminus\{I,III\}. \end{aligned}$$

Proof. We have proven the uniqueness. We only need to verify that the solution given above satisfies the inequalities in (6.29). It is obvious that both $x_i > 0$ for each $0 \le i \le i_1$. This implies that $\rho_{III}(i) > 0$. Next, we bound $\rho_{III}(i)$ and $\rho_\tau(i)$ for $\tau \in \Lambda\setminus\{I,III\}$. Since $C_1 \ge x_{i+1}^*\rho_I^*(i+1)\underline{m}(i)/\overline{m}_I(i+1)$ we have

$$\rho_{III}(i) = \frac{C_2}{C_1} \le \frac{\overline{m}_{III}(i)}{\underline{\widehat{m}}_{III}(i)\underline{m}(i)}.$$

By (6.32), (6.36) and (6.37),

$$\rho_{III}(i) = O\left(\frac{M_3L_3}{M_2L_2M_1}\right) = o(1).$$

This verifies that $0 < \rho_{III}(i) < 1$ and $0 < \rho_I(i) < 1$ for every i.

Moreover, by (6.28) and (6.30), for each $\tau \in \Lambda\setminus\{I,III\}$,

$$\rho_\tau(i) \le \frac{x_{i+1}^*\rho_I^*(i+1)}{x_i^*}\frac{\overline{m}_\tau(i)}{\overline{m}_I(i+1)\underline{\widehat{m}}_\tau(i)} \le \frac{\overline{m}_\tau(i)}{\underline{\widehat{m}}_\tau(i)\underline{m}(i)}.$$

Therefore,

$$\sum_{\tau\in\Lambda}\rho_\tau(i) = 1-\xi+\sum_{\tau\in\Lambda\setminus\{I,III\}}\rho_\tau(i) \le 1-\xi+\xi = 1,$$

by (6.40) and so (6.29) is satisfied. Hence, $(x_i^*, \rho_\tau^*(i))$ is the unique solution of (6.26)– (6.29) satisfying $\rho_I(i) + \rho_{III}(i) = 1-\xi$ for every $0 \le i \le i_1$. ∎

In the rest of the paper, we fix ξ as in (6.40) and we set the type distribution $\rho_\tau(i)$ to be $\rho_\tau^*(i)$, the unique solution specified in Lemma 6.4.

6.2.1 Probability of a t-rejection

Lemma 6.5. *The expected number of iterations during the phase of double edge reduction is at most $10B_D$. Moreover, with probability $1-o(1)$, the phase of double edge reduction terminates within $8B_D$ iterations.*

Proof. Let $t \geq 8B_D$ be an integer. Assume the phase does not stop before step t. Let x denote the number of steps among the first t steps that the number of double edges does not decrease. Since the number of double edges in P_0 is at most i_1, if Phase 5 lasts at least t steps then $i_1 + x - (t-x) \geq 0$, since the number of double edges in each of the x steps will increase by at most one whereas in the other $t-x$ steps it decreases by exactly one. It follows then that $x \geq (t-i_1)/2$. The probability that the number of double edges (if there is any) does not decrease in any step is bounded by $1 - \min_{1\leq i\leq i_1}\{\rho_I(i)\} < 1/10$. Hence, the probability that$x \geq (t-i_1)/2$ is at most

$$\binom{t}{(t-i_1)/2}(1/10)^{(t-i_1)/2} \leq 2^t(1/10)^{7t/16} < 0.8^t,$$

as $(t-i_1)/2 \leq (t-t/8)/2 = 7t/16$. Putting $t = 8i_1$, the above probability is clearly $o(1)$. Hence, with probability $1-o(1)$, this phase lasts for at most $8i_1$ steps. The expected number of steps in this phase is at most

$$8i_1 + \sum_{t\geq 8i_1} 0.8^t \leq 10i_1. \quad \blacksquare$$

Recall that B_D from (6.19) and recall that i_1 is set to B_D in Phase 5.

