Decycling Numbers of Random Regular Graphs

Sheng Bau,[1],* Nicholas C. Wormald,[2] Sanming Zhou[2],†
[1] *School of Mathematics, Statistics and Information Technology, University of Natal, Pietermaritzburg, South Africa; e-mail: bausheng@nu.ac.za*
[2] *Department of Mathematics and Statistics, University of Melbourne, VIC 3010, Australia; e-mail: nick@ms.unimelb.edu.au; smzhou@ms.unimelb.edu.au*

Received 30 November 2001; accepted 20 July 2002

DOI 10.1002/rsa.10069

ABSTRACT: The decycling number $\phi(G)$ of a graph G is the smallest number of vertices which can be removed from G so that the resultant graph contains no cycles. In this paper, we study the decycling numbers of random regular graphs. For a random cubic graph G of order n, we prove that $\phi(G) = \lceil n/4 + 1/2 \rceil$ holds asymptotically almost surely. This is the result of executing a greedy algorithm for decycling G making use of a randomly chosen Hamilton cycle. For a general random d-regular graph G of order n, where $d \geq 4$, we prove that $\phi(G)/n$ can be bounded below and above asymptotically almost surely by certain constants $b(d)$ and $B(d)$, depending solely on d, which are determined by solving, respectively, an algebraic equation and a system of differential equations. © 2002 Wiley Periodicals, Inc. Random Struct. Alg., 21: 397–413, 2002

1. INTRODUCTION

The problem of eliminating all cycles in a graph by removing a set of vertices goes back at least to the work of Kirchhoff [14] on spanning trees. In the literature such a set is called

Correspondence to: N. C. Wormald

*The work of this author was supported by The University of Natal, South Africa, and The University of Melbourne, Australia.

†The work of the second and third authors was supported by a large grant from the Australian Research Council.

a feedback vertex set, or following [5], a decycling set, of the graph. Formally, for a graph G, a subset $S \subseteq V(G)$ is said to be a *decycling set* of G if $G - S$ is acyclic. The minimum cardinality of a decycling set of G is defined to be the *decycling number* of G, denoted by $\phi(G)$ in this paper. Unlike its counterpart problem of destroying cycles by removing edges, the problem of determining the decycling number does not have a simple solution. The corresponding decision problem has long been known to be NP-complete, as shown in [12]. (See Problem 7 on the feedback node set in the main theorem of [12].) Moreover, the same is true even we restrict to the families of planar graphs, bipartite graphs and perfect graphs. On the other hand, the problem is known to be polynomial for various other families, including cubic graphs [15, 24], permutation graphs [16], and interval and comparability graphs [17]. These results naturally suggest further investigations for good bounds on the parameter and for exact results when possible. Partial results on exact values or bounds on decycling number have been obtained for cubes and grids in [4] and [5]. For a cubic graph G of order n, it is not difficult to see that $\phi(G) \geq \lceil n/4 + 1/2 \rceil$, and this follows from a general lower bound given in [5]. Indeed, if G is d-regular, then, since the decycling set is incident with at most $d\phi(G)$ edges, and its removal leaves a forest containing at most $n - \phi(G) - 1$ edges, it follows that

$$\phi(G) \geq (n(d/2 - 1) + 1)/(d - 1). \tag{1}$$

The gap between this lower bound and the actual value of $\phi(G)$ can be arbitrarily large when n is large. In fact, Bondy, Hopkins, and Staton [7] constructed a class of cubic graphs with decycling number $\lceil 3n/8 + 1/4 \rceil$. This class contains graphs obtained from taking cubic trees and replacing every vertex of degree 3 by a triangle and attaching K_4 with one edge subdivided at each vertex of degree 1. Except for K_4, graphs in this class are not 2-connected. Yet there are 3-connected cubic graphs with decycling number just one-third of their order—every cubic graph constructed by replacing each vertex of any 3-connected cubic graph by a triangle has this property. Note that these constructions make use of triangles. For a connected cubic graph G of order n with no triangles, it was proved in [29] that $\phi(G) \leq \lceil n/3 \rceil$, and this settled a conjecture of [7] in the affirmative. For more results concerning the decycling number, the reader is referred to Alon et al. [1], Bau and Beineke [3], Bafna et al. [2] and Liu and Zhao [18]. The related question of finding a maximum induced tree has also been considered; see Erdős et al. [11] for example.

