

Uniform generation of random regular graphs*

Pu Gao[†]
 Monash University
 jane.gao@monash.edu

Nicholas Wormald[‡]
 Monash University
 nick.wormald@monash.edu

Abstract

We develop a new approach for uniform generation of combinatorial objects, and apply it to derive a uniform sampler *REG* for d -regular graphs. *REG* can be implemented such that each graph is generated in expected time $O(nd^3)$, provided that $d = o(\sqrt{n})$. Our result significantly improves the previously best uniform sampler, which works efficiently only when $d = O(n^{1/3})$, with essentially the same running time for the same d . We also give a linear-time approximate sampler *REG**, which generates a random d -regular graph whose distribution differs from the uniform by $o(1)$ in total variation distance, when $d = o(\sqrt{n})$.

1 Introduction

Research on uniform generation of random graphs is almost as old as modern computing. Tinhofer [17] gave a generation algorithm in which the probabilities of the graphs are computed *a posteriori*. In theory, this could be used to produce any desired distribution by rejection sampling, but no explicit bounds on the time complexity of this method are known. The earliest method useful in practice for achieving exactly uniform generation arose from the enumeration methods of Békéssy, Békéssy and Komlós [1], Bender and Canfield [3] and Bollobás [4]. This works in linear expected time for graphs with bounded maximum degree d , though the multiplicative constant behaves like $e^{(d-1)^2/4}$ as a function of d (making it rather impractical for moderately large d , even $d = 12$). It generalises easily to a simple algorithm for uniform generation of graphs with given degrees. (See, for example, [18], which in addition gives an algorithm for 3-regular graphs that has linear deterministic time.) The algorithm starts by generating a pairing, to be defined below, uniformly at random. If the corresponding graph is simple then it is outputted. Otherwise, the algorithm is restarted.

A big advance on exactly uniform generation, which significantly relaxed the constraint on the maximum degree, was by McKay and Wormald [12]. Their algorithm efficiently and uniformly generates random graphs with given degrees as long as the maximum degree is $O(M^{1/4})$ where M is the total degree. A case of particular interest is the generation of

*The conference version will appeared in *Proc. FOCS 2015*.

[†]Research supported by NSERC while this author was affiliated with University of Waterloo.

[‡]Research supported by Australian Laureate Fellowships grant FL120100125.

d -regular graphs, where the expected running time is $O(nd^3)$, for any $d = O(n^{1/3})$. This is currently the best exactly uniform sampler for regular graphs and graphs with given degrees.

When uniform generation of some class of objects seems difficult, a fallback position is to investigate approximate solutions. One approach is to use the Markov Chain Monte Carlo (MCMC) method. In this, an ergodic Markov chain on the set of graphs with given degrees is designed so that the stationary distribution is uniform. Then, the random graph obtained after taking a sufficiently large number of steps (i.e. the so-called mixing time of the chain) has distribution that is close to uniform. Jerrum and Sinclair [9] gave a fully polynomial time approximation scheme (FPTAS) for approximate uniform generation of graphs with given degrees. Their algorithm works for a large class of degree sequences. In particular, it works for all regular graphs. The bound on the mixing time, and hence the runtime of the algorithm required for any guarantee of approximation to the uniform distribution, is a polynomial in n (and is not specified, nor optimised, in their paper). Kannan, Tetali and Vempala [10] used another Markov chain, whose transitions are defined by switching two properly chosen edges in a certain way, to approximately sample random regular bipartite graphs. Again, they proved that the mixing time is polynomial in n without specifying an asymptotic bound. This Markov chain was further extended by Cooper, Dyer and Greenhill [5] for generation of random d -regular graphs. They showed that the mixing time is then bounded by roughly $d^{24}n^9 \log n$. Very recently, Greenhill [6] extended that result to the non-regular case, with a bound on the mixing time of $\Delta^{14}M^{10} \log M$ where Δ and M denote the maximum and total degrees respectively. This result applies only for $3 \leq \Delta \leq \frac{1}{4}\sqrt{M}$. These MCMC-based algorithms generate every graph with a probability within a factor $1 \pm \epsilon$ of the probability in the uniform distribution, and $\epsilon > 0$ can be made arbitrarily small by running the chain sufficiently long. So the output of such algorithms is almost as good as that from an exactly uniform sampler, for any practical use. However, the provable bounds on the degree of the polynomials involved are too high for any practical use. Actually, there is a general belief that MCMC algorithms such as this do mix much faster, and produce near-uniform results much more quickly, than what has been proved. If this were proved, they could give practical algorithms for approximately uniform generation. Without such a proof, one can only guess the accuracy of the results.

There is a variation of MCMC called *coupling from the past* [15], which is capable of producing a target distribution precisely. However, it is often hard to prove useful bounds on the running time, and the method has not been successfully applied to generating random graphs given degrees.

There are algorithms faster than the MCMC-based ones, that generate graphs with a weaker approximation of the distribution to the uniform. Steger and Wormald [16] gave an $O(d^2n)$ -time algorithm that generates random d -regular graphs for d up to $n^{1/28}$, where all graphs are generated with asymptotically the same probability. Kim and Vu [11] proved that the same algorithm works to the same extent for all $d \leq n^{1/3-\epsilon}$. Bayati et al. [2] subsequently modified and extended it for the non-regular case, under certain conditions, and generated random d -regular graphs for all $d \leq n^{1/2-\epsilon}$, but with a weaker approximation to uniform (bounding the total variation distance by $o(1)$). Using a different approach, Zhao [19] obtained an $O(dn)$ -time approximate algorithm, which can generate d -regular graphs for $d = o(n^{1/3})$ under weak approximation (bounding the total variation distance). These algorithms are much faster than using MCMC, but the approximation (to the uniform)

is achieved only asymptotically as $n \rightarrow \infty$. Thus, whereas MCMC permits the approximation error to be made arbitrarily small for graphs of a fixed size by running on the chain sufficiently long, the approximation error in [2, 11, 16, 19] depends on n and cannot be improved by more computation.

The main purpose of the present paper is to introduce a new general framework of uniform generation of combinatorial structures. We will apply it in this paper to the special, nevertheless particularly interesting, case of random d -regular graphs on n vertices. The result is an algorithm, *REG*, effective for $d = o(\sqrt{n})$. This significantly improves the bounds on d in [12]. *REG* is an exactly uniform sampler and thus there is no uncontrollable distortion as in [2, 11, 16, 19]. Moreover, its expected running time per graph generated is $O(nd^3)$ (the same as [12]), which remains quite practical, comparing favourably with the quite impractical running time bounds of MCMC samplers.

Theorem 1. *Algorithm REG generates d -regular graphs uniformly at random.*

Theorem 2. *REG can be implemented so that for $1 \leq d = o(\sqrt{n})$, the expected time complexity for generating a graph is $O(nd^3)$.*

The same time complexity seems likely to apply under the weaker assumption that $d = O(\sqrt{n})$, as one might expect comparing with the constraint $d = O(n^{1/3})$ in [12]. However, this is not proved here, the difficulty being that the generation method is different and the analysis is now much more intricate.

In some applications, one might care more for a low time complexity than a perfect uniform sampling. As a byproduct of Theorem 1, by omitting a particular feature of *REG* that requires excessive computation, we will obtain a simpler, linear-time algorithm (that is, linear in the output size, which is the number of edges) called *REG**, which approximately generates a random d -regular graph. *REG** will be defined in Section 10. The total variation distance between the distribution of the output of *REG** and the uniform distribution is bounded as follows.

Theorem 3. *Algorithm REG* randomly generates a d -regular graph whose total variation distance from the uniform distribution is $o(1)$, for any $d = o(\sqrt{n})$. Moreover, the expected number of steps required for generating a graph is $O(dn)$.*

This improves the running time $O(nd^2)$ of [2], and the range $d = o(n^{1/3})$ of [19], while achieving a quality of output distribution similar to both. A similar modification to the algorithm of [12] would achieve the same result, provided that $d = O(n^{1/3})$.

We outline our general framework in Section 3. The framework includes several operations and parameters that will be defined in accordance with the types of combinatorial structures to be generated. For the application in the present paper, this framework is used in each of the three phases of the main algorithm, called *REG*, that is a uniform sampler for d -regular graphs on n vertices. The framework requires operations and parameters to be defined for the various phases of *REG*. These are given in Sections 7.2 and 7.4 and at the beginning of Sections 7.3, 9.1 and 9.2. The full structure of this paper is explained at the end of the following section, after the new ideas have been discussed in relation to the algorithm *DEG* of [12].

We note that several papers have adapted the approach of [12] for generation of other structures (e.g. McKay and Wormald [13]), and also for enumeration (e.g. Greenhill and McKay [7]). Such works have not led to improvements in the result [12] achieves for d -regular graphs. We expect the ideas introduced here will filter out to improved results for several kinds of structures, including graphs with non-regular degree sequences. Such issues will be examined elsewhere.

2 The old and the new

In this section we first summarise the procedure *DEG* used in [12], which provides some of the foundations required for applying our method to regular graphs. We then highlight the new ideas used in *REG*, and give a skeleton description of that algorithm. Finally, we describe the layout of the paper in relation to exposing the new framework and defining and analysing *REG* and *REG**.

For generating random graphs with given degrees, we use the *pairing model*, first introduced in [4], defined as follows. Let $\mathbf{d} = (d_1, \dots, d_n)$ be a degree sequence (thus $\sum_{i=1}^n d_i$ is always assumed to be even). Represent each vertex $i \in [n]$ as a bin v_i . Place d_i distinct points in bin v_i for every $1 \leq i \leq n$. Take a uniformly random perfect matching of the $\sum_i d_i$ points. This perfect matching is called a *pairing*; each pair of points joined in the matching is called a *pair* of the pairing. Note that each pairing P corresponds to a multigraph with degree sequence \mathbf{d} , denoted by $G(P)$, obtained by regarding each pair in the pairing as an edge. Moreover, by a simple counting argument, we see that every simple graph of degree sequence \mathbf{d} corresponds to the same number of pairings. Thus, letting Φ denote the whole set of pairings and $\mathcal{B} \subseteq \Phi$ the set of pairings corresponding to simple graphs, if an algorithm can generate a pairing $P \in \mathcal{B}$ uniformly at random, then $G(P)$ has the uniform distribution over all graphs with degree sequence \mathbf{d} .

Given a pairing P , and two vertices i and j , the set of pairs between i and j in P , if non-empty, is called an *edge*, i.e. edge ij , and the size of the set is the *multiplicity* of ij . If the multiplicity of ij is 1, then ij is a *single edge*; otherwise it is a *multi-edge*. In particular, we say ij is a *double edge*, a *triple edge*, or a *quadruple edge*, if its multiplicity is two, three, or four respectively. An edge ij with $i = j$ is called a *loop* at i .

Outline of *DEG*

The algorithm *DEG* begins by generating a uniformly random pairing $P_0 \in \Phi$. An appropriate set $\mathcal{A} \subseteq \Phi$ is pre-defined, such that pairings in \mathcal{A} have no multi-edges other than double non-loop edges, and have limited numbers of double edges and loops. If $P_0 \notin \mathcal{A}$, then the algorithm terminates. We call this an *initial rejection*. The set \mathcal{A} is chosen in such a way that $\mathbb{P}(P_0 \in \mathcal{A})$ is bounded away from 0. This is possible when $d = O(n^{1/3})$. If no initial rejection occurs, two phases are applied in turn. In the first phase, called *loop reduction*, P_0 is used to obtain a uniformly random pairing $P_1 \in \mathcal{A}$ with no loops but the same number of double edges as P_0 . Then, starting with P_1 , the algorithm enters the *double-edge reduction* phase, and obtains a uniformly random pairing $P_2 \in \mathcal{B}$. (Recall that \mathcal{B} denotes the set of pairings associated with simple graphs.) Thus, in each phase, the number of undesirable structures (which are in turn loops and double edges) is reduced to 0. Termination is also possible during the phases, due to rejection aimed at maintaining uniformity. In that case,

no pairing is outputted by the algorithm. Otherwise, the pairing outputted by the last phase is the output of the algorithm. Naturally, the algorithm is repeated until output occurs.

For exposition purposes, we focus on the second phase, which starts from a pairing $P_1 \in \mathcal{A}$. In particular, P_1 has no loops and at most i_1 double edges, where i_1 is specified in the definition of \mathcal{A} . Then P_1 is distributed uniformly at random (u.a.r.) in $\bigcup_{0 \leq i \leq i_1} \mathcal{S}_i$, where \mathcal{S}_i denotes the set of pairings in \mathcal{A} containing exactly i double edges. (In the present paper, these sets are called *strata*.) We will describe how, with probability bounded away from 0, we can use P_1 to u.a.r. generate a pairing in \mathcal{S}_0 (note that $\mathcal{S}_0 = \mathcal{B}$ in the second phase).

Initially, the algorithm sets $P = P_1$. There is an inductive step which assumes that, conditional on $P \in \mathcal{S}_i$, P occurs uniformly at random in \mathcal{S}_i . The algorithm then randomly performs a certain kind of operation called a *switching*, that produces a pairing $P' \in \mathcal{S}_{i-1}$, and then P is reset equal to P' . (The definition of the switching is perhaps not necessary at this point, but the curious reader may consult Figure 2, Section 7.2.) The general idea is to reject P' with a small probability, which is a function of P and P' , so that P' becomes a uniformly random member of \mathcal{S}_{i-1} if not rejected. This process is iterated until reaching some $P \in \mathcal{S}_0$ which is then outputted as P_2 by *DEG*. By induction, P_2 is uniformly distributed over \mathcal{S}_0 .

We next consider the probability of rejection, which is crucial. Switchings are defined in such a way that the number $f(P)$ of switchings that can be performed on P depends only weakly on anything other than how many double edges P contains. A parameter $m_f(i)$ is specified such that

$$m_f(i) \geq \max_{P \in \mathcal{S}_i} f(P).$$

In discussing the inductive step, we condition on the event $P \in \mathcal{S}_i$. Firstly one of the $f(P)$ switchings is chosen u.a.r., and with probability $f(P)/m_f(i)$ the switching is performed, to obtain $P' \in \mathcal{S}_{i-1}$. Otherwise, with the remaining probability, the algorithm terminates. We call this termination an *f-rejection* (where ‘f’ stands for ‘forward’). Since P is uniformly distributed in \mathcal{S}_i , the probability that $P' \in \mathcal{S}_{i-1}$ is generated at this point by the switching is $|\mathcal{S}_i|^{-1} m_f^{-1} b(P')$, where $b(P')$ is the number of switchings that lead to P' from pairings in \mathcal{S}_i . At this point, the algorithm accepts P' with probability $m_b(i-1)/b(P')$ where $m_b(i-1)$ is a pre-determined parameter satisfying

$$m_b(i-1) \leq \min_{P'' \in \mathcal{S}_{i-1}} b(P'').$$

If P' is not accepted, the algorithm terminates, and this is called a *b-rejection* (where ‘b’ stands for ‘backward’). The probability that a given pairing $P' \in \mathcal{S}_{i-1}$ was produced is now $|\mathcal{S}_i|^{-1} m_f(i)^{-1} m_b(i-1)$, which does not depend on $P' \in \mathcal{S}_{i-1}$. Thus, if it reaches this point, the algorithm has generated a uniformly random member of \mathcal{S}_{i-1} , and the inductive step is finished.

