Fast uniform generation of random graphs with given degree sequences

Andrii Arman
School of Mathematics
Monash University
Melbourne, Australia
Email: andrii.arman@monash.edu

Pu Gao
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Canada
Email: pu.gao@uwaterloo.ca

Nicholas Wormald
School of Mathematics
Monash University
Melbourne, Australia
Email: nick.wormald@monash.edu

***Abstract*—In this paper we provide an algorithm that generates a graph with given degree sequence uniformly at random. Provided that $\Delta^4 = O(m)$, where Δ is the maximal degree and m is the number of edges, the algorithm runs in expected time $O(m)$. Our algorithm significantly improves the previously most efficient uniform sampler, which runs in expected time $O(m^2\Delta^2)$ for the same family of degree sequences. Our method uses a novel ingredient which progressively relaxes restrictions on an object being generated uniformly at random, and we use this to give fast algorithms for uniform sampling of graphs with other degree sequences as well. Using the same method, we also obtain algorithms with expected run time which is (i) linear for power-law degree sequences in cases where the previous best was $O(n^{4.081})$, and (ii) $O(nd+d^4)$ for d-regular graphs when $d = o(\sqrt{n})$, where the previous best was $O(nd^3)$.**

***Index Terms*—Uniform generation, random graphs, switchings**

I. Introduction

Sampling discrete objects from a specified probability distribution is a classical problem in computer science, both in theory and for practical applications. Uniform generation of random graphs with a specified degree sequence is one such problem that has frequently been studied. In this paper we consider only the task of generating *simple* graphs, i.e. graphs with no loops or multiple edges. An early algorithm was given by Tinhofer [15], but with unknown run time. A simple rejection-based uniform generation algorithm is usually implicit for asymptotically enumerating graphs with a specified degree sequence, for example in the papers of Békéssy [3], Bender and Canfield [4] and Bollobás [5]. The run time of this algorithm is linear in n but exponential in the square of the average degree. Hence it only works in practice when degrees are small.

A big increase in the permitted degrees of the vertices was achieved by McKay and Wormald [13], and around the same time Jerrum and Sinclair [10] found an approximately uniform sampler using Markov Chain Monte Carlo (MCMC) methods. McKay and Wormald used the configuration model introduced in [5] to generate a random (but not uniformly random) multigraph with a given degree sequence. Instead of repeatedly rejecting until finding a simple graph, McKay and Wormald used a switching operation to switch away multiple edges, reaching a simple graph in the end. The algorithm is rather efficient when the degrees are not too large. In particular, for d-regular graphs it runs in expected time $O(d^3n)$ when $d = O(n^{1/3})$. (Here and in the following we assume n is the number of vertices.) Jerrum and Sinclair's Markov chain mixes in time polynomial in n provided that the degree sequence satisfies a condition phrased in terms of the numbers of graphs of given degree sequences. In particular, the mixing time is polynomial in the d-regular case for any function $d = d(n)$. These two benchmark research papers led the study into two different research lines. More switching-based algorithms for exactly uniform generation were given which deal with new degree sequences permitting vertices of higher degrees. The regular case was treated by Gao and Wormald [7] for $d = o(\sqrt{n})$ with time complexity again $O(d^3n)$, and very non-regular but still quite sparse degree sequences (such as power law) [8] were considered by the same authors. Various MCMC-based algorithms have been investigated for generating the graphs with distribution that is only approximately uniform, e.g. algorithms by Cooper, Dyer and Greenhill [9], Greenhill [9], Kannan, Tetali and Vempala [11]. These algorithms can cope with a much bigger family of degree sequences than the switching-based algorithms. That these do not produce the exactly uniform distribution might be irrelevant for practical purposes, if it were not for the fact that the theoretically provable mixing bounds are too big. For instance, the mixing time was bounded by $d^{24}n^9 \log n$ in [6] in the regular case. We note that there have also

Second author was supported by ARC DP160100835 and NSERC. Third author was supported by ARC DP160100835.

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00084

been switching-based approximate samplers that run fast (in linear or sub-quadratic time), for instance see paper of Bayati, Kim and Saberi [2], Kim and Vu [12], Steger and Wormald [14] and Zhao [16]. For those algorithms, the bounds on error in the output distribution are functions of n which tend to 0 as n grows, but cannot be reduced for any particular n by running the algorithm longer. In this way they differ from the MCMC-based algorithms, which are fully-polynomial almost uniform generators in the sense of [10].

The goal of this paper is to introduce a new technique for exactly uniform generation. Using it to modify switching-based algorithms, we can obtain vastly reduced run times. In particular, we obtain a linear-time, i.e. $O(M)$, algorithm that works for the same family of degree sequences as the $O(M^2\Delta^2)$ algorithm in [13]. We first review the salient features of the latter.

The algorithm first generates an initial random multigraph in expected time that is linear in M. The initial pairing contains no loops of multiplicity at least two, no multiple edges of multiplicity at least three, and has a sublinear number of loops and double edges. The algorithm then uses an operation called d-switching to sequentially "switch away" all the double edges (loops are treated similarly so we ignore them at present). Provided that a multigraph G was uniform in the class of graphs with m_2 double edges, the result of applying a random d-switching to G is a random multigraph G' that is slightly non-uniformly distributed in a class of multigraphs with $m_2 - 1$ double edges. The following rejection scheme is used to equalise probabilities. Let $f_d(\tilde{G})$ be the the number of ways that a d-switching can be performed on $\tilde{G}$ and $b_d(\tilde{G})$ be the number of d-switchings that can create $\tilde{G}$. Assume that $\overline{f}_d(m)$ and $\underline{b}_d(m)$ are uniform upper and lower bounds for $f_d(\tilde{G})$ and $b_d(\tilde{G})$ respectively over all multigraphs with m double edges. If a switching that converts some multigraph G to a multigraph G' is selected by the algorithm, then the switching is accepted with probability $f_d(G)\underline{b}_d(m_2 - 1)/\overline{f}_d(m_2)b_d(G')$, and rejected otherwise. If the switching is accepted, it is applied to the multigraph, whereas rejection requires re-starting the algorithm from scratch. Computing $b_d(G')$ takes $O(M^2\Delta^2)$ time, which dominates the time complexity of [13].

The algorithm presented in this paper is obtained from the algorithm in [13] by modifying the time-consuming rejection scheme. First, it was observed in [13] that the rejection can be separated into two distinct steps, which are given the explicit names f- and b-rejection in [7]. The f-rejection step rejects the selected switching with probability $1 - f_d(G)/\overline{f}_d(m_2)$, and the b-rejection step rejects it with probability $1 - \underline{b}_d(m_2 - 2)/b_d(G')$. It is easy to see that the overall probability of accepting the switching is the same as specified originally above. By a slick observation, there is essentially no computation cost for computing the probability of f-rejection. (See the explanations in [1]). The modification in the present paper is to further separate b-rejections into a sequence of sub-rejections by a scheme we will call *incremental relaxation*. This scheme will still maintain uniformity of the multigraphs created.