In this paper we will study the decycling numbers of random regular graphs. We first give a simple algorithm which greedily generates a decycling set of a random cubic graph G, given a Hamilton cycle of G. Based on this, we show that asymptotically almost surely the decycling number of such a graph is roughly one quarter of its order. (For a sequence of probability spaces Ω_n, $n \geq 1$, an event A_n of Ω_n occurs *asymptotically almost surely,* or *a.a.s.* for brevity, if $\lim_{n\to\infty} \mathbf{P}\{A_n\} = 1$. Here and in the following we use $\mathbf{P}$ to denote the probability, and we will use $\mathbf{E}$ to denote the expectation.) Our first main result is as follows.

Theorem 1.1. *For a random cubic graph G of order n, a.a.s.*

$$\phi(G) = \left\lceil \frac{n}{4} + \frac{1}{2} \right\rceil. \tag{2}$$

TABLE 1. Lower and Upper Bounds

d	$b(d)$	$B(d)$
4	1/3	0.3787
5	0.3786	0.4512
6	0.4232	0.5043
7	0.4610	0.5459
8	0.4932	0.5800
9	0.5210	0.6085
10	0.5453	0.6328

This shows that the somewhat trivial lower bound $\lceil n/4 + 1/2 \rceil$ a.a.s. gives the correct value of the decycling number in the random case, which settles in some asymptotic sense the problem of [3] asking for the cubic graphs achieving this lower bound. This is in stark contrast with the deterministic bounds for cubic graphs mentioned above. The polynomial time algorithm given in [15] for finding the decycling number of a cubic graph seems to offer no help for analyzing the decycling number in the random case, since it makes use of complicated arguments involving relations with the maximum genus, apart from other things.

We extended the algorithm for random cubic graphs to random d-regular graphs for larger d. A description of this algorithm is given at the beginning of Section 4. However, this algorithm gives a weaker bound than a different algorithm, based on the uniform model, when $d \geq 4$. We will give the latter algorithm in Section 4, and by using a differential equation method we will prove the following theorem.

Theorem 1.2. *Let $d \geq 4$. For a random d-regular graph G of order n, a.a.s.*

$$b(d) \leq \frac{\phi(G)}{n} \leq B(d) \tag{3}$$

where b(d) and B(d) are constants given in Table 1 for small d.

The values of $b(d)$ and $B(d)$ can be obtained by solving an algebraic equation for $b(d)$ and a system of differential equations to find $B(d)$; see Sections 5 and 4 for details.

Table 1 lists values of $b(d)$ and $B(d)$ obtained by numerical solution methods for the first a few integers $d \geq 4$. Note that the trivial bound $b(d) \geq (d - 2)/2(d - 1)$, which follows from (1), is the best we have for $d = 4$.

The graph-theoretic notation and terminology used in the paper will in general follow that of [9]. For concepts and notation of random graph processes and probabilistic methods, the reader is referred to the survey paper [27]; for the differential equation method, see [26].

2. PRELIMINARIES

In the following we will use $\mathscr{G}_{n,d}$ to denote the uniform space of d-regular graphs on the set $[n] = \{1, 2, \cdots, n\}$ of n vertices, where, of course, dn is required to be even. In particular, $\mathscr{G}_{n,1}$ is the uniform space of perfect matchings on $[n]$ when n is even. A method of sampling from $\mathscr{G}_{n,d}$ is to use the following pairing model introduced by Bollobás [6], or a related model of others (see [27, Section 2.1] for a brief history about this model). Suppose that dn is even and $d \geq 1$. Consider a set of *dn points* partitioned into *n cells* $v_1, v_2, \ldots, v_n$ each containing d points. A perfect matching of these dn points into $dn/2$ pairs is called a *pairing.* A pairing P induces a multigraph $G(P)$ in which the vertices are the cells and each pair $\{x, y\} \in P$ gives rise to one edge joining the cell containing x and the cell containing y. (When x and y are in the same cell, this will be a loop.) We may assume that the points are elements of $[n] \times [d]$ so that $G(P)$ is induced by a projection onto $[n]$. Since each simple graph corresponds to precisely $(d!)^n$ pairings, a regular graph can be chosen uniform at random (u.a.r.) by choosing a pairing P u.a.r. and accepting $G(P)$ if it has no loops or multiple edges. For later reference, we denote the uniform probability space of these pairings by $\mathscr{P}_{n,d}$.