The range of applicable degree sequences for *DEG* is determined by the probability of rejection at some time during the algorithm. In [12], the bound $d = O(n^{1/3})$ for d -regular graphs could not be weakened because the probability of rejection would get too close to 1. This is caused by the probability of f-rejection becoming too large, due to the increasing gap between the typical value of $f(\mathcal{P})$ and its maximum, m_f .

The first phase, loop reduction, is similar, except that of course a different switching is used. There are less loops than double edges in expectation, so the crucial phase to improve, in order to relax the upper bound on d , is the second phase.

New features in *REG*

Our new approach extends that used in *DEG*, introducing some major new features in both the algorithm specification and its analysis, employed specifically in the double-edge reduction phase. For one thing, we narrow the gap between m_f and the average value of $f(P)$ by permitting certain switchings, called class B, that do not have the desired effect on the number of double edges. The total number of permitted switchings is then less dependent on P , and $f(P)$ increases on average. The result is a lower probability of f-rejection. The other new feature of the algorithm, which we call *boosting*, raises the value of m_b by occasionally performing a different type of switching that targets the creation of some otherwise under-represented elements of a set \mathcal{S}_i . This reduces the probability of having a b-rejection. These changes necessitate several associated alterations to the algorithm. The most notable alterations are: (i) the algorithm no longer proceeds through the sets \mathcal{S}_i step-by-step, decreasing i by 1 at each step, though this is still the most common type of step; (ii) unlike in [12], the probability that a pairing is reached in the algorithm (at all, or at a given step) no longer depends only on the set \mathcal{S}_i to which it belongs; (iii) not all switchings to pairings in such a set will be performed with the same probability. As a result of these changes, the analysis of the algorithm is entirely different. In particular, we are forced to relinquish maintaining the property that, at each step, conditional upon the current pairing being within the set \mathcal{S}_i , it is distributed uniformly in that set (except for the case $i = 0$). Instead, we focus on the expected numbers of visits to the states in the associated Markov chain.

The rest of the story

As in [12], a set $\mathcal{A}_\gamma \subseteq \Phi$ is specified such that $\mathbb{P}(\mathcal{A}_\gamma)$ is bounded away from zero. Here, $\gamma > 0$ is a pre-determined constant. It can be altered for better performance of the algorithm for particular n and d using the results of this paper. A uniformly random pairing $P \in \Phi$ is generated, and initial rejection is performed if $P \notin \mathcal{A}_\gamma$. Pairings in \mathcal{A}_γ will in general contain loops, and double and triple non-loop edges, but no other multi-edges. Three phases are performed sequentially, for reduction of loops, then triple edges, and finally double edges.

Since the new features of *REG* will be useful in other contexts, we set up a general framework for the description of a phase in Section 3. This new framework results in a different analysis from [12], which will be given in Section 5, with a glossary provided as Section 4. The proof that each phase ends with a uniformly random object of the required type is rather involved, so an example appropriate to the double-edge reduction phase is given in Section 6. This includes an illustration of how to set some of the parameters of a phase appropriately. Since the only nontrivial phase in *REG* is for double-edge reduction, the definitions of the switchings and other parameters in this phase, and the analysis required for bounding the time complexity, is done in Section 7.

With this out of the way, the basic anatomy of *REG* is completed in Sections 8 and Section 9. In Section 8, we specify the set \mathcal{A}_γ of pairings that do not trigger initial rejection, and bound the probability of an initial rejection. Then, in Section 9, we define the first two phases, for reductions of loops and of triple edges respectively. These two phases are

much simpler than phase 3. The switchings employed in these two phases are the same as used in [12], and are defined in Section 9. The analysis is similar to that in [12], though we now have a higher upper bound on degree. The probability of a rejection occurring in these phases is also bounded.

Finally in Section 10, we prove the main theorem by bounding the expected running time. This is basically determined by the task of computing the probability of rejection, in particular $b(P)$. By ignoring rejections (carrying on regardless), we will obtain the approximate sampler REG^* in Section 10. We prove that REG^* runs in linear time in expectation for generating one random regular graph, and bound the total variation distance between the distribution of the output of REG^* and the uniform distribution.

3 General description of a phase

We present here the definition of a phase that will be common to any application of our approach to reducing the number of occurrences of an undesired configuration (for instance, a double edge), by using repetitions of operations, called switchings here, that can be defined to suit the application.

A phase begins with a set Φ_1 partitioned into sets $\bigcup_{0 \leq i \leq i_1} \mathcal{S}_i$, called *strata*, for some integer $i_1 \geq 0$. For each $P \in \Phi_1$, set $\mathcal{S}(P) = i$ if \mathcal{S}_i contains P . A set of possible operations, called switchings, is specified. Each switching converts some $P \in \Phi_1$ to another element of Φ_1 . Each switching has both a *type* and a *class*. The set of possible types is denoted \mathcal{T} , and the set of possible classes is denoted \mathcal{C} . The phase begins with a random element $P \in \Phi_1$ with a known distribution π_0 , and either outputs a uniformly random element of \mathcal{S}_0 , or terminates with no output (rejection). The parameters specified for the phase are $\rho_\tau(i)$, $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$, for each switching type τ and class α , and each $i \leq i_1$.

The parameters $\rho_\tau(i)$ satisfy $\sum_\tau \rho_\tau(i) \leq 1$ for each i . If an element $P \in \mathcal{S}_i$ ($i > 0$) arises during the phase, a switching type τ is chosen with probability $\rho_\tau(i)$. If $\sum_\tau \rho_\tau(i) < 1$, a rejection will occur with the remaining probability; we call this a t-rejection (where ‘t’ stands for ‘type’).

Given a switching type τ and an element P , the number of switchings of type τ that can be applied to P is denoted by $f_\tau(P)$. The parameters $\overline{m}_\tau(i)$ satisfy

$$\overline{m}_\tau(i) \geq \max_{P \in \mathcal{S}_i} f_\tau(P).$$

Similarly, for a switching class α , the number of switchings of class α that can be applied to other elements to produce P is denoted by $b_\alpha(P)$. The parameters $\underline{m}_\alpha(i)$ satisfy

$$\underline{m}_\alpha(i) \leq \min_{P \in \mathcal{S}_i} b_\alpha(P).$$

The phase consists of repetitions of a *switching step*, specified as follows.

Given $P \in \mathcal{S}_i$,

- (i) If $i = 0$, output P .

- (ii) Choose a type: choose τ with probability $\rho_\tau(i)$, and with the remaining probability, $1 - \sum_\tau \rho_\tau(i)$, perform a t-rejection. Then select u.a.r. one of the type τ switchings that can be performed on P .
- (iii) Let P' be the element that the selected switching would produce if applied to P , let α be the class of the selected switching and let $i' = \mathcal{S}(P')$. Perform an f-rejection with probability $1 - f_\tau(P)/\overline{m}_\tau(i)$ and then perform a b-rejection with probability $1 - \underline{m}_\alpha(i')/b_\alpha(P')$;
- (iv) if no rejection occurred, replace P with P' .

The switching step is repeated until the phase terminates, which happens whenever an element $P \in \mathcal{S}_0$ is reached or a rejection occurs.

Note that in each switching step only a switching type is selected, not a switching class. Only after a switching is chosen in (ii) is the class of the switching determined.

To complete the definition of a phase, it is sufficient to specify Φ_1, i_1 , the sets \mathcal{S}_i , the set of switchings and their types and classes, and the numbers $\rho_\tau(i)$, $\overline{m}_\tau(i)$ and $\underline{m}_\tau(i)$. These parameters will be carefully chosen in such a way that the expected number of times that a given element in \mathcal{S}_i is visited during the phase depends only on i . Given this, and the fact that termination of the phase occurs as soon as \mathcal{S}_0 is reached, it follows that the element outputted is distributed u.a.r. from \mathcal{S}_0 . Subject to this, in choosing the parameters we also aim to keep the probability of rejection small.

4 Glossary

For the benefit of the notation-weary reader, we list some terms already defined, and some soon to be defined.

For any element P :

$\sigma(P)$ is the expected number of times the algorithm passes through P (defined in Page 10).

$f_\tau(P)$ is number of ways that a switching of type τ can be performed on P (defined in Page 7).

$b_\alpha(P)$ is number of switchings of class α that produce P (defined in Page 7).

P_t is the element obtained after step t , if no rejection has occurred.

$q_{P,P'}$ is the transition probability from P to P' .

$\mathcal{S}_{\tau,\alpha}^-(P)$: the set of elements that can produce P using a type τ , class α switching (defined in Page 24).

For $P \in \mathcal{S}_i$:

$\rho_\tau(i)$ is probability of choosing switching type τ to apply to $P \in \mathcal{S}_i$ (first appearing in Page 7).

$\sigma(i)$ is such that $\sigma(P) = \sigma(i)$ for all $P \in \mathcal{S}_i$ under constraints to be enforced (first appearing in (7)).

$\mathcal{S}(P)$ is another name for i . (That is, the index of the stratum containing P .)

Special states in the Markov chain:

R : a state denoting that rejection has occurred.

F : each element in \mathcal{S}_0 has transition probability 1 to F .

In a phase:

Φ_1 : the set of all possible elements arising in the phase.

i_1 : the maximum integer i such that $\mathcal{S}_i \in \Phi_1$.

\mathcal{T} : the set of switching types.

\mathcal{C} : the set of switching classes.

$\overline{m}_\tau(i)$: pre-determined parameters satisfying $f_\tau(P) \leq \overline{m}_\tau(i)$ for every $P \in \mathcal{S}_i$ (first appearing in Page 7).

$\underline{m}_\alpha(i)$: pre-determined parameters satisfying $b_\alpha(P) \geq \underline{m}_\alpha(i)$ for every $P \in \mathcal{S}_i$ (first appearing in Page 7).

5 General analysis of a phase

In this section we lay the groundwork for specification of the predefined parameters described in Section 3, in such a way that the algorithm performs the desired uniform sampling. Consider a phase with specified switchings and parameters \mathcal{S}_i , $\rho_\tau(i)$, $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$ for all appropriate i and τ , as well as a specified value of i_1 . Assume that

$$\rho_\tau(i) \geq 0 \quad \text{for all } 0 \leq i \leq i_1 \text{ and all } \tau \in \mathcal{T}; \quad (1)$$

$$\sum_{\tau \in \mathcal{T}} \rho_\tau(i) \leq 1 \quad \text{for all } 0 \leq i \leq i_1; \quad (2)$$

$$f_\tau(P) \leq \overline{m}_\tau(i) \quad \text{for all } 0 \leq i \leq i_1, P \in \mathcal{S}_i \text{ and for each } \tau \in \mathcal{T}; \quad (3)$$

$$b_\alpha(P) \geq \underline{m}_\alpha(i) \quad \text{for all } 0 \leq i \leq i_1, P \in \mathcal{S}_i \text{ and for each } \alpha \in \mathcal{C}. \quad (4)$$

Recall that $\Phi_1 = \bigcup_{i=0}^{i_1} \mathcal{S}_i$. The parameters determine a Markov chain, denoted by \mathcal{M} , on states $\Phi_1 \cup \{R, F\}$, where R and F are two artificially introduced absorbing states. The chain moves from an element P directly to R if rejection occurs in a switching step from P to P' (rather than to P' being considered at the time of rejection), and it moves to F with probability 1 in the next step after reaching any element in \mathcal{S}_0 . We refer to the state at step t as P_t , permitting $P_t = R$ or F .

We make the following assumption about \mathcal{M} :

(A1) all states in Φ_1 are transient in \mathcal{M} .

Let Q be the matrix of transition probabilities between all states of \mathcal{M} in Φ_1 . Thus $Q = (q_{P,P'})$ where $q_{P,P'}$ is the transition probability from P to P' .

The transition probability $q_{P,P'}$ can be computed as follows. Assume $P \in \mathcal{S}_i$ and that S is a type τ , class α switching that converts P into $P' \in \mathcal{S}_j$. Condition on state P being reached in \mathcal{M} . Then, from part (ii) of the switching step, the probability that S is chosen equals $\rho_\tau(i)/f_\tau(P)$. Hence, from part (iii), the probability S is performed and neither t- nor f-rejection occurs is $\rho_\tau(i)/\overline{m}_\tau(i)$. On the other hand, the probability that b-rejection does not occur is $\underline{m}_\alpha(j)/b_\alpha(P')$. Hence,

$$q_{P,P'} = \sum_{(\tau,\alpha)} N_{\tau,\alpha}(P, P') \frac{\rho_\tau(i) \underline{m}_\alpha(j)}{\overline{m}_\tau(i) b_\alpha(P')}, \quad (5)$$

where $N_{\tau,\alpha}(P, P')$ is the number of switchings of type τ and class α that convert P to P' .

By assumption (A1), Q is the submatrix of the transition matrix that refers to the transient states. Hence, the matrix $(I - Q)^{-1}$ exists. Indeed, this matrix is known as the fundamental matrix of \mathcal{M} , and it is clearly equal to $I + Q + Q^2 + \dots$. (An easy argument shows that this series is convergent because these states are transient.) Moreover, the (P, P') entry of $(I - Q)^{-1}$ is clearly the expected number of visits to state P' given that the chain starts in state P (where being in the initial state is counted as a visit). Given a (row) vector $\vec{\pi}_0$ for the initial distribution of the transient states, define $\sigma(P)$ to be the expected number of times that P is visited in \mathcal{M} . Then the vector $\vec{\sigma}$, composed of the values $\sigma(P)$, is given by

$$\vec{\pi}_0(I - Q)^{-1} = \vec{\sigma}. \quad (6)$$

A key feature of our approach is to specify $\rho_\tau(i)$ in such a way that $\sigma(P)$ depends only on $\mathcal{S}(P)$, i.e., there are fixed numbers $\sigma(i)$ ($0 \leq i \leq i_1$) such that

$$\text{for all } i, \text{ and all } P \in \mathcal{S}_i, \quad \sigma(P) = \sigma(i). \quad (7)$$

To aid in finding such ρ_τ easily, we require that, for a given switching S , the expected number of switching steps during the phase in which S is chosen in (ii) for which f-rejection does not occur in (iii) is some number $q_\alpha(j)$ depending only on the class α of S and the set \mathcal{S}_j containing the element it creates. Considering the derivation of (5), this is equivalent to requiring that, for all j and α ,

$$\frac{\sigma(i)\rho_\tau(i)}{\bar{m}_\tau(i)} = q_\alpha(j) \text{ for all } (i, j, \tau, \alpha) \in \mathcal{S} \quad (8)$$

where \mathcal{S} is the set of all (i, j, τ, α) for which there exists a switching of type τ and class α taking an element in \mathcal{S}_i to an element in \mathcal{S}_j .