The basic idea of incremental relaxation, as used in the present paper, can be described as follows. Let H be a (small) graph with each edge designated as positive or negative. We say that an H*-anchoring* of a graph G is an injection $Q : V(H) \to V(G)$ that maps every positive edge of H to an edge of G, and every negative edge to a non-edge of G. (This is a generalisation of rooting at a subgraph, which usually corresponds to the case that H has positive edges only.)

Now assume that an H-anchored graph (G, Q) is chosen u.a.r., i.e. each such ordered pair with G in some given set $\mathcal{O}$, and Q, an H-anchoring of G, is equally likely. We can convert this to a random graph $G \in \mathcal{O}$ by finding the number $b(G)$ of H-anchorings of G, and accepting G with probability $\underline{b}(\mathcal{O})/b(G)$ where $\underline{b}(\mathcal{O})$ is a lower bound on the number of H-anchorings of any element $G' \in \mathcal{O}$. However, computing $b(G)$ corresponds to computing $b_d(G')$ as described above and can be time-consuming. The key idea of our new method is that we *incrementally* relax the constraints imposed on G by Q, so that rejection is split into a sequence of sub-rejections. Set $\emptyset = V_0 \subseteq V_1 \subseteq \cdots \subseteq V_k = V(H)$ and let Q_i denote the restriction of Q to V_i. With this definition, for each i, Q_i is an $H[V_i]$-anchoring of G. Thus Q_i determines some subset (increasing with i) of the constraints on G corresponding to the edges of H, and given that (G, Q_i) is uniformly random, we can obtain a uniformly random anchoring (G, Q_{i-1}) by applying a similar rejection strategy, but using only the number $b(G, Q_{i-1})$ of ways that Q_{i-1} can be extended to an $H[V_i]$-anchoring of G. This procedure of incremental relaxation of constraints can be highly advantageous if for each i, $b(G, Q_{i-1})$ can be computed much faster than $b(G)$. In this way, a sequence of uniformly random objects is obtained, involving anchorings at ever-smaller subgraphs of H, until the empty subgraph is reached, corresponding to obtaining G u.a.r.

To see that this idea applies to the problem at hand, we observe that the existence of a d-switching (defined in Section IV-B) from G to G' forces G' to include a set A of edges (the positive edges, forming two paths of length 2, in a copy of a certain graph H), and to exclude a set B (the negative edges, forming a matching, in H). So G' comes accompanied by an H-anchoring.(Refer to right side of Figure 2 for a drawing of H.) To apply incremental relaxation we first compute the number of ways to complete such an anchoring given the first 2-path and use that to obtain a random 2-path-anchored graph, and then relax the 2-path anchoring in a similar manner. The details of applying this scheme to d-switchings are given in Section IV-B.

In Section III we present the incremental relaxation technique in a more general setting, avoiding injections but instead employing more arbitrary sets of constraints. We apply the incremental relaxation scheme in detail in the case $\Delta^4 = O(M)$ (e.g. $d = O(n^{1/3})$ in the regular degree case) in Sections IV. The switchings we use are exactly the same as those in [13]. When the incremental relaxation scheme is combined with the new techniques introduced in [7, 8], it allows us to obtain fast uniform samplers of graphs for the family of degree sequences permitted in [7, 8]. In particular, we obtain a linear-time algorithm to generate graphs with power-law degrees, and a sub-quadratic-time algorithm to generate d-regular graphs when $d = o(n^{1/2})$. See [1] for detailed descriptions of these algorithms.

II. Main results

Let $\mathbf{d} = (d_1, \ldots, d_n)$ be specified where $M = \sum d_i$ is even. Let $\Delta = \max\{d_1, \ldots, d_n\}$ and for positive integers j define $M_j = \sum_{i=1}^{n} d_i(d_i - 1)\ldots(d_i - j + 1)$. We say that $\mathbf{d}$ is *graphical* if there exists a simple graph with degree sequence $\mathbf{d}$. For the rest of this paper we only consider graphical sequences $\mathbf{d}$. Our first result is that our algorithm INC-GEN uniformly generates a random graph with degree sequence $\mathbf{d}$ and runs in linear time provided that $\mathbf{d}$ is "moderately sparse". The description of INC-GEN is given in Section IV. The proof of the uniformity will be presented in Section IV-C, and the time complexity is bounded in Section IV-D.

Theorem 1. *Let* $\mathbf{d}$ *be a graphical sequence. Algorithm* INC-GEN *uniformly generates a random graph with degree sequence* $\mathbf{d}$*. If* $\Delta^4 = O(M)$ *then the expected run time of* INC-GEN *is* $O(M)$.

Our second algorithm INC-REG is an almost-linear-time algorithm to generate random regular graphs. The run time is $O(dn+d^4)$ when $d = o(n^{1/2})$. This improves the $O(d^3n)$ run time of the uniform sampler in [7].

Theorem 2. *Algorithm* INC-REG *uniformly generates a random* d*-regular graph. If* $d = o(n^{1/2})$ *then the expected run time of* INC-REG *is* $O(dn + d^4)$.

Our third algorithm INC-POWERLAW is a linear-time algorithm to generate random graphs with a power-law degree sequence. A degree sequence $\mathbf{d}$ is said to be *power-law distribution-bounded* 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 $i \geq 1$. Note that the family of power-law distribution-bounded degree sequences covers the family of degree sequences arising from n i.i.d. copies of a power-law random variable. Uniform generation of graphs with power-law distribution-bounded degree sequences with parameter $\gamma > 21/10 + \sqrt{61}/10 \approx 2.881024968$ was studied in [8], where a uniform sampler was described with expected run time $O(n^{4.081})$. This was the first known uniform sampler for this family of degree sequences. With our new rejection scheme, we improve the time complexity to linear.

Theorem 3. *Let* $\mathbf{d}$ *be a power-law distribution-bounded degree sequence with parameter* $\gamma > 21/10+\sqrt{61}/10 \approx 2.881024968$*. Algorithm* INC-POWERLAW *uniformly generates a random graph with degree sequence* $\mathbf{d}$*, and the expected run time of* INC-POWERLAW *is* $O(n)$.

Algorithms INC-REG and INC-POWERLAW are adapted from [7, 8], where the rejection scheme is replaced by the new incremental relaxation. The detailed descriptions for the adaptations are presented in [1].

Algorithms INC-GEN and INC-REG can easily be modified if $\mathbf{d}$ represents a bipartite graph's degree sequence. As an example, we present algorithm INC-BIPARTITE in Section V as the bipartite version of INC-GEN.