LEMMA 6.6. *Assume $B_D\xi = o(1)$. Then with probability $1-o(1)$, no t-rejection happens in Phase 5.*

Proof. By the definition of $\rho_\tau(i)$, the probability of a t-rejection in each iteration is at most ξ. By Lemma 6.5, a.a.s. the number of iterations in this phase is at most $8i_1$. Hence, the probability of a t-rejection during this phase is at most $8i_1\xi + o(1) = o(1)$. $\blacksquare$

6.2.2 Probability of an f-rejection, a pre-b-rejection and a b-rejection.

By bounding the difference between $f_\tau(P)$ from $\overline{m}_\tau(i)$ we can bound the probability of an f-rejection; similarly, by bounding the difference between $\widehat{b}(P,\mathfrak{z})$ and $\underline{\widehat{m}}_\tau(i)$, and bounding the difference between $b(P)$ and $\underline{m}(i)$, we can bound the probability of a pre-b-rejection and a b-rejection in Phase 5. The arguments are similar to that in [9]. Thus we just state the assumptions needed to have a small probability of a rejection and the proofs can be found in the Appendix of [11].

LEMMA 6.7. *Assume that $B_D\xi = o(1)$, $B_D(M_3d_h^2 + L_3d_1^2) + L_6 + M_3U_3 + M_2U_1U_2 = o(M_1^3)$ and $B_Dn^{(2\gamma-3)/(\gamma-1)^2-1} = o(1)$. The probability of an f-rejection during Phase 5 is $o(1)$.*

LEMMA 6.8. *Assume $U_1M_2L_2/M_1^2 + (M_2L_2/M_1^2)^2 = o(M_1)$. Then the probability of a pre-b-rejection during Phase 5 is $o(1)$.*

LEMMA 6.9. *Assume $(d_hM_2 + d_1L_2 + d_1^2d_h^2 + U_1^2)M_2L_2/M_1^2 + M_2U_1 + L_4 = o(M_1^2)$. The probability of a b-rejection during Phase 5 is $o(1)$.*

6.3 Phases 3 and 4: reduction of light loops and light triple edges

The switchings used in these two phases are the same as in the corresponding phases in [9], with almost the same analysis. Thus, we briefly describe the switchings, define the switching steps, and specify key parameters used in these two phases. Then we give the lemmas which give bounds on $f(P)$ and $b(P)$, and bounds on the probabilities of f-rejections and b-rejections. The proofs are almost the same as in [9] and is presented in the Appendix of [11].

When Phase 3 starts, the input is a pairing uniformly distributed in $\mathcal{A}_0$: the set of pairings P such that $G_{[\mathcal{H}]}(P)$ is simple. Set $i_1 = B_L$ in this phase. Then by (6.19), $i_1 = O(L_2/M_1)$. In Phase 3, $\mathcal{S}_i$ is the set of pairings in $\mathcal{A}_0$ containing exactly i loops.

We use the switching shown in Figure 1 to reduce the number of loops.

Define

$$\begin{aligned}\overline{m}_\ell(i) &= 2iM_1^2;\\ \underline{m}_\ell(i) &= L_2M_1 - 2d_hM_1(2i + 4B_D + 6B_T + id_h/2)\\ &\quad - L_2(2i + 4B_D + 6B_T + 6d_1 + 2U_1).\end{aligned}$$

LEMMA 6.10. *For each $P \in \mathcal{S}_i$,*

$$\overline{m}_\ell(i) - O(iM_1(i + B_D + B_T + d_1 + U_1)) \leq f(P) \leq \overline{m}_\ell(i);$$
$$\underline{m}_\ell(i) \leq b(P) \leq L_2M_2.$$

Let P_t denote the pairing obtained after step t of Phase 3. At step $t \geq 1$, choose uniformly at random a switching S applicable on P_{t-1}. Assume that $P_{t-1} \in \mathcal{S}_i$ and P is the pairing obtained if S is performed. Perform an f-rejection with probability $1 - f(P_{t-1})/\overline{m}_\ell(i)$ and a b-rejection with probability $1 - \underline{m}_\ell(i-1)/b(P)$. If no rejection happened, then set $P_t = P$. Repeat until $P_t \in \mathcal{S}_0$.