Rewrite (6) as $\vec{\sigma} = \vec{\pi}_0 + \vec{\sigma}Q$, and note from (5) that the component of $\vec{\sigma}Q$ referring to (i.e. indexed by) $P' \in \mathcal{S}_j$ is

$$\sum_P \sigma(P)q_{P,P'} = \sum_{(\tau,\alpha,P)} \sigma(P)N_{\tau,\alpha}(P, P') \frac{\rho_\tau(\mathcal{S}(P)) \underline{m}_\alpha(j)}{\bar{m}_\tau(\mathcal{S}(P)) b_\alpha(P')},$$

where the sum is over all τ, α and P for which there is at least one type τ , class α switching that converts P into P' . By (7) and (8), this summation is

$$\sum_{(\tau,\alpha,P)} N_{\tau,\alpha}(P, P') \frac{q_\alpha(j) \underline{m}_\alpha(j)}{b_\alpha(P')} = \sum_\alpha q_\alpha(j) \underline{m}_\alpha(j)$$

since for each $\alpha \in \mathcal{C}$, the number of class α switchings that converts some element P to P' is $\sum_{\tau,P} N_{\tau,\alpha}(P, P')$, which is $b_\alpha(P')$ by definition. Thus, provided that (7) and (8) hold, (6) is equivalent to

$$\sigma(P) = \vec{\pi}_0(P) + \sum_{\alpha \in \mathcal{C}} q_\alpha(j) \underline{m}_\alpha(j) \text{ for all } j \text{ and } P \in \mathcal{S}_j. \quad (9)$$

Noting that (6) determines $\vec{\sigma}$, we have proved the following lemma.

Lemma 4. Suppose that for given numbers $\rho_\tau(i)$, $\underline{m}_\alpha(j)$ and $\overline{m}_\tau(j)$ ($i, j \in [0, i_1]$, $\tau \in \mathcal{T}$, $\alpha \in \mathcal{C}$) satisfying the conditions (1)–(4), there is a simultaneous solution $(\sigma(i), q_\alpha(i))_{0 \leq i \leq i_1}$ to equations (8) and (9). Then, for each $i \in [0, i_1]$, the expected number of visits to any given element in \mathcal{S}_i is $\sigma(i)$.

Henceforth in this paper, we assume that the initial distribution is uniform over Φ_1 , that is, $\pi_0(j) = |\Phi_1|^{-1}$ for all $j \in \Phi_1$.

Remark. Consider the case that there is exactly one type τ and one class α of switchings involved in a phase, and each switching converts an element in \mathcal{S}_{i+1} to another element in \mathcal{S}_i for some $i \geq 0$. Then we may simply set $\rho_\tau(i) = 1$ for every $0 \leq i \leq i_1$. Clearly (A1) is satisfied, as there is no cycling in the Markov chain, and (8) and (9) combine to give

$$\sigma(j) = \frac{1}{|\Phi_1|} + \frac{\sigma(j+1)}{\overline{m}_\tau(j+1)} \cdot \underline{m}_\alpha(j).$$

Uniformity is guaranteed as $\sigma(j)$ can be inductively computed from $\sigma(j+1)$. This inductive approach is exactly the essence of *DEG* described in Section 2, and hence the method in [12] is a special case of our general method, obtained by setting $\rho_\tau(i) = 1$. It is the possibility of using different types and classes of switchings, the flexibility of setting non-trivial values to $\rho_\tau(i)$, and the much more flexible choice of Markov chains permitting cycling, that provides the power of the approach in the present paper.

6 An example: calculating ρ and proving uniformity

The parameters $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$ will be specified, depending on the particular application, such that (3) and (4) are satisfied. The tightness of these bounds on $\max f_\tau(P)$ and $\min b_\alpha(P)$ effectively influence the efficiency of the phase, as tighter bounds yield smaller rejection probabilities in substep (iii) of a switching step. However, a major task of the design of the phase is to set $\rho_\tau(i)$ properly to ensure (8), as well as to minimize the probability of a t-rejection. We achieve this by deducing a system of equations and inequalities that the parameters $\rho_\tau(i)$ and the variables $\sigma(j)$ must satisfy. Then, we find a desirable solution to the system, bearing in mind the rejection probabilities, and set the value of $\rho_\tau(i)$ accordingly. We illustrate this by considering a particular example, developing the analysis in the previous section, which we can use later since it applies to phase 3 of *REG*, where double edges are eliminated.

We assume that $\Phi_1 = \cup_{i=0}^{i_1} \mathcal{S}_i$ and all strata have been specified, as well as parameters $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$ satisfying (3) and (4). We assume that $\mathcal{T} = \{I, II\}$ and $\mathcal{C} = \{A, B\}$ and there will be three kinds of switchings in the phase: IA (type I, class A), converting an element in \mathcal{S}_{j+1} to an element in \mathcal{S}_j ; IB (type I, class B), maintaining in the same stratum; and IIB (type II, class B), converting from \mathcal{S}_{j-1} to \mathcal{S}_j . See Figure 1 for an illustration of the possible transitions into \mathcal{S}_j . If j is such that \mathcal{S}_{j-1} or \mathcal{S}_{j+1} does not exist in Φ_1 , the corresponding stratum is omitted from the diagram. Additionally, there is no IIB switching from \mathcal{S}_0 to \mathcal{S}_1 , and the Markov chain always transits from an element in \mathcal{S}_0 to F in the next step.

We have no need to define \mathcal{S}_j here, nor the switchings involved. What they are in the case of phase 3 of *REG* will be revealed in Section 7.

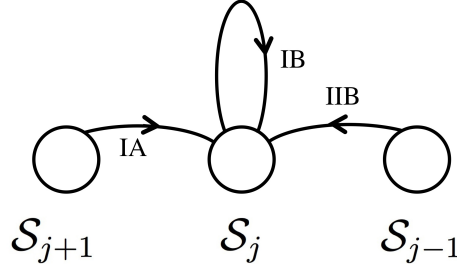


Figure 1: *Transitions into \mathcal{S}_j*

From Figure 1, we see that (8) with $\alpha = B$ in the two cases $i = j$ and $i = j - 1$ implies that the expected number of times that a given Class B switching produces $P \in \mathcal{S}_j$ (including the time it is b-rejected, in part (iii) of the switching step, if that occurs) is

$$q_B(j) = \frac{\sigma(j)\rho_I(j)}{\bar{m}_I(j)} = \frac{\sigma(j-1)\rho_{II}(j-1)}{\bar{m}_{II}(j-1)} \quad (1 \leq j \leq i_1), \quad (10)$$

and at times we will use either of these quantities in place of $q_B(j)$. For similar reasons,

$$q_A(j) = \sigma(j+1)\rho_I(j+1)/\bar{m}_I(j+1) \quad (0 \leq j \leq i_1 - 1). \quad (11)$$

It is convenient to set

$$x_j = \sigma(j)|\Phi_1|, \quad (12)$$

whence, recalling that $\bar{\pi}_0(j) = |\Phi_1|^{-1}$, substitution of (10) and (11) into (9) gives

$$x_j = 1 + \frac{x_{j+1}\rho_I(j+1)}{\bar{m}_I(j+1)} \cdot \underline{m}_A(j) + \frac{x_j\rho_I(j)}{\bar{m}_I(j)} \cdot \underline{m}_B(j) \quad (1 \leq j \leq i_1 - 1). \quad (13)$$

We need separate equations for $j \in \{0, i_1\}$ as (10) does not hold for $j = 0$ and (11) does not hold for $j = i_1$. No $P \in \mathcal{S}_0$ can be reached by any class B switching, because whenever any element in \mathcal{S}_0 is reached, the Markov chain proceeds to state F in the very next step. Therefore, $q_B(0) = 0$. Similarly, no element $P \in \mathcal{S}_{i_1}$ can be reached via a type A switching because Φ_1 does not contain \mathcal{S}_{i_1+1} . Therefore, $q_A(i_1) = 0$. So for $j = i_1$ or 0 we have in place of (13) provided $i_1 > 0$

$$x_{i_1} = 1 + \frac{x_{i_1}\rho_I(i_1)}{\bar{m}_I(i_1)} \cdot \underline{m}_B(i_1), \quad x_0 = 1 + \frac{x_1\rho_I(1)}{\bar{m}_I(1)} \cdot \underline{m}_A(0). \quad (14)$$

Moreover, the second equality in (10) implies

$$\rho_{II}(j) = \rho_I(j+1) \frac{x_{j+1}}{x_j} \cdot \frac{\bar{m}_{II}(j)}{\bar{m}_I(j+1)} \quad \text{for all } 0 \leq j \leq i_1 - 1. \quad (15)$$

As boundary cases, we require

$$\rho_I(0) = \rho_{II}(0) = 0, \quad \rho_{II}(i_1) = 0, \quad (16)$$

where the first two equalities are required because every element in \mathcal{S}_0 is forced to transit to F once it is reached, and the last because Φ_1 does not contain \mathcal{S}_{i_1+1} . Equations (13)–(16) determine a system that will be required to have a solution. In general, the system will be underdetermined, and this freedom can be used for the convenience of specifying some values of the ρ_τ variables, to have values suitable for our purpose (in which we aim to keep the probability of rejection to a minimum), and then solving for the remaining variables. However, we will also require that

$$\rho_I(i) + \rho_{II}(i) \leq 1, \quad \rho_I(i), \rho_{II}(i) \geq 0 \quad \text{for all } 0 \leq i \leq i_1. \quad (17)$$

This condition ensures that $\rho_\tau(i)$ satisfy (1) and (2), as required if they are to be used as probabilities. Naturally, it needs to be checked in any particular case that a solution of the desired type exists.

For any solution $(\rho_\tau^*(j), x_j^*)$ of the system (13)–(17), we may set $\rho_\tau(j)$ equal to $\rho_\tau^*(j)$ for every $0 \leq j \leq i_1$ and each $\tau \in \{I, II\}$. Then $(\sigma(j), q_\alpha(j))_{0 \leq j \leq i_1}$ for $\alpha \in \{A, B\}$ can be computed using x_j^* and (12), (10) and (11). Thus, $(\sigma(j), q_\alpha(j))_{0 \leq j \leq i_1}$ is a solution to equations (8) and (9), and so by Lemma 4, the expected number of visits to any given element in any stratum \mathcal{S}_i is $\sigma(i)$. Recall that the phase finishes as soon as an element of \mathcal{S}_0 is reached. Hence, every element in \mathcal{S}_0 is reached at most once, and the probability that a given $P \in \mathcal{S}_0$ is reached is equal to $\sigma(0)$, the same for all $P \in \mathcal{S}_0$. Thus, we have proved the following.

Lemma 5. *Assume that $(\rho_\tau^*(i), x_i^*)$ is a solution to system (13)–(17). Set $\rho_\tau(i)$ to be $\rho_\tau^*(i)$ for every $1 \leq i \leq i_1$ as the type probabilities in the phase. Assume that no rejection occurs during the algorithm. Then the last element visited in the phase is distributed uniformly at random in \mathcal{S}_0 .*

7 Phase 3: double edge reduction

We now turn to giving the explicit construction of the algorithm *REG*, and its analysis. We will analyse the three phases of the algorithm, essentially applying the general framework in Section 5 to each phase of *REG*. We begin with Phase 3, which is the most interesting phase and has been partially described in Section 6. It was this phase of double-edge reduction that determined the range of d for which *DEG* runs efficiently in [12]. To improve the range of d in [12] we need to allow more flexible Markov chains, which motivates the new approach in Section 5. For phase 3 of *REG*, there will be two types (I and II) of switchings involved, categorized into two classes A and B. The transitions between $(\mathcal{S}_i)_{0 \leq i \leq i_1}$ caused by performing these switchings are exactly as described in Section 6, which lead to system (13)–(17). To complete the specification of phase 3, we will specify Φ_1 , \mathcal{S}_i and i_1 , and the set of switchings for this phase. We will analyse these switchings to obtain appropriate parameters $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$ as required for the system (13)–(17). After that, we will specify values for the variables $\rho_I(i)$ and show that, given these, the whole system has a solution. Then, in Section 7.5, we bound the probability that phase 3, once begun, terminates in a rejection.

7.1 Specifying Φ_1 , \mathcal{S}_i and i_1

As described before, if no initial rejection happens, then the algorithm enters phases that sequentially reduces the numbers of loops, triple edges and double edges to zero. Uniformity of the distribution of the output of each phase is guaranteed. Let $\gamma > 0$ be a pre-specified constant. With the foresight of the definition of \mathcal{A}_γ in Section 8, and Lemma 14, define

$$i_1 = \left\lfloor (1 + \gamma) \frac{(d-1)^2}{4} \right\rfloor. \quad (18)$$

For the analysis here, we may assume that the algorithm enters phase 3 with a pairing P_0 uniformly distributed in Φ_1 defined by

$$\Phi_1 = \{P \in \Phi : P \text{ contains at most } i_1 \text{ double edges, and no other multi-edges or loops}\}.$$

We will verify this assumption in Lemma 19 in Section 9 below. Define \mathcal{S}_i to be the set of pairings in Φ_1 containing exactly i double edges. So $\Phi_1 = \cup_{0 \leq i \leq i_1} \mathcal{S}_i$.

7.2 Definition of the switchings

As indicated in Section 6, there are two types of switchings, I and II, in phase 3. The two types are different versions of the same basic switching operation, which we call a *d-switching*, defined as follows to act upon a pairing P that contains at least one double edge. As in Figure 2, pick a double edge with parallel pairs $\{1, 2\}$ and $\{3, 4\}$, with points 1 and 3 in the same vertex u_1 and points 2 and 4 in the same vertex v_1 . Also pick another two pairs $\{5, 6\}$ and $\{7, 8\}$, that represent single edges (recalling that we have assumed Φ contains no pairings with loops), and let u_2, v_2, u_3 and v_3 denote the vertices containing the points 5, 6, 7 and 8 respectively. If all six vertices $u_i, v_i, 1 \leq i \leq 3$, are distinct, then a d-switching replaces the pairs $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$ and $\{7, 8\}$ by $\{1, 5\}$, $\{3, 7\}$, $\{2, 6\}$ and $\{4, 8\}$, producing the situation in the right side of Figure 2.

We emphasise that the specification of the labels in the above and later switching definitions is significant, in that when we come to count the ways that a switching can be applied to a pairing, each distinct valid way of assigning the labels 1, 2, \dots and u_1, v_2, u_2, \dots induces a different switching.

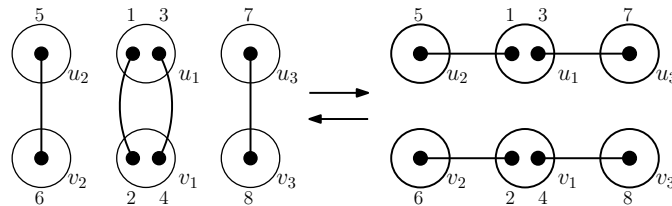


Figure 2: *A type I switching*

Type I switching. If the d-switching operation does not create more than one new double edge, it is a valid type I switching. The *class* of a type I switching depends on how many new double edges it creates. If none, it is in class A, whilst if there is one, then it is in class

B (as in Figure 3). Note that the class of a switching is not chosen; when a random type I switching is chosen in part (ii) of a switching step, the class is determined after the switching is chosen. For a type I class B switching, there was an existing pair between u_1 and u_i or between v_1 and v_i ($i = 2, 3$) before the d-switching was applied. For purposes of counting type I class B switchings, the existing pair is then labelled $\{9, 10\}$, where 9 is in u_1 (or v_1 as the case may be). Of course, this pair remains after the switching is applied. See Figure 3 for an example.