Theorem 4. *Algorithm* INC-BIPARTITE *uniformly generates a random graph with bipartite degree sequence* $\mathbf{d} = (\mathbf{s}, \mathbf{t})$*. If* $\Delta^4 = O(M)$ *then the expected run time of* INC-BIPARTITE *is* $O(M)$.

III. Uniform generation by incremental relaxation

We provide here a general description of the relaxation procedure, so it can be applied in different setups. Let $\mathcal{F}$ and k be given, where $\mathcal{F}$ is a finite set and k is a positive integer. We are also given S_i, for $1 \leq i \leq k$, where each S_i is a multiset consisting of subsets of $\mathcal{F}$. Let $\otimes$ denote the Cartesian product, and let $\mathcal{F}_k$ be any subset of $\mathcal{F} \times S_1 \times \cdots \times S_k$ such that each $(G, C_1, \ldots, C_k) \in \mathcal{F}_k$ satisfies $G \in C_k \subseteq \cdots \subseteq C_1$. Given $F = (G, C_1, \ldots, C_k) \in \mathcal{F}_k$, define $P_i(F) = (G, C_1, \ldots, C_i)$ for each $1 \leq i < k$. For each $i \in [k-1]$ set $\mathcal{F}_i = \{P_i(F) : F \in \mathcal{F}_k\}$ and set $\mathcal{F}_0 = \mathcal{F}$.

For any $i \in [k]$ and $F := (G, C_1, \ldots, C_i) \in \mathcal{F}_i$, define $P(F) = (G, C_1, \ldots, C_{i-1}) \in \mathcal{F}_{i-1}$; i.e. $P(F)$ is the prefix of F.

Later in our applications of relaxation, we will let $\mathcal{F}$ be a set of multigraphs. Each element F of $\mathcal{F}_i$ can be identified with a multigraph that contains a specified substructure (determined by the C_i-s) on a specified set of vertices. In terms of the notation introduced in Section I, elements of $\mathcal{F}_i$ will correspond to $H[V_i]$-anchorings of multigraphs for some graph H and some sequence $\emptyset = V_0 \subseteq V_1 \subseteq \cdots \subseteq V_k = V(H)$. Permitting multiple copies of elements in S_i is useful in the case where two distinct constraints may correspond to the same subset of $\mathcal{F}$. This happens in our applications due to the symmetry of the substructures in H.

Next we define a procedure Loosen, which takes an $F = (G, C_1, \ldots, C_i) \in \mathcal{F}_i$ as input, and outputs an $P(F) \in \mathcal{F}_{i-1}$ with a certain probability and otherwise 'rejects' it and terminates. Our Relaxation Lemma (Lemma 5 below) shows that if F is uniformly distributed in $\mathcal{F}_i$ then the output of Loosen is uniformly distributed in $\mathcal{F}_{i-1}$.

For $0 \leq i \leq k-1$ and $F \in \mathcal{F}_i$, let $b(F)$ be the number of $F' \in \mathcal{F}_{i+1}$ such that $P(F') = F$. In other words, $b(F)$ is the number of ways to extend F to an element of $\mathcal{F}_{i+1}$. Let $\underline{b}(i)$ be a lower bound on $b(F)$ over all $F \in \mathcal{F}_i$, and assume that for all $i \in [k-1]$, $\underline{b}(i) > 0$. For $F \in \mathcal{F}_i$ with $i \geq 1$ we define the following procedure.

procedure Loosen(F):
Output $P(F)$ with probability $\underline{b}(i-1)/b(P(F))$,
and reject otherwise.

Procedure Relax is defined for $F = (G, C_1, \ldots, C_k) \in \mathcal{F}_k$. It repeatedly calls Loosen until reaching a $G \in \mathcal{F}_0$. We say that procedure Relax performs *incremental relaxation* on $(G, C_1, \ldots, C_k)$.

```
procedure Relax(F):
i ← k;
while i ≥ 1 do
  | (G, C_1, ..., C_{i-1}) = Loosen(G, C_1, ..., C_i);
  | i ← i − 1.
end
Output G.
```

Lemma 5 (Relaxation Lemma)**.** *Assume that $i \in [k]$ and $\underline{b}(i-1) > 0$. Provided that $F \in \mathcal{F}_i$ is chosen uniformly at random, the output of* Loosen(F) *is uniform in $\mathcal{F}_{i-1}$ assuming no rejection.*

Proof. Let $p = \frac{1}{|\mathcal{F}_i|}$. For any $F' \in \mathcal{F}_{i-1}$, the probability that Loosen outputs F' is equal to

$$\sum_{F \in \mathcal{F}_i \,:\, P(F) = F'} \mathbb{P}(A_F)\mathbb{P}(\text{no rejection} \mid A_F),$$

where A_F denotes the event that the input of Loosen is F. The second probability above is the conditional probability that no rejection occurs in Loosen, given A_F. By our assumption, the first probability above is always equal to p. By the definition of Loosen, the second probability above is equal to $\underline{b}(i-1)/b(F')$. By definition, $b(F')$ is exactly the number of $F \in \mathcal{F}_i$, such that $P(F) = F'$, so the sum has exactly $b(F')$ terms, each of which is equal to $p\underline{b}(i-1)/b(F')$. Hence, the probability for Loosen to output F' is equal to $p\underline{b}(i-1)$, for every $F' \in \mathcal{F}_{i-1}$. ∎

Recalling that $\mathcal{F}_0 = \mathcal{F}$, the Relaxation Lemma immediately yields the following corollary for the uniformity of Procedure Relax.

Corollary 6. *Assume that for all $i \in [k]$, $\underline{b}(i-1) > 0$, and assume $F \in \mathcal{F}_k$ is chosen uniformly at random. Then the output of* Relax(F) *is uniform in $\mathcal{F}$, if there is no rejection.*

The description of Relax as repeated calls of Loosen is useful for analysing the algorithm, but for practical implementations we refer to the following corollary.

Corollary 7. *Procedure* Relax*, when applied to $(G, C_1, \ldots, C_k) \in \mathcal{F}_k$, outputs G with probability $\prod_{i=0}^{k-1} \underline{b}(i)/b(G, C_1, \ldots, C_i)$, and ends in rejection otherwise.*

In practice, we predefine the numbers $\underline{b}(i)$. Once the numbers $b(G, C_1, \ldots, C_i)$ are computed, the b-rejection can be performed in one step using Corollary 7, and there is no need to perform Relax with its iterated calls to Loosen. As mentioned in Section I, these numbers can be much faster to compute than the number of H-anchorings of G, which would be required using the scheme in [13]. We also reiterate that, unlike the scheme in [13], the rejection probability depends on the anchoring imposed by C_k, as well as G.