In order to achieve our main result for the case of cubic graphs, the pairing model is not enough, and we rely on more advanced theory. Let $\mathscr{P}_n$ and $\mathscr{Q}_n$ be two discrete probability spaces over the same underlying set for each $n \geq 1$. The sequences of spaces $\{\mathscr{P}_n\}$ and $\{\mathscr{Q}_n\}$ are said to be *contiguous,* denoted $\mathscr{P}_n \approx \mathscr{Q}_n$, if any sequence of events $A_n (n \geq 1)$ occurs a.a.s. in $\{\mathscr{P}_n\}$ if and only if it occurs a.a.s. in $\{\mathscr{Q}_n\}$. In this case for simplicity we also say that the spaces $\mathscr{P}_n$ and $\mathscr{Q}_n$ are contiguous. For two probability spaces $\mathscr{P}$, $\mathscr{Q}$ of random graphs on the same vertex set, as in [27] define the *sum* $\mathscr{P} + \mathscr{Q}$ to be the space whose elements are determined by the random multigraphs $G \cup H$ (called the *superposition* of G and H), where $G \in \mathscr{P}$ and $H \in \mathscr{Q}$ are generated independently. Define the *graph-restricted sum* $\mathscr{P} \oplus \mathscr{Q}$ to be the space which is the restriction of $\mathscr{P} + \mathscr{Q}$ to simple graphs. In order to maintain identical underlying sets for spaces that are to be related, the sum space $\mathscr{P} + \mathscr{Q}$ is extended to include all d-regular multigraphs on the same vertex set (where, as usual, a loop contributes 2 to the degree of a vertex), with all multigraphs not already appearing given probability 0. Similarly, $\mathscr{P} \oplus \mathscr{Q}$ is extended to include the underlying set of $\mathscr{G}_{n,d}$. The operations $+$ and $\oplus$ are clearly commutative and associative. Hence, for k spaces $\mathscr{P}_i$ on the same vertex set, the meaning of $\mathscr{P}_1 \oplus \cdots \oplus \mathscr{P}_k$ is unambiguous. In particular, we will use $k\mathscr{P}$ to denote the graph-restricted sum of k copies of a random graph space $\mathscr{P}$.

The proof of Theorem 1.1 will be based on an algorithm for finding a decycling set in a random cubic graph. In turn, this algorithm relies on a special case (namely, $d = 3$) of the fundamental result implicitly proved by Robinson and Wormald [22], which asserts that for $d \geq 3$ a random d-regular graph with an even number of vertices is contiguous to the superposition of a random Hamilton cycle and $d - 2$ random perfect matchings. See also [27]. Define $\mathscr{H}_n$ to be the uniform space of random Hamilton cycles on the same vertex set as $\mathscr{G}_{n,d}$. The following result is a special case of a general result which was implied by the proofs in [22] and stated explicitly in [13] and [27, Corollary 4.17].

Theorem 2.1. *Let $d \geq 3$ and n be even. Then*

$$\mathscr{G}_{n,d} \approx \mathscr{H}_n \oplus (d-2)\mathscr{G}_{n,1}.$$

By definition, if S is a decycling set of a graph G, then the subgraph $G - S$ of G induced by $V(G)\backslash S$ is a forest, and vice versa. So the problem of finding the decycling number is equivalent to that of finding the maximum number of vertices which induce a forest. The sum of these two numbers is equal to n.

3. RANDOM CUBIC GRAPHS

In this section we will study the decycling number of a random cubic graph.