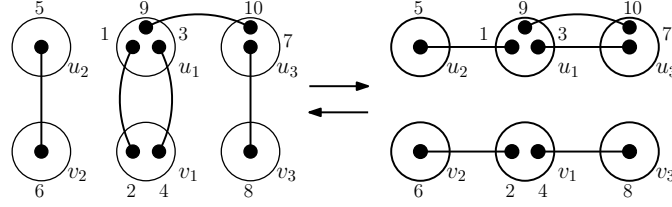


Figure 3: A type I switching of class B

Type II switching. If exactly two new double edges, both incident with u_1 or both incident with v_1 , are created by performing a d-switching, then it is a valid type II switching. For counting purposes, the existing pairs that become part of double edges are labelled $\{9, 10\}$ and $\{11, 12\}$, with 9 and 12 both in u_1 or both in v_1 . (See Figure 4.) A type II switching is always in class B. Note that the choice of placing point 10 in u_2 or in u_3 , say, leads to different type II switchings arising from the same d-switching. See Figures 4 and 5 for an example.

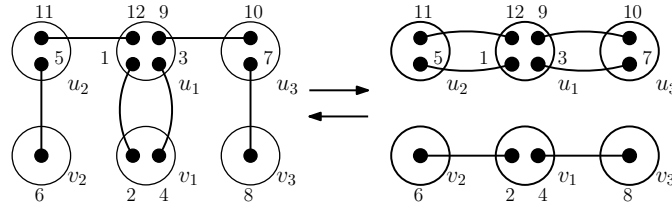


Figure 4: A type II switching

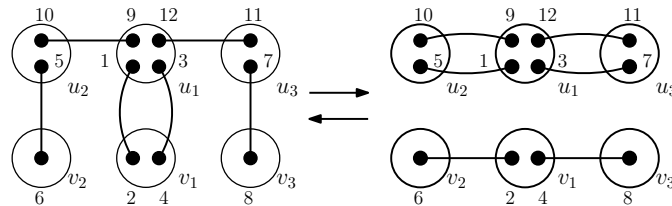


Figure 5: A different type II switching

We close this subsection by elaborating on the need for the various types and classes of switchings. The type I class A switching is the same switching used in [14] to decrease the number of double edges by one. There, choices of the d-switching are not allowed if

u is adjacent to u_1 or u_2 or if v is adjacent to v_1 or v_2 . This makes it hard to bound the f-rejection probability away from 1 unless $d = O(n^{1/3})$. In order to permit d to be larger, we permit the d-switching if at most one new double edge is created. These are the type I class B switchings. However, permitting these switchings causes a problem. Each one creates a 2-path containing exactly one double edge. Among all pairings in a stratum \mathcal{S}_i , the number of such 2-paths can vary immensely, which could lead to a large b-rejection probability. For instance, consider the extreme case where d is even, $i = (d/2)(d/2 + 1)$ and $P \in \mathcal{S}_i$ is a pairing such that all the i double edges in P are contained in an isolated clique of order $d/2 + 1$. Then the number of ways to reach P via a type I, class B switching is 0. If there were no class B switchings of any other types, this would force $\underline{m}_\alpha(i)$ to equal 0, and hence in part (iii) of the general switching step, all class B switchings would be given b-rejection, rendering them useless. In order to repair this situation, we introduce the type II switching (of class B), which boosts the probability of such pairings being reached.

7.3 Specifying $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$

For any positive integer k , we define $M_k = n[d]_k$, where $[x]_k$ denotes the falling factorial, $x(x-1)\cdots(x-k+1)$. Define

$$\overline{m}_I(i) = 4i(M_1 - 4i)^2 \quad (19)$$

$$\overline{m}_{II}(i) = 16i(d-1)^2(d-2)(d-3) \quad (20)$$

$$\underline{m}_A(i) = M_2^2 - M_2(d-1)(16i + 3d^2 + d + 6) \quad (21)$$

$$\underline{m}_B(i) = 16i(d-2)M_2 - 16id(8id + 9d^2 + 3d^3). \quad (22)$$

The next lemma shows that these provide the required bounds on $f_\tau(P)$ and $b_\alpha(i)$, as given by (3) and (4).

Lemma 6. *In phase 3, for every $P \in \mathcal{S}_i$,*

$$(i) \ f_\tau(P) \leq \overline{m}_\tau(i) \text{ for every } 1 \leq i \leq i_1 \text{ and } \tau \in \{I, II\};$$

$$(ii) \ b_\alpha(i) \geq \underline{m}_\alpha(i) \text{ for every } 0 \leq i \leq i_1 \text{ and } \alpha \in \{A, B\} \text{ such that } (\alpha, i) \neq (A, i_1).$$

Proof. Let $P \in \mathcal{S}_i$. Then P has i double edges. For an upper bound on $f_\tau(P)$, there are $4i$ ways to choose a double edge and label its points 1, 2, 3 and 4. The number of choices for each of u_2v_2 and u_3v_3 is at most $M_1 - 4i$, since their end points cannot be in any of the i double edges. Thus $f_I(P) \leq \overline{m}_I(i)$. For the type II switching, we may assume by symmetry that u_1 , rather than v_1 , contains points 9 and 12, and that the point 10 lies in vertex u_3 rather than u_2 . These give four choices. There are again $4i$ ways to choose 1, 2, 3 and 4. Then — recalling that each vertex contains precisely d points — there are at most $(d-2)(d-3)$ ways to choose the two points 9 and 12 in u_1 as points 1 and 3 are excluded from the options, at most $d-1$ ways to choose point 5 in u_2 and at most $d-1$ ways to choose point 7 in u_3 . Hence, $f_{II}(P) \leq 4 \cdot 4i(d-2)(d-3)(d-1)^2 = \overline{m}_{II}(i)$. This shows (i).

For (ii), consider $P \in \mathcal{S}_i$, where $0 \leq i \leq i_1 - 1$. Then $b_A(P)$ is the number of class A switchings that produce P . To determine such a switching, we may choose two 2-paths,

$\{5,1\}$, $\{3,7\}$ and $\{6,2\}$, $\{4,8\}$, as in the right hand side of Figure 2. The number of these arising (with these labels) from the specification of valid class A switchings is $M_2^2 - X$, where M_2 is clearly the total number of 2-paths, and X counts the choices such that at least one of the following occurs:

- (a) either of the 2-paths has a pair contained in a double edge;
- (b) both of the 2-paths consist of only pairs representing single edges, and they share at least one vertex;
- (c) all six vertices u_i, v_i are distinct, and there is an edge between u_i and v_i for some $i \in \{1, 2, 3\}$.

We only need to bound X from above. For (a), by symmetry we consider the case that u_1u_3 is a double edge (see the right side of Figure 3) and we multiply the bound of the number of such choices by four. There are $4i$ ways to label point 3 contained in a double edge. Then, there are $d - 1$ ways to choose another point in u_1 . That bounds the number of 2-paths containing a pair which is part of a double edge. The number of ways to choose the other 2-path is M_2 . This gives the bound $4 \cdot 4i(d - 1)M_2 = 16i(d - 1)M_2$ for (a).

For (b), all u_i are distinct and so are all v_j . So, there are 9 choices for i and j for the common vertex $u_i = v_j$. Consider the case $u_2 = v_2$. There are at most M_2 ways to choose points 5, 1, 3 and 7. There are d ways to choose point 6 (note that point 6 does not need to be distinct from point 5 as this is a bad case counted in M_2^2 as well) and then $d - 1$ ways to choose point 4. This gives a bound $M_2d(d - 1)$. It is easy to see that the same bound holds for each of the other eight cases. So the number of choices for (b) is at most $9M_2d(d - 1)$.

For (c), first consider the case that u_2v_2 is an edge. There are at most M_2 ways to choose points 5, 1, 3, 7. Since u_2v_2 is an edge, there are at most $(d - 1)^2$ ways to choose point 6 and then $d - 1$ ways to choose point 4. This gives at most $M_2(d - 1)^3$ choices for the case that u_2v_2 is an edge. By symmetry, the same bound holds for the case that u_3 is adjacent to v_3 . For the case $u_1 = v_1$, we have at most $(d - 2)(d - 1)$ ways to choose point 2 and then at most $d - 2$ ways to choose point 4, resulting in a bound $M_2(d - 1)(d - 2)^2$. Hence, the number of choices for (c) is at most $M_2(2(d - 1)^3 + (d - 1)(d - 2)^2)$. Adding the three cases, we get $X \leq M_2(d - 1)(16i + 3d^2 + d + 6)$, and so $b_A(P) \geq M_2^2 - X \geq \underline{m}_A(i)$.

Finally, to bound $b_B(P)$ from below, we estimate the number of class B switchings producing P . Note that these can be of either type, I or II. Specifying such a switching involves labelling six vertices and 10 or 12 points, depending on the type of switching, as described in the right hand sides of Figures 3 and 4. By symmetry, we only discuss the case that point 9 is in u_1 and point 10 is in u_3 , and multiply the resulting bound by 4. Choose a double edge z and choose a point in z and label it 3 (in $4i$ ways). This determines the points 7, 9 and 10. Choose another point in u_1 and label it 1 (in $d - 2$ ways, as points 3 and 9 are already specified in u_1). The mate of 1 is labelled 5. If $\{1, 5\}$ represents a single edge, this corresponds to a type I switching. Otherwise, it is in a double edge, and this corresponds to a type II switching (with points 11 and 12 uniquely determined). In each case, another 2-path can be chosen in at most M_2 ways. In all, the number of ways to validly choose all these points is $4i(d - 2)M_2 - Y$ where Y accounts for choices such that at least one of

- (a') the second 2-path also contains a double edge

(b') the two 2-paths share at least one vertex;

or (c) above holds. Similar to the previous argument, the number of choices satisfying (a') is at most $2 \cdot 4id \cdot 4id$, where factor 2 accounts for whether pair $\{6,2\}$ or $\{4,8\}$ is contained in a double edge. The number satisfying (b') is at most $9 \cdot 4id \cdot d^2$, and for (c) it is at most $3 \cdot 4id \cdot d^3$. Hence, $Y \leq 32i^2d^2 + 36id^3 + 12id^4$ and so (recalling the factor 4 from the start)

$$b_B(P) \geq 4(4i(d-2)M_2 - Y) = \underline{m}_B(i). \quad \blacksquare$$

7.4 Fixing $\rho_\tau(i)$

We assume that $d \geq 2$ in the rest of the paper as the case $d = 1$ is trivial: in that case a random pairing immediately gives a random simple 1-regular graph.

With the definition of the three kinds of switchings (IA, IB and IIB) in Section 7.2, the transitions among strata \mathcal{S}_j in the Markov chain is exactly as in the example in Section 6. Recalling the description in Section 6, in order to define $\rho_\tau(i)$, it is sufficient to specify a solution $(\rho_\tau^*(i), x_i^*)$ of the system (13)–(17). We have some latitude here because the system is underdetermined. To keep the probability of t-rejection small in every step, it is desirable to find a solution such that $1 - (\rho_I^*(i) + \rho_{II}^*(i))$ is as small as possible. We prove that the system has a unique solution such that $\rho_I^*(i) = 1 - \epsilon$ for all $1 \leq i \leq i_1$, for some $0 < \epsilon < 1$, at least for n sufficiently large.

Lemma 7. *Let $0 < \gamma < 1$ be fixed, which determines i_1 via (18), and $\epsilon > 0$. Assume that each of the following holds:*

$$\frac{4(1+\gamma)(d-1)^2(d-2)(d-3)}{dn(dn-7d^2+7d-10-4\gamma(d-1)^2)} \leq \epsilon < 1; \quad (23)$$

$$dn-7d^2+7d-10-4\gamma(d-1)^2 > 0; \quad (24)$$

$$(d-2)(d-1)n - (2(1+\gamma)d(d-1)^2 + 9d^2 + 3d^3) > 0. \quad (25)$$

Then, with the values of $\overline{m}_\tau(i)$ and $\underline{m}_\alpha(i)$ defined in Section 7.3, the system (13)–(17) has a unique solution $(\rho_\tau^(i), x_i^*)$ satisfying $\rho_I^*(i) = 1 - \epsilon > 0$ for every $1 \leq i \leq i_1$. For fixed $\gamma > 0$, and n sufficiently large, there is an ϵ satisfying (23) with $\epsilon = O(d^2/n^2)$.*

Proof. For $1 \leq i \leq i_1$ we set $\rho_I^*(i) = 1 - \epsilon$, which is positive by (23), and necessarily set $\rho_I^*(0) = 0$ and $\rho_{II}^*(i_1) = 0$ by (16). Then (14) determines $x_{i_1}^*$. Then for every $0 \leq i \leq i_1 - 1$, the value of x_i^* is determined recursively, in decreasing order of the value of i , by (13) and (14). Equation (25) ensures that $\underline{m}_B(i) > 0$. Moreover, $\underline{m}_B(i) < \overline{m}_I(i)$ for every i , which we will justify below. It follows then that $x_i^* > 0$ for each i . From here, each $\rho_{II}^*(i)$ for $0 \leq i \leq i_1 - 1$ is determined using (15), and is moreover strictly positive for $1 \leq i \leq i_1 - 1$ because $x_i^* > 0$ for $0 \leq i \leq i_1$. Note also that $\rho_{II}^*(0) = 0$ because $\overline{m}_{II}(0) = 0$ by (20) and this is consistent with (16).

We have shown that all of x_i^* , $\rho_\tau^*(i)$ are uniquely determined. It only remains to show that $\underline{m}_B(i) < \overline{m}_I(i)$ and $(\rho_\tau^*(i), x_i^*)$ satisfies (17). First note that

$$\begin{aligned} \underline{m}_A(i) \geq \underline{m}_A(i_1) &\geq M_2^2 - M_2(d-1)(4(1+\gamma)(d-1)^2 + 3d^2 + d + 6) \\ &= d(d-1)^2n(dn-7d^2+7d-10-4\gamma(d-1)^2) \end{aligned} \quad (26)$$

which is positive by (24).

By (13),

$$\frac{x_{i+1}^*}{x_i^*} \leq \frac{\overline{m}_I(i+1)}{\underline{m}_A(i)(1-\epsilon)}, \quad (27)$$

and hence by (15),

$$\rho_{II}^*(i) = (1-\epsilon) \frac{x_{i+1}^*}{x_i^*} \cdot \frac{\overline{m}_{II}(i)}{\overline{m}_I(i+1)} \leq \frac{\overline{m}_{II}(i)}{\underline{m}_A(i)}.$$

By (20) and (21), and using $i \leq i_1 \leq (1+\gamma)(d-1)^2/4$,

$$\rho_{II}^*(i) \leq \frac{4(1+\gamma)(d-1)^4(d-2)(d-3)}{\underline{m}_A(i)},$$

which is at most ϵ by (26) and (23). Hence $\rho_I^*(i) + \rho_{II}^*(i) \leq 1$ as required for (17). In the first paragraph of this proof it was observed that $\rho_I^*(i)$ and $\rho_{II}^*(i)$ are both nonnegative.

To prove $\underline{m}_B(i) < \overline{m}_I(i)$, noting that $\underline{m}_B(i) < 16id^4(n/d-3)$ and $\overline{m}_I(i) > 4i(dn-d^2)^2 = 4id^4(n/d-1)^2$, it is sufficient to prove that $4(n/d-3) < (n/d-1)^2$. This inequality is trivially true as $(n/d-1)^2 - 4(n/d-3) = (n/d-3)^2 + 4 > 0$.

The last claim in the lemma is immediate since $d = o(\sqrt{n})$. ■

Note that the proof of Lemma 7 provides a computation scheme for a solution to system (13)–(15) with initial conditions (16). It is solved by iterated substitutions in $O(i_1) = O(d^2)$ steps.