IV. Algorithm INC-GEN

In this section we provide a description of INC-GEN. Let $\mathbf{d}$ be given. We will use the configuration model [5] to generate a random pairing, defined as follows. For every $1 \leq i \leq n$, represent vertex v_i as a bin containing exactly d_i points. Take a uniformly random perfect matching over the set of points in the n bins. Call the resulting matching P a *pairing* and call each edge in P a *pair*. Finally identify the bins as vertices, and represent each pair in P as an edge. This produces a multigraph from P, denoted by $G(P)$. If a set of pairs in P form a multiple edge or loop in $G(P)$ then this set of pairs is called a multiple edge in P as well, with the same multiplicity as it has in $G(P)$. A loop is a pair with both ends contained in the same bin/vertex. If there is a set containing more than one pair with all ends contained in the same vertex, then this set of pairs form a multiple loop. We always use loop to refer to a single loop with multiplicity equal to one. We call a multiple edge with multiplicity 2 or 3 a double or triple edge respectively.

Let $\Phi(\mathbf{d})$ denote the set of all pairings with degree sequence $\mathbf{d}$. Define

$$B_1 = \frac{M_2}{M}, \quad B_2 = \left(\frac{M_2}{M}\right)^2, \tag{1}$$

if $22\Delta^3 < M_2$ and define $B_1 = B_2 = 0$ otherwise. Let Φ_0 denote the set of pairings in $\Phi(\mathbf{d})$ where there are no multiple edges with multiplicity at least 3, and no multiple loops with multiplicity at least 2, and the number of loops and double edges are at most B_1 and B_2 respectively. The following result is essentially contained in [13] so we only give a brief description of the proof.

Lemma 8. *Let $\mathbf{d}$ be a graphical degree sequence with $\Delta^4 = O(M)$ and P be a uniformly random pairing in $\Phi(\mathbf{d})$. Then there exists a constant $0 < c < 1$ such that $\mathbb{P}(P \in \Phi_0) > c$ for all sufficiently large M.*

Proof. We first note that if $22\Delta^3 \geq M_2$, then since M is large enough and $\Delta^4 = O(M)$, we have $M_2/M \to 0$. So we only need to consider the case when B_1 and B_2 are defined by (1).

If $\Delta^4 = o(M)$ then the claim follows by [13, Lemmas 2 and 3′]. If $\Delta^4 = \Theta(M)$ then P contains $O(\Delta^4/M)$ triple edges in expectation, whereas the expected number of multiple edges of higher multiplicity in the pairing is bounded by $o(1)$. In the case that the expected number of triple edges is asymptotically a positive constant, the standard method of moments can be used to show that the joint distribution of the numbers of triple edges, double edges and loops are asymptotically independent Poisson variables. This implies our assertion. See also the discussion of this case in the proof of [13, Theorem 3]. ∎

The first step of our algorithm is to use the configuration model to generate a uniformly random pairing $P \in \Phi(\mathbf{d})$. Proceed if $P \in \Phi_0$. Otherwise, reject P and restart the algorithm. This type of rejection is called *initial rejection*. By Lemma 8, this initial rejection stage takes only $O(1)$ rounds in expectation before successfully producing a multigraph $G = G(P)$ with at most B_2 double edges, at most B_1 loops, and no multiple loops or edges of multiplicity higher than two.

Then the algorithm calls two procedures, `NoLoops` and `NoDoubles`. Each of these is composed of a sequence of *switching steps*. In each switching step, a loop (in `NoLoops`) or a double edge (in `NoDoubles`) will be removed using the corresponding switching operation in the procedure. Various types of rejections may occur in procedures `NoLoops` and `NoDoubles`. In all cases, if a rejection occurs then the algorithm restarts from the first step.

Algorithm `INC-GEN`$(n, \mathbf{d})$:
Generate a uniformly random pairing $P \in \Phi(\mathbf{d})$.
Reject P if $P \notin \Phi_0$ (initial rejection) and
otherwise set $G = G(P)$;
`NoLoops`(G);
`NoDoubles`(G).

Let $\mathbf{m} = (m_1, m_2)$ and $\mathcal{G}_{\mathbf{m}}$ be the set of multigraphs with degree sequence $\mathbf{d}$, m_1 loops, m_2 double edges and no other types of multiple edges. The following lemma guarantees uniformity of the multigraph obtained after initial rejection.

Lemma 9. *Let P be a uniformly random pairing in Φ_0. Let $\mathbf{m} = (m_1, m_2)$ where $m_1 \leq B_1$ and $m_2 \leq B_2$. Conditional on the number of loops and double edges in P being m_1 and m_2, $G(P)$ is uniformly distributed over $\mathcal{G}_{\mathbf{m}}$.*

Proof. This follows from the simple observation that every pairing in Φ_0 appears with the same probability, and every multigraph in $\mathcal{G}_{\mathbf{m}}$ corresponds to exactly $\prod_{i=1}^{n} d_i!/2^{m_1+m_2}$ distinct pairings. ∎

Note that if $22\Delta^3 \geq M_2$, then $B_1 = 0$, $B_2 = 0$ and so `INC-GEN` never calls `NoLoops` or `NoDoubles`. By Lemma 9, output of `INC-GEN` is a uniformly distributed in $\mathcal{G}_{0,0}$. Also, by Lemma 8, `INC-GEN` restarts constant number of times in expectation before outputting a graph. Hence, in this case we proved Theorem 1. For the rest of this section we assume $22\Delta^3 < M_2$.

In the next subsection we define the procedure `NoLoops`. This procedure uses the same switchings as in [13] (but applied to multigraphs rather than pairings) to reduce the number of loops to 0.

A. `NoLoops`

Definition 10 (ℓ-switching)**.** *For a graph $G \in \mathcal{G}_{m_1,m_2}$, choose five distinct vertices $v_1, \ldots, v_5$ such that*

- *there is a loop on v_2.*
- *v_1v_4 and v_3v_5 are single edges;*
- *there are no edges between v_1 and v_2, v_2 and v_3, v_4 and v_5.*

An ℓ-switching replaces loop on v_2 and edges v_1v_4, v_3v_5, by edges v_1v_2, v_2v_3 and v_4v_5.

See Figure 1 for an illustration of an ℓ-switching. Note that this switching is the same as the one used in [13], except performed on graphs, not pairings.

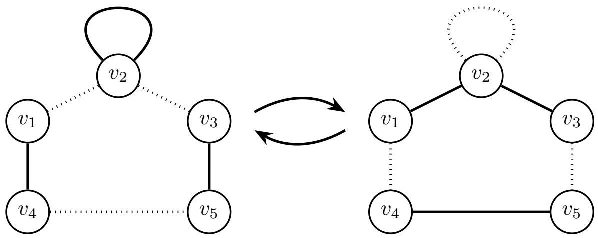

Fig. 1: ℓ-switching.