Proof of Theorem 1.1. By Theorem 2.1, $G \in \mathscr{G}_{n,3}$ is contiguous to the graph-restricted superposition of a random Hamilton cycle H and a random perfect matching M. We will work with this sum space. The edges in M will be called the *matching edges.* We may suppose the vertices of G are labeled $1, 2, \dots, n$ around H, so that $H = (1, 2, \dots, n, 1)$. We first give a greedy algorithm for finding a decycling set of G based on the following very simple idea: Start from 1, walk along H and delete the vertex being visited if it creates a cycle when added to the undeleted vertices. The algorithm is carried out simultaneously with generating the perfect matching M. When a vertex i is visited in the walk, the "direction" of the incident matching edge is first revealed—whether it goes *forwards* to the vertices $\{i + 1, \dots, n\}$ or *backwards* to the vertices already visited—by generating it at random with the correct probability. Only if it is a backward edge is the other end of the edge then chosen; otherwise the next vertex along H is visited. (Of course, if a matching edge goes backwards, then it can only join to an undeleted vertex.) This is an instance of the "method of deferred decisions": one aspect of the random edge is determined (its direction) while the choice of the other end of the edge is deferred. Consequently, at any point, there are some vertices which have already been visited but are still unmatched. Suppose there are k such vertices when the walk reaches the vertex i. At this point, the distribution of the edges matching them is that of a uniformly distributed perfect matching, subject to the condition that these k vertices are precisely the ones in the set $\{1, \dots, i - 1\}$ which match to vertices in the set $\{i, \dots, n\}$. The set of vertices matching them will consequently be a subset of $\{i, \dots, n\}$ chosen uniformly at random. It follows easily that

$$\text{the probability that the matching edge at } i \text{ goes backwards is } k/(n - i + 1). \tag{4}$$

In the case that it is a backward edge, the other end (call it j) of the matching edge is chosen uniformly at random from the k unmatched vertices in $\{1, \dots, i - 1\}$. The vertex i is deleted if and only if j is in the *latest* component of the forest generated by undeleted vertices up to $i - 1$, that is, the component of this forest containing $i - 1$. It is easy to see that this process generates the final matching uniformly at random. (If any matching edge corresponds to an edge of H, we can start the process again; the probability that this never happens is asymptotic to a nonzero constant—see [27].)

The algorithm which generates the random matching and simultaneously the decycling set is as follows.

Algorithm Cubic

Input An even integer $n \geq 4$.

Output A random matching M and a decycling set S of the random cubic graph G which is the union of M with the Hamilton cycle $H = (1, 2, \dots, n, 1)$.

1. Set $S_1 = M_1 = \emptyset$ and let G_1 be the empty graph with vertex set $\{1\}$. Set $i = 2$.
2. Decide whether the matching edge at i goes backwards or forwards. The probability of the former is equal to the number of unmatched vertices in $[i - 1]$ divided by $n - i + 1$.

2a. In the former case the vertex which matches i is chosen u.a.r. from those available, and we add the corresponding matching edge to M_{i-1} to form M_i. If adding this edge and the edge $\{i - 1, i\}$ (plus $\{n, 1\}$ if $i = n$) to G_{i-1} creates no cycle, then let G_i be the graph obtained this way and set $S_i = S_{i-1}$; otherwise let $G_i = G_{i-1}$ and $S_i = S_{i-1} \cup \{i\}$.

2b. In the latter case, leave the vertex i unmatched. Set $M_i = M_{i-1}$, $S_i = S_{i-1}$, and let G_i be the graph obtained by adding the edge $\{i - 1, i\}$ to G_{i-1}.

3. If $i = n$ then stop and output $S := S_n$ and $M := M_n$; otherwise set $i = i + 1$ and go to Step 2.

One can see that G_i is the subgraph of G induced by $[i]\backslash S_i$. Alternatively, it consists of the subgraph of H induced by $[i]\backslash S_i$ and the matching edges at these vertices. From the algorithm it is obvious that G_i contains no cycles; in particular this applies to $G_n = G - S_n$ and hence S_n is a decycling set of G. In Step 2a, provided $i < n$, a cycle is created if and only if $i - 1 \notin S_{i-1}$ and the vertex to be matched with i lies in the latest component of G_{i-1}. (If $i - 1 \in S_{i-1}$, we regard the latest component as being empty.) We will next show the less obvious fact that a.a.s. S_n has cardinality no more than $n/4 + 1$.