Let $\gamma > 0$ and ϵ be chosen to satisfy (23) and (24). Then γ will be used to define \mathcal{A}_γ in Section 8. To complete the definition of phase 3 of *REG*, as foreshadowed in Section 6, we then set $\rho_\tau(i)$ for phase 3 to be the solution determined in Lemma 7. Our main concern in this paper is for large n , but there is a technical issue here. If no such γ and ϵ satisfy (23) and (24), *REG* would not be technically defined. This is circumvented by defining (in Section 8) $i_1 = 0$ in that case. Then the system (13)–(17) has the trivial solution, and phase 3 does, and needs to do, nothing.

7.5 Probability of rejection

Assume that γ and ϵ are chosen to satisfy (23)–(24). By Lemma 7, (13)–(17) has a solution and this properly defines $\rho_\tau(i)$.

We are assuming that γ is fixed, and $\epsilon = O(d^2/n^2)$ has been chosen as described at the end of the previous subsection. Our next task is to bound the number of switching steps that the algorithm will perform in phase 3, in expectation, per graph generated. The key points to consider are the distribution of the number of steps in the phase, and the probability of rejection occurring during the phase.

Lemma 8. *The number of switching steps in phase 3 is $O(i_1)$, both in expectation for sufficiently large n , and with probability $1 - O(n^{-2})$.*

Proof. Let $t \geq i_1$ be an integer. Assume that phase 3 contains at least t switching steps. Among the first t switching steps, let x denote the number of times that the number of double edges does not decrease. Since it increases by at most 1 per step, and the number

of double edges in P_0 is at most i_1 , we have $i_1 + x - (t - x) \geq 0$. It follows then that $x \geq (t - i_1)/2$. If a switching does not decrease the number of double edges, it must be either of type I and class B, or of type II. The probability of performing a type II switching for any given switching step in phase 3 (conditional upon the history of the process) is $\rho_{II}(i)$, which is at most $\epsilon = O(d^2/n^2)$. Assume that a type I switching is applied to a pairing $P \in \mathcal{S}_i$ for some $1 \leq i \leq i_1$ in a switching step. It is easy to see that $f_I(P) = \Omega(iM_1^2)$ (see a more accurate lower bound for $f_I(P)$ in the proof of Lemma 12 below), and similarly the number of type I class B switchings that can be applied to P is $O(id^2M_1)$. Thus, conditional upon the chosen type of switching being I, the probability that the switching is in class B is $O(d^2/M_1) = O(d/n)$. Combining the above two cases, the probability that the number of double edges does not decrease in a given switching step is $O(d/n)$. Hence, the probability that $x \geq (t - i_1)/2$ is at most

$$\binom{t}{(t - i_1)/2} (O(d/n))^{(t - i_1)/2} \leq 2^t (O(d/n))^{(t - i_1)/2}.$$

Putting $t = 17i_1$, it is clear that the above probability is $O(n^{-2})$ (for all sufficiently large d it is clear that choosing $t = 3i_1$ suffices; choosing $t = 17i_1$ allows to get the uniform probability bound for cases such as $d = 2$). Hence, with probability $1 - O(n^{-2})$, phase 3 contains at most $17i_1$ switching steps. The expected number of switching steps in phase 3 is at most

$$2i_1 + \sum_{t \geq 2i_1} t 2^t (O(d/n))^{t/4} = 2i_1 + O(1). \blacksquare$$

Lemma 9. *The probability that phase 3 terminates with a t-rejection is $O(d^4/n^2)$.*

Proof. By our choice of $\rho_\tau(i)$, in each switching step of phase 3, a t-rejection occurs with probability at most $\max_{1 \leq i \leq i_1} (1 - \rho_I(i) - \rho_{II}(i)) \leq 1 - \rho_I(i) = \epsilon = O(d^2/n^2)$. By Lemma 8, the total number of switching steps in phase 3 is $O(i_1)$ with probability $1 - O(n^{-2})$. This implies that the probability of a t-rejection during phase 3 is $O(n^{-2} + i_1 d^2/n^2) = O(d^4/n^2)$. \blacksquare

Before attempting to bound the probability of an f-rejection or b-rejection in phase 3, we prove two technical lemmas about properties of a random pairing in \mathcal{S}_i . Given a set of pairs S , let $V(S)$ denote the set of vertices that are incident with S . We say a set of pairs K *excommunicate* S if all pairs in K are single edges, $V(K) \cap V(S) = \emptyset$, and no vertex in $V(K)$ is adjacent to a vertex in $V(S)$.

Lemma 10. *Let u, v be two specified vertices. Assume P is a random pairing of \mathcal{S}_i conditional on containing uv as a single edge (or a double edge, or that u is not adjacent to v). Let x be a randomly selected pair in P . The probability that one end vertex of x is adjacent to u , and the other adjacent to v , and all four vertices are distinct, is $O(d^2/n^2)$.*

Proof. The proof is basically the same for all three cases, so it is enough to assume that uv is a double edge. Label the pairs in the edge uv as $\{1,2\}$ and $\{3,4\}$, with points 1 and 3 in u . We will use a “subsidiary” switching. To select a random pair x , we may let it be the pair containing a randomly selected point. Let the end points of x be labelled 5 and 6, the vertex containing 5 labelled u_1 and the one containing 6 labelled v_1 . We will bound the probability that u_1 is adjacent to u and v_1 is adjacent to v by $O(d^2/n^2)$. The lemma then

follows by symmetry. So far we have only exposed pairs in the edge uv and pair x , and we can assume that all four vertices involved are distinct. All other points are paired uniformly at random in P conditional on $P \in \mathcal{S}_i$.

Pick a point 7 from u_1 , a point 8 from u , a point 9 from v_1 and a point 10 from v . We next estimate the probability that $y = \{7, 8\}$ and $z = \{9, 10\}$ are pairs in P , and both represent single edges.

Let C_1 be the set of pairings that contain the pairs $\{2i-1, 2i\}$ for all $1 \leq i \leq 5$, such that y and z represent single edges. For any $P \in C_1$, pick sequentially another two single-edge pairs labelled $\{11, 12\}$ and $\{13, 14\}$ that each excommunicates all previously chosen/exposed pairs. Perform a switching of pairs as in Figure 6. Then the number of valid switchings is $\Theta(M_1^2)$, since there are $M_1 - O(d^2) = M_1 - o(n) \sim M_1$ options for each of these two single-edge pairs. Let C be the set of pairings in \mathcal{S}_i containing $\{u, v\}$ as a double edge, as well as the pair x . Observe that each subsidiary switching produces a pairing in \mathcal{S}_i and indeed in C . For any pairing $P' \in C$, at most one subsidiary switching can produce P' , because all points $1 \leq i \leq 10$ are specified and so are points $11 \leq i \leq 14$. Hence, the probability that a random $P \in C$ contains pairs y and z as single-edge pairs is

$$\frac{|C_1|}{|C|} = O(1/M_1^2).$$

There are at most d^4 ways to choose points $7 \leq i \leq 10$. Thus, the probability that u_1 is adjacent to u with a single edge and v_1 is adjacent to v with a single edge is $O(d^4/M_1^2) = O(d^2/n^2)$.

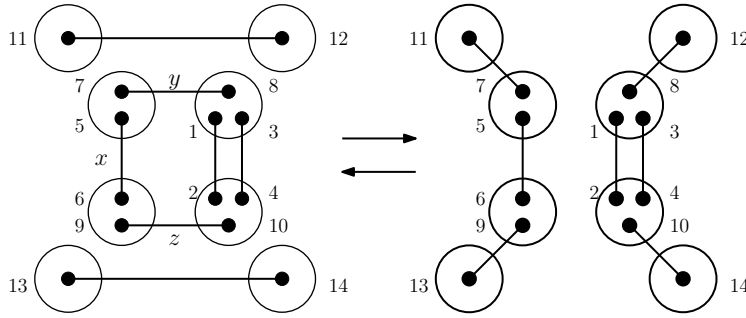


Figure 6: *A subsidiary switching*

Next, we bound the probability that u_1 is adjacent to u via a double edge. Pick points 7 and 9 from u_1 and 8 and 10 from u . We first bound the probability that $\{7, 8\}$ and $\{9, 10\}$ form a double edge u_1u . Suppose the pairing P does contain $\{7, 8\}$ and $\{9, 10\}$ (as on the left side of Figure 7). Pick sequentially another two 2-paths that each excommunicates all previous chosen/exposed pairs. Switch some of the pairs as in Figure 7. In this case, the number of possible switchings is $\Theta(M_2^2)$. On the other hand, the number of switchings that produce any given pairing in C is $O(i)$. This is because all points $1 \leq i \leq 10$ are specified and so are their mates, and there are i ways to choose the double edge to play the role of the one on the right side of Figure 7 created by the switching. Hence, the probability that $\{7, 8\}$ and $\{9, 10\}$ are both pairs in P is $O(i/M_2^2)$. There are at most d^4 ways to choose

points 7, 8, 9, and 10. Hence, the probability that u_1 is adjacent to u with a double edge is $O(id^4/M_2^2) = O(i/n^2)$. By symmetry, the probability that v_1 is adjacent to v with a double edge is also $O(i/n^2)$.

Noting that $i \leq i_1 = O(d^2)$, this probability is $O(d^2/n^2)$. This completes the proof. ▀

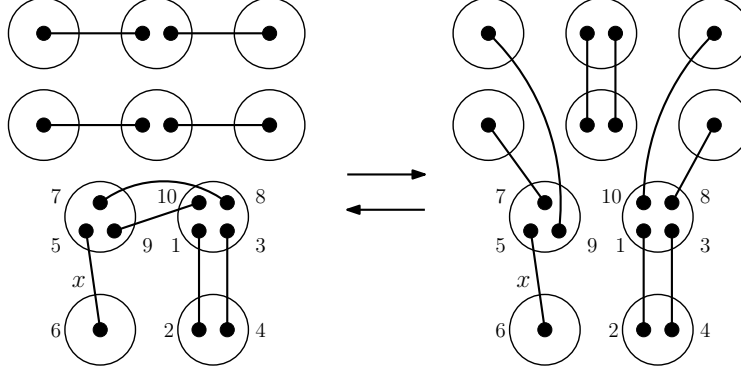


Figure 7: *Another subsidiary switching*

Lemma 11. *Let P be a random pairing in \mathcal{S}_i with $i \geq 2$. The expected number of pairs of double edges sharing a common vertex is $O(i^2/n)$.*

Proof. Let u , u_1 and u_2 be three distinct vertices. Pick four points in u , two points in u_1 and two points in u_2 . Label these points as in Figure 8. We estimate the probability that pairs $\{2i-1, 2i\}$ for all $1 \leq i \leq 4$ are in P . Let C_1 be the set of pairings containing these four pairs. For any $P \in C_1$, consider the following switching. Choose sequentially four 2-paths, each of which excommunicates all previously chosen and exposed pairs. Switch these pairs as in Figure 8. There are $\Omega(M_2^4)$ ways to perform such a switching for every pairing in C_1 , whereas for every pairing $P' \in \mathcal{S}_i$ ($i \geq 2$), the number of switchings producing P' is $O(i^2)$, as all points $1 \leq i \leq 8$ are specified and there are $O(i^2)$ ways to choose the two double edges as on the right side of Figure 8. Hence, the probability that P contains pairs $\{2i-1, 2i\}$ for all $1 \leq i \leq 4$ is $O(i^2/M_2^4)$. The number of ways to choose u , u_1 and u_2 and then to choose points $1 \leq i \leq 8$ is at most $M_4 M_2^2$. By linearity, the expected number of pairs of double edges sharing a common vertex is $O(M_4 M_2^2 \cdot i^2/M_2^4) = O(i^2/n)$.

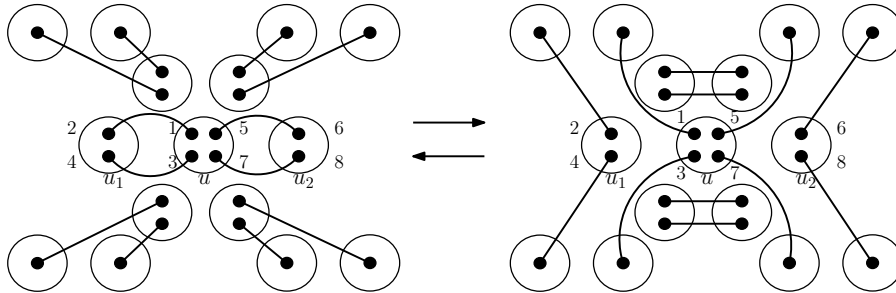


Figure 8: *Switching away a pair of double edges sharing a common vertex*

7.5.1 Probability of an f-rejection

Recall that P_t is the pairing obtained after the t -th switching step of phase 3. For a given pairing P , conditional on $P_t = P$, we bound the probability an f-rejection occurs at the $(t+1)$ -th switching step. Assume $P \in \mathcal{S}_i$; the probability that a type I switching is performed on P and is f-rejected, is

$$\rho_I(i) \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right).$$

On the other hand, the probability that a type II switching is performed on P and is f-rejected is $O(d^2/n^2)$, as $\rho_{II}(i) = O(d^2/n^2)$ by Lemma 7. Taking the union bound over all switching steps, the probability of an f-rejection during phase 3 is at most

$$\sum_{t \geq 0} \sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \mathbb{P}(P_t = P) \left(\rho_I(i) \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right) + O(d^2/n^2) \right).$$

Note that $\sum_{t \geq 0} \mathbb{P}(P_t = P)$ is the expected number of times that P is reached, which is $\sigma(i)$ for every $P \in \mathcal{S}_i$ by Lemma 4. Hence, the above probability is

$$\sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \sigma(i) \left(\rho_I(i) \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right) \right) + \sum_{1 \leq i \leq i_1} |\mathcal{S}_i| \sigma(i) \cdot O(d^2/n^2). \quad (28)$$

Since $\rho_I(i) \leq 1$ always, and since by Lemma 8,

$$\sum_{1 \leq i \leq i_1} |\mathcal{S}_i| \sigma(i) = O(i_1) = O(d^2), \quad (29)$$

the probability of an f-rejection during phase 3 is at most

$$\sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \sigma(i) \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right) + O(d^4/n^2). \quad (30)$$

We will prove the following lemma by bounding the first term of (30).

Lemma 12. *The probability of an f-rejection during phase 3 is $O(d^2/n)$.*

Proof. By (30), it is sufficient to bound $\sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \sigma(i) (1 - f_I(P)/\overline{m}_I(i))$ by $O(d^2/n)$. Note that

$$\sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \sigma(i) \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right) = \sum_{1 \leq i \leq i_1} \sigma(i) |\mathcal{S}_i| \cdot \mathbb{E} \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right),$$

where the above expectation is taken on a random pairing $P \in \mathcal{S}_i$. It will be sufficient to show that this expectation is $O(1/n)$ for every $1 \leq i \leq i_1$, since then

$$\sum_{1 \leq i \leq i_1} \sigma(i) |\mathcal{S}_i| \cdot \mathbb{E} \left(1 - \frac{f_I(P)}{\overline{m}_I(i)} \right) = O(i_1/n) = O(d^2/n)$$

and the lemma follows.