Let $f_\ell(G)$ be the number of ℓ-switchings that can be performed on G. We will specify a parameter $\overline{f}_\ell(\mathbf{m})$ such that

$$f_\ell(G) \leq \overline{f}_\ell(\mathbf{m}) \quad \text{for all } G \in \mathcal{G}_{\mathbf{m}}.$$

In each switching step, a uniformly random switching S converting $G \in \mathcal{G}_{m_1,m_2}$ to some $G' \in \mathcal{G}_{m_1-1,m_2}$ is selected. An f-rejection occurs with probability $1 - f_\ell(G)/\overline{f}_\ell(\mathbf{m})$. We will next describe how to use incremental relaxation to do b-rejection. If S is neither f-rejected nor b-rejected, then S will be performed in this switching step.

We first give some notation. In a multigraph, a *(simple) ordered edge* is an ordered pair of vertices (u, v) such that uv is a (simple) edge in the multigraph. Similarly, a *(simple) ordered i-path* is an ordered set of vertices $(u_1, \ldots, u_{i+1})$ such that $u_1 u_2 \cdots u_{i+1}$ forms a (simple) i-path in the multigraph.

Define $b_\ell(G', \emptyset)$ to be the number of simple ordered 2-paths uvw in G' such that there is no loop on v. For a simple ordered 2-path uvw in G' define $b_\ell(G', uvw)$ to be the number of simple ordered edges $u'w'$ in G' that are vertex disjoint from uvw and such that uu' and ww' are non-edges. For $\mathbf{m} = (m_1 - 1, m_2)$ let $\underline{b_\ell}(\mathbf{m}; 0)$ and $\underline{b_\ell}(\mathbf{m}; 1)$ be lower bounds on $b_\ell(G', \emptyset)$ and $b_\ell(G', uvw)$ respectively over all $G' \in \mathcal{G}_{\mathbf{m}}$ and all simple ordered 2-paths uvw in G'. Positive constants $\underline{b_\ell}(\mathbf{m}; 0)$ and $\underline{b_\ell}(\mathbf{m}; 1)$ will be defined in Section IV-A1. Any switching S that can be used to create a fixed multigraph $G' \in \mathcal{G}_{m_1-1,m_2}$ from multigraphs in $\mathcal{G}_{m_1,m_2}$ can be identified with the ordered set of vertices $\overline{V}_2(S) = (v_1, \ldots, v_5)$ whose adjacencies were changed by S. Set $\overline{V}_0(S) = \emptyset$ and $\overline{V}_1(S) = (v_1, v_2, v_3)$.

Informally, each iteration of `NoLoops` starts with a multigraph $G \in \mathcal{G}_{m_1,m_2}$ and chooses a random ℓ-switching S that converts G to some $G' \in \mathcal{G}_{m_1-1,m_2}$. In terms of the notation defined in Section I, each such switching S can be viewed as an H-anchoring of G', where H is a graph on the right side of Figure 1 (with positive signs on solid edges, and negative signs on dashed edges). `NoLoops` then performs f-rejection, after which every pair $(G', \overline{V}_2(S))$ (denoting an H-anchoring of G'), where $G' \in \mathcal{G}_{m_1-1,m_2}$ and S is an ℓ-switching that creates G', arises with the same probability. After that `NoLoops` sequentially relaxes constraints enforced by H-anchoring of G' by performing a b-rejection. The following is the formal description of `NoLoops`.

procedure `NoLoops`(G):
while *G has a loop* **do**
  let $\mathbf{m} = (m_1, m_2)$ be such that $G \in \mathcal{G}_{\mathbf{m}}$;
  obtain $(G', \overline{V}_2(S))$ from G by performing a random ℓ-switching S on G;
  f-rejection: **restart** with probability $1 - \frac{f_\ell(G)}{\overline{f}_\ell(\mathbf{m})}$;
  $\mathbf{m} \leftarrow (m_1 - 1, m_2)$;
  b-rejection: **restart** with probability $1 - \frac{\underline{b_\ell}(\mathbf{m};0)\underline{b_\ell}(\mathbf{m};1)}{b_\ell(G,\overline{V}_0(S))b_\ell(G,\overline{V}_1(S))}$;
  $G \leftarrow G'$;
end

In Section IV-C we show that if G is distributed uniformly at random in $\mathcal{G}_{m_1,m_2}$, the output of `NoLoops`(G) is uniform in $\mathcal{G}_{0,m_2}$. We do this by showing that the quantities $b_\ell(G, \overline{V}_0(S))$ and $b_\ell(G, \overline{V}_1(S))$ defined above coincide with the quantities $b(G, C_1)$ and $b(G, C_1, C_2)$ in an application of Corollary 7.

1) Parameters in `NoLoops`*:* We now specify the values of the parameters mentioned above, which will be shown in the following lemma to satisfy the required inequalities. Define

$$\overline{f}_\ell(\mathbf{m}) = m_1 M^2,$$
$$\underline{b_\ell}(\mathbf{m}; 1) = M\left(1 - \frac{6\Delta^2 - 4\Delta}{M}\right),$$
$$\underline{b_\ell}(\mathbf{m}; 0) = M_2\left(1 - \frac{8m_2\Delta + m_1\Delta^2}{M_2}\right).$$

Recall that we assumed $22\Delta^3 < M_2$ and so $\underline{b_\ell}(\mathbf{m}; 0)$ and $\underline{b_\ell}(\mathbf{m}; 1)$ are positive constants. The following Lemma establishes necessary bounds on $b_\ell(G, \emptyset)$, $b_\ell(G, uvw)$ and $f_\ell(G)$.

Lemma 11. *Let $G \in \mathcal{G}_{m_1,m_2}$ with $m_1 \leq M_2/M$ and $m_2 \leq M_2^2/M^2$. For any simple ordered 2-path $v_1v_2v_3$ in G, we have*

$$\underline{b_\ell}(\mathbf{m}; 0) \leq b_\ell(G, \emptyset) \leq M_2,$$
$$\underline{b_\ell}(\mathbf{m}; 1) \leq b_\ell(G, v_1v_2v_3) \leq M.$$

For forward ℓ-switchings

$$m_1 M^2 \left(1 - \frac{11\Delta^2 - 4\Delta + 4}{M}\right) \leq f_\ell(G) \leq \overline{f}_\ell(\mathbf{m}).$$

This completes the description of `NoLoops`.

B. `NoDoubles`

After `NoLoops` is finished, we have a multigraph $G \in \mathcal{G}_{0,m_2}$. Next we describe how to reduce the number of double edges in G.

Definition 12 (d-switching)**.** *For a graph $G \in \mathcal{G}_{0,m_2}$, choose six distinct vertices $v_1, \dots, v_6$ such that*

- *there is a double edge between v_2 and v_5.*
- *v_1v_4, v_3v_6, are single edges;*
- *the following are non-edges: v_1v_2, v_2v_3, v_4v_5, v_5v_6.*

A d-switching replaces double edges between v_2v_5 and edges v_1v_4, v_3v_6, by edges v_1v_2, v_2v_3, v_4v_5, v_5v_6.