As proved in [18, 9], by the end of Phase 3, the output is uniformly distributed in $\mathcal{A}_3 = \{P \in \mathcal{A}_0 : L(P) = 0\}$.

LEMMA 6.11. *Assume $d_hM_1(B_D + B_T + B_Ld_h) + L_2(d_1 + U_1) = o(M_1^2)$. The probability of an f-rejection or a b-rejection during Phase 3 is $o(1)$.*

Phase 4 starts with a pairing P_0 uniformly distributed in $\mathcal{A}_3$. We will use the switching shown in Figure 2 to reduce the triple edges.

In this phase, i_1 is set to B_T defined in (6.19) and hence $\mathcal{A}_3 = \cup_{0\leq i\leq i_1}\mathcal{S}_i$ where $\mathcal{S}_i$ is the set of pairings in $\mathcal{A}_3$ containing exactly i triple edges.

Recalling U_k from (6.34) and (6.35), define

$$\overline{m}_t(i) = 12iM_1^3$$

(6.41)

$$\underline{m}_t(i) = M_3L_3 - 3M_3(4B_Dd_h^2 + 6id_h^2) - L_6 - 16M_3U_3 - 3L_3(4B_Dd_1^2 + 6id_1^2) - 3M_2U_1U_2.$$

LEMMA 6.12. *For each $P \in \mathcal{S}_i$,*

$$\overline{m}_t(i) - O(iM_1^2(i + B_D + d_1 + U_1)) \le f(P) \le \overline{m}_t(i);$$
$$\underline{m}_t(i) \le b(P) \le M_3L_3.$$

The algorithm in Phase 4 is similar to that in Phase 3. At step t, the algorithm chooses uniformly at random a switching S applicable on P_{t-1}. Assume that $P_{t-1} \in \mathcal{S}_i$ and P is the pairing obtained if S is performed. Perform an f-rejection with probability $1 - f(P_{t-1})/\overline{m}_t(i)$ and a b-rejection with probability $1 - \underline{m}_t(i-1)/b(P)$. If no rejection happened, then set $P_t = P$. Repeat until $P_t \in \mathcal{S}_0$. The output then is uniformly distributed in $\mathcal{A}_4 = \{P \in \mathcal{A}_0 :\ L(P) = T(P) = 0\}$.

LEMMA 6.13. *Assume $(d_1 + U_1)L_3M_3 = o(M_1^4)$ and $M_3d_h^2(B_D + B_T) + L_3(B_Dd_1^2 + B_Td_1^2) + L_6 + M_3U_3 + M_2U_1U_2 = o(M_1^3)$. The overall probability that either an f-rejection or a b-rejection occurs during Phase 4 is $o(1)$.*

6.4 Fixing δ We show that there exists δ such that the various conditions on γ and δ together with the assumptions of Lemmas 3.1, 3.3, 6.1, 6.3, 6.6, 6.7, 6.8, 6.9, 6.11, 6.13 and [11, Lemmas 29,30] are simultaneously satisfied when $\gamma > 21/10 + \sqrt{61}/10$. After removing redundant constraints (see the full details in Appendix), these conditions finally reduce to

$$f_1(\gamma) < \delta < f_2(\gamma), \quad 5/2 < \gamma < 3 \tag{6.42}$$

where

$$f_1(\gamma) = \max\left\{\frac{4/(\gamma-1) - 2}{\gamma - 2}, \frac{3-\gamma}{\gamma-2}, \frac{1}{2\gamma-3}\right\},$$

$$f_2(\gamma) = \min\left\{\frac{1}{2}, 1 - \frac{1}{\gamma-1}, \frac{2}{7-\gamma}, \frac{2 - \frac{2}{\gamma-1} - \frac{2\gamma-3}{(\gamma-1)^2}}{3-\gamma}, \frac{2 - \frac{3}{\gamma-1}}{4-\gamma}\right\}$$

It is easy to see that (6.42) is feasible whenever

$$21/10 + \sqrt{61}/10 < \gamma < 3,$$

where $21/10 + \sqrt{61}/10 \approx 2.881024968$.