Assume $P \in \mathcal{S}_i$. Let $X = \overline{m}_I(i) - f_I(P)$. Pick and label pairs and vertices as in the description of the type I switching (see Figure 2). Here, X counts choices and labellings such that any of the following occur:

- (a) the six vertices are not all distinct;
- (b) the vertices are distinct and at least two new double edges are created after performing the switching;
- (c) some triple edge is created by the switching because there is a double edge already between u_1 and u_2 or u_3 , or between v_1 and v_2 or v_3 .

It is easy to see that the number of choices for (a) is $O(idM_1)$. For (b), there are two sub-cases (we use $u \sim v$ to denote that u is adjacent to v).

- (b1) $u_1 \sim u_2$ and $u_1 \sim u_3$, or $v_1 \sim v_2$ and $v_1 \sim v_3$, or $u_1 \sim u_2$ and $v_1 \sim v_3$, or $u_1 \sim u_3$ and $v_1 \sim v_2$;
- (b2) $u_1 \sim u_3$ and $v_1 \sim v_3$, or $u_1 \sim u_2$ and $v_1 \sim v_2$.

It is straightforward to see that the number of choices for (b1) is at most $O(id^4)$. For (b2), we consider a random $P \in \mathcal{S}_i$ which contains u_1v_1 as a double edge. Let x be a random pair chosen uniformly from P ; with u_2 and v_2 denoting its end vertices. By Lemma 10, the probability that u_1 is adjacent to u_2 and v_1 is adjacent to v_2 is $O(d^2/n^2)$. Hence, the expected number of choices for (b2) is $O(iM_1^2 \cdot d^2/n^2)$, as there are $O(i)$ ways to choose u_1v_1 and label their end points, M_1^2 ways to sequentially choose two random pairs, uniformly at random and with repetition, and by the above argument, the probability that one of these two pairs forcing (b2) is $O(d^2/n^2)$. By (19) and since $i \leq i_1 = O(d^2) = o(M_1)$, $iM_1^2 \cdot d^2/n^2 = \overline{m}_I(i) \cdot O(d^2/n^2)$.

We can count the possibilities for (c) by first choosing a vertex (for u_1 or v_1) incident with two double edges — by Lemma 11 there are $O(i^2/n)$ ways to do this in expectation — and then choosing the remaining pairs to be switched, in $O(dM_1)$ ways. For a random $P \in \mathcal{S}_i$, the expected number of type I switchings satisfying (c) is therefore $O(i^2dM_1/n) = o(idM_1)$ since $i = O(d^2) = o(n)$.

Combining the three bounds, we find that $\mathbb{E}X = O(idM_1 + id^4) + \overline{m}_I(i) \cdot O(d^2/n^2)$ and so since $d^2 = o(n)$ we have $\mathbb{E}(1 - f_I(P)/\overline{m}_I(i))$ is $O(idM_1 + id^4)/\overline{m}_I(i) + O(d^2/n^2) = O(1/n)$, as desired. ■

7.5.2 Probability of a b-rejection

Let $\mathcal{S}_{\tau,\alpha}^-(P)$ denote the set of pairings that can produce P using a type τ , class α switching.

Now, we bound the probability of a b-rejection during phase 3. Given $P \in \mathcal{S}_i$, we consider a pairing $P' \in \mathcal{S}_{I,A}^-(P)$. Then P' must be in \mathcal{S}_{i+1} . Conditional on $P_{t-1} = P'$, the probability that the switching S , which converts P' to P , is chosen and is not f-rejected at the t -th switching step is

$$\frac{\rho_I(i+1)}{\overline{m}_I(i+1)}.$$

Conditional on that, the probability that P is b-rejected is $1 - \underline{m}_A(i)/b_A(P)$. Similarly, for any $P' \in \mathcal{S}_{I,B}^-(P) \cup \mathcal{S}_{II,B}^-(P)$, conditional on $P_{t-1} = P'$, the probability that the switching converting P' to P is chosen at the t -th switching and is not f-rejected but is b-rejected is

$$\frac{\rho_I(i)}{\overline{m}_I(i)} \left(1 - \frac{\underline{m}_B(i)}{b_B(P)} \right),$$

if $P' \in \mathcal{S}_{I,B}^-(P)$, and is

$$\frac{\rho_{II}(i-1)}{\bar{m}_{II}(i-1)} \left(1 - \frac{\underline{m}_B(i)}{b_B(P)}\right).$$

if $P' \in \mathcal{S}_{II,B}^-(P)$. Hence, by the union bound, the probability that a b-rejection occurs during phase 3 is at most

$$\begin{aligned} & \sum_{t \geq 1} \sum_{0 \leq i \leq i_1 - 1} \sum_{P \in \mathcal{S}_i} \sum_{P' \in \mathcal{S}_{I,A}^-(P)} \mathbb{P}(P_{t-1} = P') \frac{\rho_I(i+1)}{\bar{m}_I(i+1)} \left(1 - \frac{\underline{m}_A(i)}{b_A(P)}\right) \\ & + \sum_{t \geq 1} \sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \left(1 - \frac{\underline{m}_B(i)}{b_B(P)}\right) \left(\sum_{P' \in \mathcal{S}_{I,B}^-(P)} \mathbb{P}(P_{t-1} = P') \frac{\rho_I(i)}{\bar{m}_I(i)} \right. \\ & \quad \left. + \sum_{P' \in \mathcal{S}_{II,B}^-(P)} \mathbb{P}(P_{t-1} = P') \frac{\rho_{II}(i-1)}{\bar{m}_{II}(i-1)} \right) \\ & = \sum_{0 \leq i \leq i_1 - 1} \sum_{P \in \mathcal{S}_i} \sum_{P' \in \mathcal{S}_{I,A}^-(P)} \sigma(i+1) \frac{\rho_I(i+1)}{\bar{m}_I(i+1)} \left(1 - \frac{\underline{m}_A(i)}{b_A(P)}\right) \\ & + \sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \left(1 - \frac{\underline{m}_B(i)}{b_B(P)}\right) \left(\sum_{P' \in \mathcal{S}_{I,B}^-(P)} \sigma(i) \frac{\rho_I(i)}{\bar{m}_I(i)} + \sum_{P' \in \mathcal{S}_{II,B}^-(P)} \sigma(i-1) \frac{\rho_{II}(i-1)}{\bar{m}_{II}(i-1)} \right). \end{aligned}$$

By (10), $\sigma(i)\rho_I(i)/\bar{m}_I(i) = \sigma(i-1)\rho_{II}(i-1)/\bar{m}_{II}(i-1)$ for every $1 \leq i \leq i_1$. So, using $\rho_I(i) \leq 1$, the probability of a b-rejection during phase 3 is at most

$$\begin{aligned} & \sum_{0 \leq i \leq i_1 - 1} \sum_{P \in \mathcal{S}_i} \left(\sum_{P' \in \mathcal{S}_{I,A}^-(P)} \frac{\sigma(i+1)}{\bar{m}_I(i+1)} \left(1 - \frac{\underline{m}_A(i)}{b_A(P)}\right) \right) \\ & + \sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \left(\sum_{P' \in \mathcal{S}_{I,B}^-(P) \cup \mathcal{S}_{II,B}^-(P)} \frac{\sigma(i)}{\bar{m}_I(i)} \left(1 - \frac{\underline{m}_B(i)}{b_B(P)}\right) \right). \end{aligned} \quad (31)$$

By the definition of $b_\alpha(P)$ for $\alpha \in \{A, B\}$,

$$|\mathcal{S}_{I,A}^-(P)| = b_A(P), \quad |\mathcal{S}_{I,B}^-(P) \cup \mathcal{S}_{II,B}^-(P)| = b_B(P).$$

Then, the above expression equals

$$\sum_{0 \leq i \leq i_1 - 1} \sum_{P \in \mathcal{S}_i} \frac{\sigma(i+1)}{\bar{m}_I(i+1)} (b_A(P) - \underline{m}_A(i)) + \sum_{1 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} \frac{\sigma(i)}{\bar{m}_I(i)} (b_B(P) - \underline{m}_B(i)). \quad (32)$$

By (27) and recalling that $x_j^* = \sigma(j)/|\Phi_1|$,

$$\frac{\sigma(i+1)}{\bar{m}_I(i+1)} = O\left(\frac{\sigma(i)}{\underline{m}_A(i)}\right), \quad (33)$$

whereas by (19) and (22),

$$\frac{\sigma(i)}{\overline{m}_I(i)} = \frac{\sigma(i)}{\underline{m}_B(i)} \frac{\underline{m}_B(i)}{\overline{m}_I(i)} = \frac{\sigma(i)}{\underline{m}_B(i)} \cdot O\left(\frac{idM_2}{iM_1^2}\right) = \frac{\sigma(i)}{\underline{m}_B(i)} \cdot O(d/n). \quad (34)$$

Substituting bounds (33) and (34) into (32) yields

$$\begin{aligned} & \sum_{0 \leq i \leq i_1} \sum_{P \in \mathcal{S}_i} O(\sigma(i)) \left(\frac{b_A(P) - \underline{m}_A(i)}{\underline{m}_A(i)} + (d/n) \left(\frac{b_B(P) - \underline{m}_B(i)}{\underline{m}_B(i)} \right) \right) \\ &= \sum_{1 \leq i \leq i_1} O(\sigma(i)|\mathcal{S}_i|) \left(\frac{\mathbb{E}b_A(P) - \underline{m}_A(i)}{\underline{m}_A(i)} + (d/n) \left(\frac{\mathbb{E}b_B(P) - \underline{m}_B(i)}{\underline{m}_B(i)} \right) \right), \end{aligned} \quad (35)$$

by noting that $b_A(P) = 0$ for $P \in \mathcal{S}_{i_1}$ and $b_B(P) = \underline{m}_B(0) = 0$ for $P \in \mathcal{S}_0$, and by resetting $\underline{m}_A(i_1) = 0$ as the first term in (32) does not include $i = i_1$. By bounding these expectations we will prove the following lemma.

Lemma 13. *The probability of a b -rejection during phase 3 is $O(d^2/n)$.*

Proof. We first bound $\mathbb{E}b_A(P)$ from above for a random $P \in \mathcal{S}_i$. Clearly M_2^2 is an upper bound. However, we can exclude from it the choices of the pair of 2-paths such that

- (a) Exactly one of these four pairs is contained in a double edge and the two paths do not share any vertex;
- (b) All of the four pairs in the 2-paths represent single edges, and the two paths are vertex-disjoint, and u_i is adjacent to v_i for exactly one of $i = 1, 2, 3$.

Clearly, the sets of pairings counted in (a) and (b) do not intersect. Let X_a and X_b denote the sizes of these two sets. Then $b_A(P) \leq M_2^2 - X_a - X_b$. We wish to estimate lower bounds for X_a and X_b .

For (a), consider the case that the pair $\{3, 7\}$ is contained in a double edge, as on the right side of Figure 3. There are $4i$ ways to choose point 3. That choice automatically sets points 7, 9 and 10. Then, there are $(d-2)$ ways to pick point 1, which sets point 5 as well. Given the choice of the first 2-path, there are $M_2 - O(d^2 + id)$ ways to choose the other 2-path so that it does not use any vertex u_i , and it does not use a pair contained in a double edge. Hence, by symmetry, $X_a = 16i(d-2)(M_2 - O(d^2 + id)) - O(X')$, where X' counts choices where one of the two 2-paths uses two pairs such that each pair is contained in a double edge. By Lemma 11, $\mathbb{E}(X') = O(M_2 i^2/n) = O(d^6)$, as $i \leq i_1 = O(d^2)$, and the expected number of choices for the 2-paths with both pairs contained in double edges is $O(i^2/n)$, whereas M_2 bounds the number of choices for the other 2-path. Now, we have $\mathbb{E}X_a \geq 16i(d-2)(M_2 - O(d^2 + id)) - O(d^6)$.

For (b), there are M_2 ways to choose the first path and then approximately $2(d-1)^3 + (d-2)^2(d-1)$ ways to choose the other 2-path. So, $X_b = M_2(2(d-1)^3 + (d-2)^2(d-1)) - Y$, where Y counts the choices of the two 2-paths such that

- (c) the two paths are not vertex disjoint, or

- (d) they are vertex-disjoint, and one of the two paths contains a double edge and $u_i \sim v_i$ for some i , or
- (e) they are vertex-disjoint, and at least two of $u_i v_i$ are edges.

The number of choices for (c) is $O(M_2 d^2)$; the number of choices for (d) is $O(id^4)$. For (e), first consider the case that $u_1 v_1$ and $u_2 v_2$ are edges. There are $O(M_1)$ choices for the edge $u_1 v_1$. Consider a random pairing $P \in \mathcal{S}_i$ that contains $u_1 v_1$ as an edge. Pick a random pair x in P . Denote its end vertices by u_2 and v_2 . By Lemma 10, the probability that $u_1 \sim u_2$ and $v_1 \sim v_2$ is $O(d^2/n^2)$. Thus, the expected number of such choices for u_1, u_2, v_1, v_2 and pairs between them is $O(M_1^2 \cdot d^2/n^2)$. After fixing these, there are $O(d^2)$ ways to choose edges $u_1 u_3$ and $v_1 v_3$. This gives $O(M_1^2 d^4/n^2) = O(d^6)$ for the number of choices such that $u_1 \sim v_1$ and $u_i \sim v_i$ for some $i = 2, 3$ by symmetry. Lastly, consider the case that $u_i \sim v_i$ for $i = 2$ and 3 . There are at most n^2 ways to choose u_1 and v_1 . Then, randomly pick 2 pairs and denote their end vertices by u_2 and v_2 , and u_3 and v_3 respectively. A trivial adaptation of the proof as Lemma 10 shows that the probability that the four edges $u_1 u_i$ and $v_1 v_i$ for $i = 2, 3$ are present is $O(d^4/n^4)$. Hence, the number of choices in this case is $O(n^2 \cdot M_1^2 \cdot d^4/n^4) = O(d^6)$. This gives

$$\mathbb{E}X_b = 2M_2(d-1)^3 + M_2(d-2)^2(d-1) - O(M_2 d^2 + id^4 + d^6) = 3M_2 d^3 + O(M_2 d^2 + d^6),$$

using that $i \leq i_1 = O(d^2)$.

Combining (a) and (b), $\mathbb{E}b_A(P) \leq M_2^2 - 16idM_2 - 3M_2 d^3 + O(d^6 + M_2 d^2)$. Recall from (21) that $\underline{m}_A(i) = M_2^2 - M_2 d(16i + 9d + 3d^2)$. So, $\mathbb{E}b_A(P) - \underline{m}_A(i) = O(d^6 + M_2 d^2)$.