See Figure 2 for an illustration.

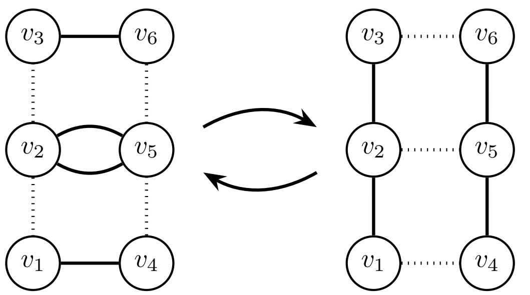

Fig. 2: d-switching.

For a graph $G \in \mathcal{G}_{\mathbf{m}}$, we use notation $f_d(G)$ for the number of ways to perform a d-switching on G. We will specify $\overline{f}_d(\mathbf{m})$ such that

$$f_d(G) \le \overline{f}_d(\mathbf{m}) \quad \text{for all } G \in \mathcal{G}_{\mathbf{m}}.$$

In each switching step, a uniformly random switching S converting $G \in \mathcal{G}_{0,m_2}$ to some $G' \in \mathcal{G}_{0,m_2-1}$ is selected. An f-rejection occurs with probability $1 - f_d(G)/\overline{f}_d(\mathbf{m})$.

The incremental relaxation scheme for b-rejection is analogous to that in `NoLoops`. Define $b_d(G', \emptyset)$ to be the number of simple ordered 2-paths uvw in G'. For a simple ordered 2-path uvw in G' define $b_d(G', uvw)$ to be the number of simple ordered 2-paths $u'v'w'$ that are vertex disjoint from uvw such that uu', vv' and ww' are non-edges.

For $\mathbf{m} = (0, m_2 - 1)$ let $\underline{b}_d(\mathbf{m}; 0)$ and $\underline{b}_d(\mathbf{m}; 1)$ be positive lower bounds (to be specified in Section IV-B1) on $b_d(G', \emptyset)$ and $b_d(G', uvw)$ over all $G' \in \mathcal{G}_{\mathbf{m}}$ and simple ordered 2-paths uvw in G'. For a d-switching S let $\overline{V}_2(S) = (v_1, \dots, v_6)$ be the vertices whose adjacencies were changed by S. Set $\overline{V}_0(S) = \emptyset$ and $\overline{V}_1(S) = (v_1, v_2, v_3)$.

As in case of `NoLoops`, in Section IV-C we show the desired uniformity property holds for `NoDoubles` .

procedure `NoDoubles`(G):
while *G has a double edge* **do**
  let $\mathbf{m} = (0, m_2)$ be such that $G \in \mathcal{G}_{\mathbf{m}}$;
  obtain $(G', \overline{V}_2(S))$ from G by performing a random d-switching S on G;
  f-rejection: **restart** with probability $1 - \frac{f_d(G)}{\overline{f}_d(\mathbf{m})}$;
  $\mathbf{m} \leftarrow (0, m_2 - 1)$;
  b-rejection: **restart** with probability $1 - \frac{\underline{b}_d(\mathbf{m};0)\underline{b}_d(\mathbf{m};1)}{b_d(G,\overline{V}_0(S))b_d(G,\overline{V}_1(S))}$;
  $G \leftarrow G'$;
end

1) Parameters for `NoDoubles`*:* Define

$$\begin{aligned}
\overline{f}_d(\mathbf{m}) &= 2m_2M^2, \\
\underline{b}_d(\mathbf{m}; 0) &= M_2\left(1 - \frac{8m_2\Delta}{M_2}\right), \\
\underline{b}_d(\mathbf{m}; 1) &= M_2\left(1 - \frac{4m_2(2\Delta - 3) + 3\Delta^3}{M_2}\right).
\end{aligned}$$

Note that $\underline{b}_d(\mathbf{m}; 0)$ and $\underline{b}_d(\mathbf{m}; 0)$ are positive constants, as in Section IV-A1.

Lemma 13. *Let $G \in \mathcal{G}_{0,m_2}$. Then for any simple ordered 2-path $v_1v_2v_3$ in G we have*

$$\begin{aligned}
&\underline{b}_d(\mathbf{m}; 0) \le b_d(G, \emptyset) \le M_2, \\
&\underline{b}_d(\mathbf{m}; 1) \le b_d(G, v_1v_2v_3) \le M_2, \\
&2m_2M^2\left(1 - \frac{12\Delta^2 - 4\Delta + 8}{M}\right) \le f_d(G) \le \overline{f}_d(\mathbf{m}).
\end{aligned}$$

C. Uniformity

Theorem 14. `INC-GEN` *generates graphs with degree sequence $\mathbf{d}$ uniformly at random.*

Proof. We start the proof by showing that b-rejection in both `NoLoops` and `NoDoubles` can be performed as `Relax` for appropriate choice of $\mathcal{F}, S_1, S_2$. We deal here with `NoDoubles` only, as the issues with `NoLoops` are identical.

Let $\mathcal{S}$ be the set of d-switchings that convert a multigraph in $\mathcal{G}_{0,m_2}$ to some multigraph in $\mathcal{G}_{0,m_2-1}$. Recall that switching $S \in \mathcal{S}$ can be identified with an ordered set of vertices $\overline{V}_2(S) = (v_1, \dots, v_6)$ whose adjacencies were changed by S, and $\overline{V}_0(S) = \emptyset$, $\overline{V}_1(S) = (v_1, v_2, v_3)$.

Let $\mathcal{F} = \mathcal{G}_{0,m_2-1}$ and let $v_1, \dots, v_6$ be distinct vertices. Using the notation $\{\}^*$ to denote a multiset, and

$E_1(G)$ to denote the set of simple edges in G, define

$$C_1^{(v_1,v_2,v_3)} = \{\tilde{G} \in \mathcal{F} : v_1v_2, v_2v_3 \in E_1(\tilde{G})\},$$
$$C_2^{(v_1,\ldots,v_6)} = \{\tilde{G} \in C_1^{(v_1,v_2,v_3)} : v_4v_5, v_5v_6 \in E_1(\tilde{G}) \text{ and } v_1v_4, v_2v_5, v_3v_6 \notin E(\tilde{G})\},$$
$$S_1 = \{C_1^{(v_1,v_2,v_3)} : v_1, v_2, v_3 \text{ all distinct}\}^*,$$
$$S_2 = \{C_2^{(v_1,\ldots,v_6)} : v_1, \ldots, v_6 \text{ all distinct}\}^*,$$
$$\mathcal{F}_2 = \{(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}) : v_1, \ldots, v_6 \text{ all distinct}, G \in C_2^{(v_1,\ldots,v_6)}\},$$
$$\mathcal{F}_0 = \mathcal{F}.$$