For γ in that range, we may choose any

$$\frac{1}{2\gamma-3} < \delta < \frac{2 - 3/(\gamma-1)}{4-\gamma}.$$

6.5 Uniformity

LEMMA 6.14. *Assume that* **d** *is a plib sequence with parameter $\gamma > 21/10 + \sqrt{61}/10 \approx 2.881024968$ with even sum of the components. Then, the output of PLD is uniform over the set of graphs with degree sequence* **d**. *Moreover, the probability of any rejection occurring in Stage 2 (Phases 3–5) is $o(1)$.*

Proof. The uniformity follows by Lemma 3.4 and (6.31). By choosing δ satisfying (4.18), the probability of any rejection occurring in Stage 2 is $o(1)$ by Lemmas 3.1, 3.3, 6.1, 6.3, 6.6, 6.7, 6.8, 6.9, 6.11, 6.13 and [11, Lemmas 29,30]. ∎

References

[1] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *science* 286.5439 (1999): 509–512.

[2] J. Blitzstein and P. Diaconis. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. *Internet mathematics* 6.4 (2011): 489–522.

[3] M. Bayati, J. H. Kim, and A. Saberi. A sequential algorithm for generating random graphs. *Algorithmica*, 58(4):860–910, 2010.

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

[5] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European J. Combin.*, 1(4):311–316, 1980.

[6] F. Chung and L. Lu. The volume of the giant component of a random graph with given expected degrees. *SIAM J. Discrete Math.*, 20(2):395–411, 2006.

[7] C. Cooper, M. Dyer, and C. Greenhill. Sampling regular graphs and a peer-to-peer network. *Combin. Probab. Comput.*, 16(4):557–593, 2007.

[8] P. Gao and N. Wormald. Enumeration of graphs with a heavy-tailed degree sequence. *Advances in Mathematics*, 287:412–450, 2016.

[9] P. Gao and N. Wormald. Uniform generation of random regular graphs. *Proc. FOCS'15*: 1218–1230, IEEE.

[10] P. Gao and N. Wormald. Uniform generation of random regular graphs. *arXiv*: 1511.01175.

[11] P. Gao and N. Wormald. Uniform generation of random graphs with power-law degree sequences. *arXiv*: 1709.02674.

[12] C. Greenhill. The switch markov chain for sampling irregular graphs. *Proc. SODA'15*: 1564–1572, SIAM.

[13] S. Janson. The probability that a random multigraph is simple. *Combin. Probab. Comput.*, 18(1-2):205–225, 2009.

[14] M. Jerrum, B. McKay and A. Sinclair. When is a graphical sequence stable? *Random graphs, Vol. 2* (Poznań, 1989), 101–115.

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

[16] J. H. Kim and V. H. Vu. Sandwiching random graphs: universality between random graph models. *Adv. Math.*, 188(2):444–469, 2004.

[17] J. H. Kim and V. H. Vu. Generating random regular graphs. *Combinatorica*, 26(6):683–708, 2006.

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

[19] M. E. Newman. The structure and function of complex networks. *SIAM review* 45.2 (2003): 167–256.

[20] A. Steger and N. C. Wormald. Generating random regular graphs quickly. *Combin. Probab. Comput.*, 8(4):377–396, 1999. Random graphs and combinatorial structures (Oberwolfach, 1997).

[21] G. Tinhofer. On the generation of random graphs with given properties and known distribution. *Appl. Comput. Sci., Ber. Prakt. Inf* 13 (1979): 265–297.

[22] R. Van Der Hofstad. Random graphs and complex networks. *Cambridge Series in Statistical and probabilistic Mathematics* 43 (2016).

[23] J. Y. Zhao. Expand and Contract: Sampling graphs with given degrees and other combinatorial families. arXiv preprint arXiv:1308.6627 (2013).