It is easy to see that $b_B(P) \leq 16i(d-2)M_2$ for every $P \in \mathcal{S}_i$. Hence, by (22),

$$\mathbb{E}(b_B(P) - \underline{m}_B(i)) = O(id^4).$$

By (35), the probability of a b-rejection during phase 3 is

$$\sum_{1 \leq i \leq i_1} O(\sigma(i)|\mathcal{S}_i|) \left(\frac{d^6 + M_2 d^2}{\underline{m}_A(i)} + (d/n) \left(\frac{id^4}{\underline{m}_B(i)} \right) \right).$$

It is easy to see that $\underline{m}_A(i) = \Omega(M_2^2)$ and $\underline{m}_B(i) = \Omega(idM_2)$ by (22) and the assumption that $d^2 = O(n)$. So the above probability is bounded by

$$O(1/n) \sum_{1 \leq i \leq i_1} \sigma(i)|\mathcal{S}_i| = O(i_1/n) = O(d^2/n),$$

by (29). The lemma follows. ■

8 Initial rejection: defining \mathcal{A}_γ

Here we define \mathcal{A}_γ , after some preliminaries. It is easy to see that

$$|\Phi| = (M_1 - 1)!! := \prod_{i=0}^{M_1/2-1} (M_1 - 2i - 1),$$

and the probability that a random $P \in \Phi$ contains a given set of s pairs is

$$\frac{(M_1 - 2s - 1)!!}{(M_1 - 1)!!} = \prod_{i=0}^{s-1} \frac{1}{M_1 - 1 - 2i}. \quad (36)$$

Let $D(P)$, $L(P)$ and $T(P)$ denote the number of double edges, loops and triple edges, respectively, in a pairing in $P \in \Phi$. Define

$$\mathcal{A}_\gamma = \{P \in \Phi : D(P) \leq B_D, L(P) \leq B_L, T(P) \leq B_T\}$$

where

$$B_L = \max\{\lceil \log dn \rceil, 4d\}, \quad B_T = \max\{\lceil \log dn \rceil, \lceil d^3/n \rceil\}, \quad B_D = \left\lfloor (1 + \gamma) \frac{(d-1)^2}{4} \right\rfloor, \quad (37)$$

noting that the definition of B_D agrees with the definition of i_1 in (18), and subject to the following proviso. If γ is such that no ϵ satisfies the conditions in Lemma 7 for phase 3, we replace the bound on $D(P)$ by the condition $D(P) = 0$. That is, $i_1 = 0$ for phase 3. (This will increase the probability of initial rejection, but only for $n = O(d)$, as discussed after Lemma 7.)

Lemma 14. *Assume $d = o(\sqrt{n})$. For any $\gamma > 0$, the probability of an initial rejection is at most $1/(1 + \gamma) + o(1)$.*

Proof. By [14, Lemma 3.2], the probability that a random pairing $P \in \Phi$ contains more than B_L single loops or more than B_T triple edges, or any multi-edge other than double or triple edges, is $O(d^2/n) = o(1)$. We also have

$$\mathbb{E}D = \sum_{1 \leq i < j \leq n} \frac{\binom{d}{2} \binom{d}{2} 2!}{(M_1 - 1)(M_1 - 3)} = (1 + o(1))(d-1)^2/4.$$

By Markov's inequality, the probability that a random pairing $\mathcal{P} \in \Phi$ has at least $(1 + \gamma)(d-1)^2/4$ double edges is at most $1/(1 + \gamma) + o(1)$. The lemma follows, recalling again that by the remarks after Lemma 7, the condition $D(P) = 0$ will only be imposed for $n = O(d)$. ■

9 Reduction of loops and triple edges

Let P_0 be a random pairing in Φ . If $P_0 \in \mathcal{A}_\gamma$, then *REG* sequentially enters phases 1 and 2 where the loops and triple edges will be switched away, one at a time. In each of these two phases, all switchings have the same type and class. Moreover, these are the same switchings as used for enumeration purposes in [14]. In view of the remark below Lemma 4, phase 1 will be the same as the corresponding phase in the algorithm *DEG* of [12]. Phase 2 is similar to phase 1, just with different switchings. (For the range of d considered in [12] there are no triple edges in pairings in \mathcal{A} and so *DEG* has no phase of reduction of triple edges.)

We define the switchings and analyse the rejection probabilities for phases 1 and 2 in Sections 9.1 and 9.2 respectively. This analysis is similar to that in [12].

9.1 Phase 1: loop reduction

In this phase, only one type of switching is used, which is defined as follows.

Type L switching. Let P be a pairing containing at least one loop. Pick a loop from P and label its end points by 1 and 2. Pick another two pairs y and z , both of which represent single non-loop edges, and label their end points by 3 and 4, and 5 and 6 respectively. Let the vertices containing these points be labelled as in Figure 9. The type L switching replaces these pairs by $\{1,3\}$, $\{4,6\}$ and $\{2,5\}$. This switching is valid only if the five vertices v_i ($1 \leq i \leq 5$) are all distinct, and there are no new multi-edges or loops created after its performance. All type L switchings are in class A .

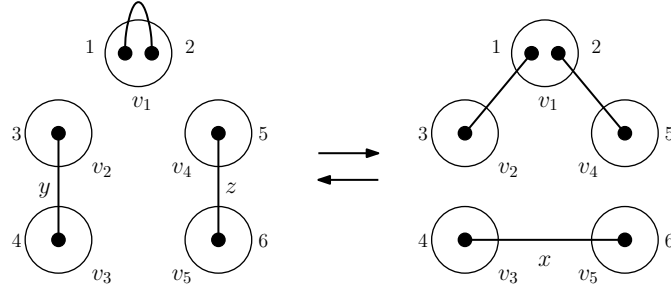


Figure 9: *type L switching*

Note that a type L switching always reduces the number of loops in a pairing by one. In this phase, \mathcal{S}_i is the set of pairings in \mathcal{A}_γ with exactly i loops, and i_1 is set equal to B_L defined in (37). Here, $\Phi_1 = \mathcal{A}_\gamma$, which is $\cup_{0 \leq i \leq i_1} \mathcal{S}_i$. We set $\rho_L(i) = 1$ for every $1 \leq i \leq i_1$.

Define

$$\begin{aligned} \overline{m}_L(i) &= 2iM_1^2 \\ \underline{m}_A(i) &= M_2M_1 - 2dM_1(3i + 6B_D + 9B_T + 3d + d^2). \end{aligned} \quad (38)$$

Lemma 15. *During the loop reduction phase, for each $P \in \mathcal{S}_i$, $f_L(P) \leq \overline{m}_L(i)$ and $b_A(P) \geq \underline{m}_A(i)$.*

Proof. We first prove the upper bound. Let $P \in \mathcal{S}_i$. The number of ways to choose a loop and label its end points in P is $2i$. Then, the number of ways to choose another two pairs and label their end points is at most M_1^2 . This shows that $f_L(P) \leq \overline{m}_L(i)$ for any $P \in \mathcal{S}_i$.

Next we prove the lower bound. Given $P \in \mathcal{S}_i$, $b_A(P)$ is the number of type L switchings that produce P . To count such switchings, we need to specify a 2-path and label the points inside by $\{3,1\}$ and $\{2,5\}$, together with another pair representing a single edge, whose end points are labelled by 4 and 6, and label all vertices containing these points as on the right side of Figure 9. Moreover, the five vertices must be distinct. There are M_2 ways to pick the 2-path and M_1 ways to pick the other pair. We need to exclude every choice in which at least one of the following holds:

- (a) the 2-path contains a loop, or a double edge, or a triple edge;
- (b) pair $\{4,6\}$ is a loop or is contained in a multi-edge;

- (c) pair $\{4, 6\}$ share a vertex with the 2-path;
- (d) v_2 and v_3 are adjacent or v_4 and v_5 are adjacent.

We first show that the number of choices for (a) is at most $2(2idM_1 + 4B_DdM_1 + 6B_TdM_1)$, starting with the choices in which $\{1, 3\}$ is a loop. There are $2i$ ways to choose a loop and label their end points by 1 and 3. Then, there are at most d ways to choose 2 and 5 as v_1 , the vertex containing point 1 has degree d . So the number of ways to choose the 2-path and x so that $\{1, 3\}$ is a loop is at most $2idM_1$. Similarly, the number of ways to choose such a path and a pair so that $\{1, 3\}$ is contained in a double edge or a triple edge is at most $4B_DdM_1$ and $6B_TdM_1$ respectively as there are at most B_D double edges and B_T triple edges. The extra factor 2 accounts for the case that $\{2, 5\}$ is a loop or is contained in a double or triple edge. Hence, the number of choices for (a) is at most $2dM_1(2i + 4B_D + 6B_T)$.

The number of choices for (b) is at most $M_2(2i + 4B_D + 6B_T)$, for (c) is at most $6M_2d$ and for (d) is at most $2M_2d^2$. Hence, using $M_2 \leq dM_1$,

$$\begin{aligned} b_A(P) &\geq M_2M_1 - 2dM_1(2i + 4B_D + 6B_T) - M_2(2i + 4B_D + 6B_T + 6d + 2d^2) \\ &\geq M_2M_1 - 2dM_1(3i + 6B_D + 9B_T + 3d + d^2) = \underline{m}_A(i). \blacksquare \end{aligned}$$

Let $\Phi' = \{P \in \mathcal{A}_\gamma : L(P) = 0\}$.

Lemma 16. *Assume that the algorithm is not rejected initially or during the loop reduction phase and that P' is the output after phase 1. Then P' is uniformly distributed in Φ' .*

Proof. Let P_t be the pairing obtained after the t -th switching step in phase 1. We prove by induction that P_t is uniformly distributed in the stratum that contains P_t for every t . As the algorithm is not initially rejected, P_0 is uniformly distributed in \mathcal{A}_γ , and thus is uniformly distributed in the stratum containing P_0 , providing the base case. Assume that $t \geq 1$, and P_{t-1} is uniformly distributed in \mathcal{S}_i , where $i = L(P_{t-1})$. Recall that $S_{L,A}^-(P)$ is the set of pairings that can be switched to P via a type L switching (and recall that all type L switchings are of class A). By (5), for every $P \in \mathcal{S}_{i-1}$,

$$\mathbb{P}(P_t = P) = \sum_{P' \in S_{L,A}^-(P)} \mathbb{P}(P_{t-1} = P') \frac{\underline{m}_A(i-1)}{\overline{m}_L(i)b_A(P)},$$

as every $P' \in S_{L,A}^-(P)$ must be in \mathcal{S}_i . Since P_{t-1} is uniformly distributed in \mathcal{S}_i by the inductive hypothesis, we have $\mathbb{P}(P_{t-1} = P') = 1/|\mathcal{S}_i|$. On the other hand, $b_A(P) = |S_{L,A}^-(P)|$ as all class A switchings are of type L. Immediately we have that

$$\mathbb{P}(P_t = P) = \frac{\underline{m}_A(i-1)}{|\mathcal{S}_i|\overline{m}_L(i)},$$

which is independent of P . So P_t is uniformly distributed in \mathcal{S}_{i-1} . Inductively, P' is uniformly distributed in \mathcal{S}_0 , which is Φ' . \blacksquare

Lemma 17. *The overall probability that either an f - or b -rejection occurs in phase 1 is $O((d^2 + d \log^2 n)/n)$.*

Proof. We estimate the probability that an f-rejection occurs at a given step. Assume that $P \in \mathcal{S}_i$. Then, $f_L(P) = \overline{m}_L(i) - X(P)$, where $X(P)$ is the number of choices of a loop and the two pairs $\{3, 4\}$ and $\{5, 6\}$ so that

- (a) $\{3, 4\}$ or $\{5, 6\}$ is a loop or is contained in a multi-edge — at most $2 \cdot 2iM_1(2i + 4B_D + 6B_T)$;
- (b) $v_1 \in \{v_2, v_3, v_4, v_5\}$ — at most $4 \cdot 2idM_1$;
- (c) v_1 is adjacent to v_2 or v_4 ; or v_3 is adjacent to v_5 — at most $2 \cdot 2iM_1d^2 + 2iM_1d^2$.

Hence, $X(P) \leq 2iM_1(4i + 8B_D + 12B_T + 4d + 3d^2)$. Then, the probability that P is f-rejected is

$$1 - \frac{f_L(P)}{\overline{m}_L(i)} = \frac{X(P)}{\overline{m}_L(i)} = \frac{4i + 8B_D + 12B_T + 4d + 3d^2}{M_1} \leq \frac{4i_1 + 8B_D + 12B_T + 4d + 3d^2}{M_1}.$$

As the number of loops decreases by one in each step and $L(P) \leq i_1 = B_L$, the total number of steps in phase 1 is at most i_1 . Hence, the overall probability of an f-rejection in phase 1 is at most

$$\frac{i_1(4i_1 + 8B_D + 12B_T + 4d + 3d^2)}{M_1}.$$

By (37), this is at most $O((d^2 + d \log^2 n)/n)$.

On the other hand, to assess b-rejection, assume $P \in \mathcal{S}_i$. It is obvious that $b_A(P) \leq M_2M_1$. Hence, by (38) the probability that P is b-rejected is

$$1 - \frac{m_A(i)}{b_A(P)} \leq \frac{2dM_1(3i + 6B_D + 9B_T + 3d + d^2)}{M_2M_1 - 2dM_1(3i + 6B_D + 9B_T + 3d + d^2)}.$$

As $i \leq i_1 = B_L$ and by (37), $2dM_1(3i_1 + 6B_D + 9B_T + 3d + d^2) < M_2M_1/2$ for all sufficiently large n . Therefore, the probability that P is b-rejected is at most

$$\frac{4(3i_1 + 6B_D + 9B_T + 3d + d^2)}{(d-1)n}.$$

As there are at most i_1 steps, the overall probability that a b-rejection occurs in phase 1 is at most

$$\frac{4i_1(3i_1 + 6B_D + 9B_T + 3d + d^2)}{(d-1)n} = O((d^2 + d \log^2 n)/n). \blacksquare$$

9.2 Phase 2: triple edge reduction

In this phase, we start with a pairing P_0 uniformly distributed in $\Phi_1 = \{P \in \mathcal{A}_\gamma : L(P) = 0\}$, which is a valid assumption at the start of phase 2 by Lemma 16. We will use one type of switching, called type T, defined as follows.

Type T switching. Let P be a pairing containing at least one triple edge. Pick a triple edge from P and label its end points as in Figure 10. Pick another three pairs x, y and z , all of which represent single edges, and label their end points as in Figure 10. A type

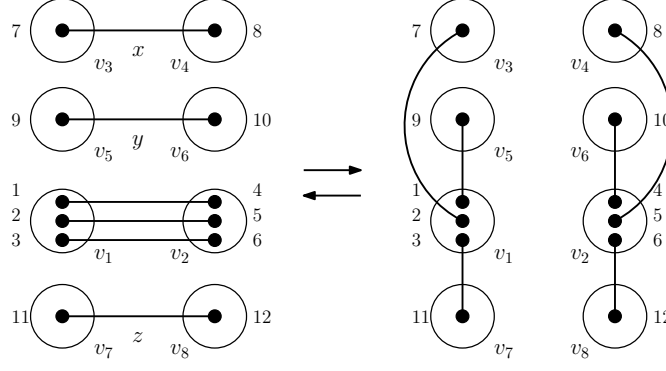


Figure 10: *type T switching*

T switching manipulates the pairs as in the figure. This switching is valid only if the eight vertices involved as in the figure are all distinct, and there are no new multi-edges or loops created by its action. All type T switchings are of class A.

In this phase, i_1 is set equal to B_T defined in (37), and hence $\Phi_1 = \cup_{0 \leq i \leq i_1} \mathcal{S}_i$ where \mathcal{S}_i is the set of pairings in Φ_1 containing exactly i triple edges. As only one type of switching is involved, we set $\rho_T(i) = 1$ for every $1 \leq i \leq i_1$.

Define

$$\begin{aligned} \overline{m}_T(i) &= 12iM_1^3 \\ \underline{m}_A(i) &= M_3^2 - 4M_3d^2(6B_D + 9i + 4d + d^2). \end{aligned} \quad (39)$$

Lemma 18. *During phase 2, for each $P \in \mathcal{S}_i$, $f_T(P) \leq \overline{m}_T(i)$ and $b_A(P) \geq \underline{m}_A(i)$.*

Proof. The upper bound is trivial. There are i ways to choose a triple edge and then 12 ways to label their end points and then at most M_1^3 ways to choose the other 3 pairs.