Recall that

$$\mathcal{F}_1 = \{(G, C_1^{(v_1,v_2,v_3)}) : (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}) \in \mathcal{F}_2 \text{ for some } v_4, v_5, v_6\}.$$

We now show that

$$\mathcal{F}_1 = \{(G, C_1^{(v_1,v_2,v_3)}) : v_1, v_2, v_3 \text{ all distinct}, G \in C_1^{(v_1,v_2,v_3)}\}.$$

Indeed, for a given simple ordered 2-path $v_1v_2v_3$ in G, the number of simple ordered 2-paths $v_4v_5v_6$ such that v_1v_4, v_2v_5 and v_3v_6 are non-edges is equal to $b_d(G, v_1v_2v_3)$ and is at least one according to Lemma 13. So for every pair $(G, C_1^{(v_1,v_2,v_3)})$ with $G \in C_1^{(v_1,v_2,v_3)}$ there exists a simple ordered 2-path $v_4v_5v_6$, such that $(G, C_1^{(v_1,v_2,v_3)}, C_1^{(v_1,\ldots,v_6)}) \in \mathcal{F}_2$, which establishes the desired claim for $\mathcal{F}_1$.

Similarly we have

$$\mathcal{F}_0 = \{G : (G, C_1^{(v_1,v_2,v_3)}) \in \mathcal{F}_1 \text{ for some } v_1, v_2, v_3\}.$$

If S is a switching from G to G', we have that $G' \in C_1^{\overline{V}_1(S)}$ and $G' \in C_2^{\overline{V}_2(S)}$ so $(G', C_1^{\overline{V}_1(S)}, C_2^{\overline{V}_2(S)}) \in \mathcal{F}_2$. So every pair $(G', \overline{V}_2(S))$, where switching $S \in \mathcal{S}$ creates G', can be identified with an element $(G', C_1^{\overline{V}_1(S)}, C_2^{\overline{V}_2(S)}) \in \mathcal{F}_2$, hence we can apply `Relax` to $(G', \overline{V}_2(S))$. In this setup, the quantities $b(G')$ and $b(G', C_1^{\overline{V}_1(S)})$ (as in Section III) are equal to $b_d(G', \overline{V}_0(S))$ and $b_d(G', \overline{V}_1(S))$ respectively. (Recall the definitions for $b_d(G', \overline{V}_0(S))$ and $b_d(G', \overline{V}_1(S))$ in Section IV-B.) It remains to note that we can set $\underline{b}(i) = \underline{b}_d(\mathbf{m}; i)$ for $i \in \{0, 1\}$ where $\mathbf{m} = (0, m_2 - 1)$.

According to Corollary 7, `Relax`$(G', C_1^{\overline{V}_1(S)}, C_2^{\overline{V}_2(S)})$ outputs G' with probability

$$\underline{b}(0)\underline{b}(1)/b(G', C_1^{\overline{V}_1(S)})b(G', C_1^{\overline{V}_1(S)}, C_2^{\overline{V}_2(S)}),$$

which is exactly equal to the probability that G' is not b-rejected in `NoDoubles`.

Hence b-rejection in `NoDoubles` is just an effective implementation of `Relax`$(G', C_1^{\overline{V}_1}, C_2^{\overline{V}_2})$. As a result of Corollary 6 we have the following

Claim 15. *Let $(G', \overline{V}_2(S))$ be chosen u.a.r from the class of all pairs $(\tilde{G}, \overline{V}_2(\tilde{S}))$, where $\tilde{G} \in \mathcal{G}_{m_1,m_2}$ ($\tilde{G} \in \mathcal{G}_{0,m_2}$) and $\tilde{S}$ is an ℓ (d)-switching that creates G'. If G' is not b-rejected by* `NoLoops` *(*`NoDoubles`*), then G' is uniform in $\mathcal{G}_{m_1-1,m_2}$ ($\mathcal{G}_{0,m_2-1}$).*

Now we are ready to prove the theorem. Assume that we initially generated a graph in $G_0 \in \mathcal{G}_{m_1,m_2}$ for some $m_1 \leq M_2/M$ and $m_2 \leq M_2^2/M^2$.

We say that a graph G was reached in `NoLoops` if a switching creating G was selected in a switching step, and G was not rejected. Let G_t denote the multigraph reached after t switching steps of `NoLoops`, if no rejection occurred (let $G_t = \emptyset$ if a rejection occurs during the t-th step or earlier). We will prove by induction on t, that conditional on $G_t \in \mathcal{G}_{m_1,m_2}$, G_t is uniformly distributed in G_{m_1,m_2}. The base case $t = 0$ holds by Lemma 9. Assume $t \geq 0$ and G_t is uniformly distributed in $\mathcal{G}_{m_1,m_2}$. Then, there exists σ_{m_1,m_2} such that the probability that G is reached after t switching steps is equal to σ_{m_1,m_2}, for every $G \in \mathcal{G}_{m_1,m_2}$. Now, for every $G' \in \mathcal{G}_{m_1-1,m_2}$ and every ℓ-switching S that results in G', the probability that $(G', \overline{V}_2(S))$ was obtained during the $(t+1)$-st iteration of `NoLoops` and not f-rejected is equal to

$$\sigma_{m_1,m_2} \frac{1}{f_\ell(G)} \frac{f_\ell(G)}{\overline{f}_\ell(m_1, m_2)} = \frac{\sigma_{m_1,m_2}}{\overline{f}_\ell(m_1, m_2)}.$$

So, $(G_{t+1}, \overline{V}_2(S))$ is uniform in class of all pairs $(\tilde{G}, \overline{V}_2(\tilde{S}))$, where $\tilde{G} \in \mathcal{G}_{m_1-1,m_2}$ and $\tilde{S}$ is an ℓ-switching that creates $\tilde{G}$. By Claim 15, if $(G_{t+1}, \overline{V}_2(S))$ is not b-rejected then G_{t+1} is uniform in $\mathcal{G}_{m_1-1,m_2}$. Inductively, the output of `NoLoops` is uniform in $\mathcal{G}_{0,m_2}$ provided no rejection. This holds as well for `NoDoubles`. Therefore, `INC-GEN` generates every graph in $\mathcal{G}_{0,0}$ with the same probability. ∎

D. Time complexity

The initial generation of the pairing takes $O(M)$ time, during which we can record the locations of loops and multiple edges and these are updated after each switching operation.

The leading contribution to the time complexity in [13] is from the computation of $b_\ell(G')$ and $b_d(G')$, whereas in `INC-GEN` the leading contribution is from computing $b_\alpha(G, \emptyset)$ and $b_\alpha(G, v_1v_2v_3)$, for $\alpha \in \{\ell, d\}$. With a simple inclusion-exclusion argument, and a brute-force search, it is easy to see that each of these numbers can be computed in $O(\Delta^3)$ time, by exploring the third neighbourhood of a specified vertex. We can improve it to $O(\Delta^2)$ by properly employed data structures. The expected number of switching steps in `INC-GEN` is $O(\Delta^2)$, and therefore the total running time is bounded by $O(M + \Delta^4) = O(M)$. The detailed analysis is presented in [1].

V. Bipartite graphs

With some minor modification our algorithm can be adjusted for generation of bipartite graphs with one part X having degrees $\mathbf{s} = (s_1, \ldots, s_m)$ and the other part Y having degrees $\mathbf{t} = (t_1, \ldots, t_n)$. Define

$$M = \sum_{i \in X} s_i = \sum_{j \in Y} t_j;$$
$$S_2 = \sum_{i \in X} s_i(s_i - 1); \quad T_2 = \sum_{j \in Y} t_j(t_j - 1).$$

The algorithm `INC-BIPARTITE` first uses the configuration model to generate a uniformly random pairing P with bipartite degree sequence $(\mathbf{s}, \mathbf{t})$. The configuration model for a bipartite degree sequence is similar to the one for a general degree sequence, except that points in vertices of X are restricted to be matched to points in vertices of Y. Let $\Phi(\mathbf{s}, \mathbf{t})$ denote the set of pairings with bipartite degree sequence $(\mathbf{s}, \mathbf{t})$, and $\Phi_0 \subseteq \Phi(\mathbf{s}, \mathbf{t})$ be those containing at most $S_2 T_2 / M^2$ double edges and no other types of multiple edges. An initial rejection is applied if $P \notin \Phi_0$.

The following lemma, which is based on Lemmas 2B and 3B′ from [13], guarantees that the probability of an initial rejection is bounded away from 1, provided $\Delta^4 = O(M)$.

Lemma 16. *Let P be a uniformly random pairing in $\Phi(\mathbf{d})$. There exists a constant $0 < c < 1$ such that $\mathbb{P}(P \in \Phi_0) > c$ for all sufficiently large n.*

To remove the double edges, Algorithm `INC-BIPARTITE` uses the bipartite version of the d-switching operation in Section IV, in which vertices v_2, v_4, v_6 are in X and vertices v_1, v_3, v_5 are in Y.

We define $b_d(G', \overline{V}(S))$ as before and we redefine

$$\underline{b}_d(\mathbf{m}; 0) = T_2 \left(1 - \frac{4m_2\Delta}{T_2}\right),$$
$$\underline{b}_d(\mathbf{m}; 1) = S_2 \left(1 - \frac{4m_2\Delta + 4\Delta^2 + 3\Delta^3}{S_2}\right)$$

Following a similar proof we have the following bipartite version of Lemma 13.

Lemma 17. *Let $G' \in \mathcal{G}_{0,m_2}$ with $m_2 \le S_2T_2/M^2$. Then for any simple ordered 2-path $v_1v_2v_3$ in G' we have*

$$\underline{b}_d(\mathbf{m}; 0) \le b_d(G, \emptyset) \le T_2$$
$$\underline{b}_d(\mathbf{m}; 1) \le b_d(G, v_1v_2v_3) \le S_2$$
$$m_2 M^4 \left(1 - \frac{8m_2 + 6\Delta^2 + 20\Delta}{M}\right) \le f_d(G) \le \overline{f}_d(\mathbf{m}).$$

Now we modify `NoDoubles` in Section IV by using the bipartite version of the d-switching operation, and the new definition of the parameters $\underline{b}_d(\mathbf{m}; i)$. Algorithm `INC-BIPARTITE` is given as follows.

procedure `INC-BIPARTITE`$(\mathbf{s}, \mathbf{t})$:
Generate a uniformly random $P \in \Phi(\mathbf{s}, \mathbf{d})$.
Initial reject if $P \notin \Phi_0$;
Construct $G = G(P)$;
`NoDoubles`(**G**);

Theorem 4 follows by a proof almost identical to that of Theorem 1.

References

[1] A. Arman, P. Gao, and N. Wormald, *Fast uniform generation of random graphs with given degree sequences*, arXiv preprint arXiv:1905.03446 (2019).
[2] M. Bayati, J. H. Kim, and A. Saberi, *A sequential algorithm for generating random graphs*, Algorithmica **58** (2010), no. 4, 860–910.
[3] A. Békéssy, *Asymptotic enumeration of regular matrices*, Stud. Sci. Math. Hungar. **7** (1972), 343–353.
[4] E. A Bender and E R. Canfield, *The asymptotic number of labeled graphs with given degree sequences*, Journal of Combinatorial Theory, Series A **24** (1978), no. 3, 296–307.
[5] B. Bollobás, *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*, European Journal of Combinatorics **1** (1980), no. 4, 311–316.
[6] C. Cooper, M. Dyer, and C. Greenhill, *Sampling regular graphs and a peer-to-peer network*, Combinatorics, Probability and Computing **16** (2007), no. 4, 557–593.
[7] P. Gao and N. Wormald, *Uniform generation of random regular graphs*, SIAM Journal on Computing **46** (2017), no. 4, 1395–1427.
[8] ———, *Uniform generation of random graphs with power-law degree sequences*, Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms, 2018, pp. 1741–1758.
[9] C. Greenhill, *The switch markov chain for sampling irregular graphs*, Proceedings of the twenty-sixth annual acm-siam symposium on discrete algorithms, 2014, pp. 1564–1572.
[10] M. Jerrum and A. Sinclair, *Fast uniform generation of regular graphs*, Theoretical Computer Science **73** (1990), no. 1, 91–100.
[11] R. Kannan, P. Tetali, and S. Vempala, *Simple markov-chain algorithms for generating bipartite graphs and tournaments*, Random Structures & Algorithms **14** (1999), no. 4, 293–308.
[12] J. H. Kim and V. H Vu, *Generating random regular graphs*, Proceedings of the thirty-fifth annual acm symposium on theory of computing, 2003, pp. 213–222.
[13] B. D McKay and N. C Wormald, *Uniform generation of random regular graphs of moderate degree*, Journal of Algorithms **11** (1990), no. 1, 52–67.
[14] A. Steger and N. C Wormald, *Generating random regular graphs quickly*, Combinatorics, Probability and Computing **8** (1999), no. 4, 377–396.
[15] G. Tinhofer, *On the generation of random graphs with given properties and known distribution*, Appl. Comput. Sci., Ber. Prakt. Inf **13** (1979), 265–297.
[16] J. Y Zhao, *Expand and contract: Sampling graphs with given degrees and other combinatorial families*, arXiv preprint arXiv:1308.6627 (2013).