For the lower bound, recall that for any $P \in \mathcal{S}_i$, $b_A(P)$ is the number of type T switchings that produce P . To count these, we choose two 3-stars and label their end points as on the right side of Figure 10. The number of such choices is M_3^2 . Some of these choices are excluded:

- (a) one of the 3-stars contains a double edge or a triple edge — at most $6M_3(4B_Dd^2 + 6id^2)$;
- (b) the two 3-stars share a vertex — at most $16M_3d^3$;
- (c) there is an edge between v_{2i-1} and v_{2i} for some $i = 1, 2, 3, 4$ — at most $4M_3d^4$.

Hence, $b_A(P) \geq M_3^2 - 4M_3d^2(6B_D + 9i + 4d + d^2) = \underline{m}_A(i)$. ■

Analogous to the proof of Lemma 16 we have the following lemma, where $\Phi'' = \{P \in \Phi_1 : T(P) = 0\}$.

Lemma 19. *Assume that no rejection occurs initially or during phase 1 or 2. Then, the output of phase 2 is a pairing uniformly distributed in Φ'' .*

Lemma 20. *The probability that either an f -rejection or a b -rejection occurs during phase 2 is $O(d^4/n^2 + d \log^2 n/n)$.*

Proof. We estimate the probability of an f-rejection in each step. Assume $P \in \mathcal{S}_i$ and recall the type T switchings in Figure 10; $f_T(P) = \overline{m}_T(i) - X(P)$ where $X(P)$ is the number of choices of the triple edge and pairs such that

- (a) x or y or z is contained in a multi-edge — at most $3 \cdot 12iM_1^2(4B_D + 6i)$;
- (b) the triple edge is adjacent to x or y or z — at most $12 \cdot 12idM_1^2$;
- (c) v_1 is adjacent to v_3 or v_5 or v_7 ; or v_2 is adjacent to v_4 or v_6 or v_8 — at most $6 \cdot 12id^2M_1^2$.

Hence, $X(P) \leq 72iM_1^2(2B_D + 3i + 2d + d^2)$. So, using $i \leq i_1 = B_T$, the probability that P is f-rejected is

$$1 - \frac{f_T(P)}{\overline{m}_T(i)} = \frac{X(P)}{\overline{m}_T(i)} \leq \frac{72iM_1^2(2B_D + 3i + 2d + d^2)}{12iM_1^3} \leq \frac{6(2B_D + 3i_1 + 2d + d^2)}{M_1}.$$

As $T(P_0) \leq i_1$, the number of steps in phase 2 is at most i_1 and hence, by (37), the overall probability of an f-rejection in phase 2 is at most

$$\frac{6i_1(2B_D + 3i_1 + 2d + d^2)}{M_1} = O(d^4/n^2 + d \log^2 n/n).$$

On the other hand, for $P \in \mathcal{S}_i$, $\underline{m}_A(i) \leq b_A(P) \leq M_3^2$. Using (39), the probability that P is b-rejected is

$$1 - \frac{\underline{m}_A(i)}{b_A(P)} \leq \frac{4d^2(6B_D + 9i + 4d + d^2)}{M_3 - 4d^2(6B_D + 9i + 4d + d^2)}.$$

It is easy to see that $4M_3d^2(6B_D + 9i_1 + 4d + d^2) \leq M_3^2/2$ for all sufficiently large n . Hence, the above probability is at most

$$\frac{8d^2(6B_D + 9i_1 + 4d + d^2)}{M_3} = O(d/n + \log n/dn).$$

So the overall probability of a b-rejection in phase 2 is at most $i_1 \cdot O(d/n + \log n/dn) = O(d^4/n^2 + d \log^2 n/n)$. ■

10 *REG** and proof of Theorems 1, 2 and 3

REG possesses several complicated features to achieve uniformity of the output distribution in each phase. Some of these involve time-consuming computations such as finding the probability of a b-rejection. By omitting some of these features, we obtain a simpler sampler *REG**.

*REG** starts by repeatedly generating a random pairing $P \in \Phi$ until $P \in \mathcal{A}_\gamma$ with $\gamma = 1$ (or any positive number). Then, *REG** sequentially enters the three phases for reduction of loops, triple edges and double edges. In each phase, there will be only one type of switchings: types L and T for phases 1 and 2 respectively and type I for phase 3. Each phase consists of a sequence of switching steps in which a random switching is chosen and performed. Output the resulting pairing if it is in \mathcal{S}_0 .

In other words, REG^* is a simplification of REG by omitting f- or b-rejections in each phase, and by discarding type II switchings in phase 3. Therefore, there is no need to choose switching types in each switching step as in REG . As a result, there is no need of pre-computation for solving (13)–(17), and there is no t-rejection any more.

First we prove Theorem 3, assuming Theorem 1.

Proof of Theorem 3. The probability of REG terminating with an f-rejection, a t-rejection or a b-rejection is $o(1)$ by Lemmas 9, 12 and 13 respectively in phase 3, and similarly for the other phases. The probability of REG performing a type II switching in phase 3 is $o(1)$ by Lemma 8 and the fact that $\rho_{II}(i) = O(d^2/n^2)$ as specified just after Lemma 7. So, conditioning on the two algorithms generating the same initial pairing with no initial rejection, in an event with probability $1 - o(1)$, REG^* has the same output as REG , which is uniformly distributed. In the remaining cases, an event of conditional probability $o(1)$, either some rejection occurs in REG and REG^* carries on regardless, or a type II switching is performed in REG and REG^* produces some output (exactly what is irrelevant). Bearing in mind that initial rejection occurs with probability bounded away from 1 by Lemma 14, it follows that the total variation distance between the output of REG^* and the uniform distribution is $o(1)$.

It is easy to see that uniformly generating a pairing $P \in \Phi$ takes $O(dn)$ steps, and so does verification of $P \in \mathcal{A}_\gamma$ with $\gamma = 1$, when using appropriate data structures. By Lemma 14, $P \in \mathcal{A}_\gamma$ with probability $1/2 + o(1)$, and so it takes $O(dn)$ steps in expectation until a random $P \in \mathcal{A}_\gamma$ is generated. The total number of switching steps in all phases is bounded by $O(B_L + B_D + B_T) = O(d^2 + \log n)$ in expectation and each switching step takes $O(1)$ unit of time. Thus, the total time complexity of REG^* is $O(dn + d^2 + \log n) = O(dn)$ in expectation. ■

Now we turn to REG . We first prove the uniformity of the output distribution.

Proof of Theorem 1. The uniform distribution of the output of REG is guaranteed by Lemmas 5 and 19, since the parameters $\rho_\tau(i)$ are set for REG , just after Lemma 7, using a solution $(\rho_\tau^*(i), x_i^*)$ of the system (13)–(17). (Recall that if the system had no solution, phase 3 was made redundant by requiring $D(P) = 0$ in the definition of \mathcal{A}_γ in Section 8.) ■

Finally, we estimate the time complexity of REG . Implementing a switching step as described in Section 3 would apparently require knowledge of $f_\tau(P)$ and $b_\alpha(P)$. However, we first observe that there is no need to compute $f_\tau(P)$. This number is only used to compute the f-rejection probability at substep (iii) of a switching step. Take the type I switching in phase 2 as an example. Rather than computing $f_I(P)$, we can just randomly pick a double edge and label its ends, and then uniformly at random pick two pairs, with repetition, from the total $M_1/2 - 2i$ pairs other than those contained in a double edge, and then randomly label their ends. The total number of choices is exactly $\overline{m}_I(i)$. Perform the switching if it is valid, and perform an f-rejection otherwise. The probability of an f-rejection is then exactly $1 - f_I(P)/\overline{m}_I(i)$. It is easy to see that this holds for all other types of switchings, by the way we defined each $\overline{m}_\tau(i)$.

Proof of Theorem 2. We have shown that generating a pairing $P \in \Phi$ uniformly at random takes $O(dn)$ steps. By Lemma 14, $P \in \mathcal{A}_\gamma$ with probability $\gamma/(1 + \gamma) + o(1)$, and so it takes $O(dn/\gamma)$ steps in expectation until a random $P \in \mathcal{A}_\gamma$ is generated. To set the parameters $\rho_\tau(i)$, we need to compute a solution to system (13)–(17). Using the computation

scheme in Lemma 7, the computation time is $O(B_D) = O(d^2)$, as remarked below Lemma 7.

To generate a random graph, we repeat *REG* until there is a run in which rejection does not occur. By Lemmas 17, 20, 9, 12 and 13, the probability that a rejection occurs during any of the three phases is $o(1)$. Thus, it is sufficient to bound the time complexity of each phase in one run of *REG*. Each phase consists of a sequence of switching steps, and as observed above, we only need to consider the computation time of $b_\alpha(P)$. Phase 3 is the most crucial since it runs for the greatest number, $O(d^2)$, of switching steps in expectation. The same ideas suffice easily for the other phases. An implementation that does the job is described in [12, Theorem 4]. Briefly, it goes as follows. Data structures are maintained that require an initial computation time of $O(nd^3)$ for the input pairing. These data structures record, for instance, how many 3-paths join each given pair of vertices. (Note that the number of nonzero instances of this is $O(nd^3)$.) They also record how many 3-paths join a pair of vertices using a given point, and some other counts relating to 2-paths, 3-paths starting with a double edge, *pairs* of 3-paths joining each two given vertices, and similar. These data structures can be initialised in time $O(nd^3)$ by starting at each vertex in turn and investigating each 3-path leading from it. Moreover, they can be updated in time $O(d^2)$ for each switching step. This is because each switching step alters $O(1)$ pairs, each of which would affect $O(d^2)$ of the numbers being recorded (e.g. the number of 3-paths between two vertices). As the total number of switching steps in this phase is $O(d^2)$ in expectation, this requires $O(d^4)$ steps in expectation throughout all the switchings of the phase. These data structures are used for implementing a scheme explained in [12] to calculate the number of switchings of class A that could produce the pairing P' that was actually produced in a switching step. For a pair of 2-paths not to appear as those in the right hand side of Figure 2, there must be some “coincidence” occurring: either one of the edges shown is a double edge, or there is a coincidence of vertices, or some edge joins u_2 to v_2 , or similar. The number of pairs of 2-paths satisfying any prescribed set of coincidences can be calculated using the data structures, and then one proceeds by inclusion-exclusion. For instance (one of the “hardest” cases) is when both u_2v_2 and u_3v_3 are edges. To count these configurations, we count pairs of 3-paths joining u_1 to v_1 . Instances where these two 3-paths intersect accidentally are accommodated as part of the whole inclusion-exclusion scheme. A bounded number of cases need to be considered, and each can be calculated in constant time from the appropriate data structures. (A very large bound on the number of cases is supplied in [12], but actually they can be described by approximately 20 different graphs showing the way that two 2-paths can intersect or have adjacent corresponding pairs of vertices, not counting the options for where double edges may occur among the edges of the 2-paths.) As a result, the most time-consuming part is the initialisation of the data structure, which takes $O(nd^3)$ time. Maintaining the data structure in the subsequent switching steps takes only $O(d^4) = o(nd^3)$ time. ■

Computational note. If the constraints in Lemma 7 are not satisfied, the algorithm terminates whenever the number of double edges is greater than 0, according to the definition of \mathcal{A}_γ in Section 8. For practical purposes it may be of interest to know when they cannot be satisfied for any choice of $\gamma > 0$ and ϵ , as the algorithm then becomes equivalent to *DEG* and it may take many repetitions before a successful output occurs. We note that, for $d > 3$,

the equation

$$4(d-1)^2(d-2)(d-3) = dn(dn - 7d^2 + 7d - 10)$$

is quadratic in n with a unique positive solution:

$$x(d) := \frac{7}{2}d - \frac{7}{2} + 5/d + (1/2d)\sqrt{65d^4 - 210d^3 + 461d^2 - 412d + 196}.$$

Letting $x(d)$ denote that solution, there will clearly exist a choice of ϵ and γ satisfying (23) and (24), provided that $n > x(d)$. This bound and (25) are both satisfied if n is at least 69, 64, 64, 66, 69 for $d = 4, 5, 6, 7, 8$ respectively. For such n , the extra power of the algorithm comes into play. Of course, the lower bound is asymptotically linear in d .

References

- [1] A. Békéssy, P. Békéssy and J. Komlós, Asymptotic enumeration of regular matrices, *Studia Sci. Math. Hungar.* **7** (1972), 343–353.
- [2] M. Bayati, J.H. Kim, and A. Saberi, A sequential algorithm for generating random graphs, *Algorithmica* **58** (2010), 860–910.
- [3] E.A. Bender and E.R. Canfield, The asymptotic number of labeled graphs with given degree sequences, *J. Combin. Theory, Ser. A* **24** (1978), 296–307.
- [4] B. Bollobás, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, *European J. Combin.* **1** (1980), 311–316.
- [5] C. Cooper, M. Dyer, and C. Greenhill, Sampling regular graphs and a peer-to-peer network, *Combin. Probab. Comput.*, **16**(4):557–593, 2007; Corrigendum: arXiv:1203.6111.
- [6] C. Greenhill, The switch markov chain for sampling irregular graphs, *Proc. SODA* (2014), 1564–1572.
- [7] C. Greenhill and B. McKay, Asymptotic enumeration of sparse multigraphs with given degrees, *SIAM J. Discrete Math.* **27** (2013), 2064–2089.
- [8] S. Janson, The probability that a random multigraph is simple, *Combin. Probab. Comput.*, **18** (2009), 205–225.
- [9] M. Jerrum and A. Sinclair, Fast uniform generation of regular graphs, *Theoret. Comput. Sci.* **73** (1990), 91–100.
- [10] R. Kannan, P. Tetali and S. Vempala, Simple Markov-chain algorithms for generating bipartite graphs and tournaments, *Random Structures Algorithms* **14** (1999), 293–308.
- [11] J.H. Kim and V.H. Vu, Generating random regular graphs, *Combinatorica* **26** (2006), 683–708.
- [12] B.D. McKay and N.C. Wormald, Uniform generation of random regular graphs of moderate degree, *J. Algorithms* **11** (1990), 52–67.

- [13] B.D. McKay and N.C. Wormald, Uniform generation of random Latin rectangles, *J. Combin. Math. Combin. Comput.* **9** (1991), 179–186.
- [14] B.D. McKay and N.C. Wormald, Asymptotic enumeration by degree sequence of graphs with degrees $o(\sqrt{n})$, *Combinatorica* **11** (1991), 369–382.
- [15] J. Propp and D. Wilson, Coupling from the past: a user’s guide, *Microsurveys in Discrete Probability* **41** (1998), 181–192.
- [16] A. Steger and N.C. Wormald, Generating random regular graphs quickly, *Combin. Probab. Comput.* **8** (1999), 377–396.
- [17] G. Tinhofer, On the generation of random graphs with given properties and known distribution, *Appl. Comput. Sci., Ber. Prakt. Inf.* **13** (1979), 265–297.
- [18] N.C. Wormald, Generating random regular graphs, *J. Algorithms* **5** (1984), 247–280.
- [19] J.Y. Zhao, Expand and Contract: Sampling graphs with given degrees and other combinatorial families, arXiv:1308.6627.