Each vertex i is included in S_i only when the matching edge at i is joined to a vertex in the latest component of G_{i-1}. From this one can see that each S_i, $1 \le i \le n - 1$, is an independent set of G. Moreover, the only possible edge between vertices of $S = S_n$ is the edge joining $n - 1$ and n. This is a special case because the edge joining n to 1 causes the matching edge from vertex n to create a cycle if it joins to the component containing the vertex 1. In this case, n is placed into S even when $n - 1$ is in S. Hence there are at least $3|S| - 1$ edges incident with vertices in S. The major part of the proof is to show that a.a.s. the subgraph G_n of G is connected (and hence is a tree). Once this is achieved, then counting the total number of edges of G gives $3n/2 \ge (3|S| - 1) + (n - |S| - 1)$, implying $|S| \le n/4 + 1$ and hence $\phi(G) \le n/4 + 1$ holds a.a.s. But $\lceil n/4 + 1/2 \rceil$ is a lower bound for $\phi(G)$, as mentioned earlier, so Theorem 1.1 follows since $\lceil n/4 + 1/2 \rceil = \lfloor n/4 + 1 \rfloor$ for even integers n.

The algorithm is well defined when G is not necessarily a graph; i.e., we work with multigraphs. It is sufficient to show that G_n is a.a.s. connected in the multigraph setting, since the probability that G_n is a graph is asymptotically constant (see [27, Proof of Lemma 4.14]). The proof has similarities with the proof that a random d-process a.a.s. results in a connected graph [23]. First fix an integer $K \ge 3$. Define a vertex $i < n$ to be *special* if $i \in S$ and the latest component of G_{i-1} has at most K vertices not yet matched. Let $j = \lceil n^{1/3} \rceil$. We first prove the following.

Claim 1. *The number of special vertices in* $[n - j]$ *is a.a.s.* $O(\log^3 n)$.

Proof. As in the derivation of (4), if the latest component of G_{i-1} has at most K unmatched vertices, the probability that one of them is chosen to be matched in Step 2a of the algorithm is at most $K/(n - i)$, and this is independent of the number of special vertices chosen previously. Thus, for any integer k with $0 \le k < \frac{2}{3} \log_2 n$, each vertex i in the interval $I_k = \{i : n - 2^{k+1}j < i \le n - 2^k j\}$ has probability at most K/N of being special (independently of the earlier ones), where $N = 2^k j$. Hence the probability that some fixed set $R \subseteq I_k$ of vertices is special is at most $(K/N)^{|R|}$. Since $|I_k| = N$, the probability of more than $\log^2 n$ (note that the base of this logarithm is irrelevant) such vertices being special is at most

$$\binom{N}{\log^2 n}\left(\frac{K}{N}\right)^{\log^2 n} = o(1/n).$$

Hence a.a.s. none of the intervals I_k has more than $\log^2 n$ special vertices. Since there are $O(\log n)$ such intervals which altogether cover all vertices of $[n - j]$, Claim 1 follows. ■

Claim 2. *A.a.s. no vertex in the interval $n - j + 1, \dots, n$ is matched with another such vertex.*

Proof. This follows from the fact that the matching is chosen u.a.r. The probability that any particular vertex is matched with a vertex in this interval is at most $j/(n - 1)$, and so the expected number of such vertices from this interval is at most $j^2/(n - 1) = o(1)$. Markov's inequality completes the proof. ■

Armed with the two claims above, we are now ready to prove that G_n is a.a.s. connected. We first show that a.a.s. there exists no $i < n - j$ such that $i \in S$ and all vertices in the latest component of G_{i-1} are matched. For any such i and each component of G_i containing at most K unmatched vertices, if the last vertex in this component is t, then $t + 1$ is special. By Claim 1, there are at most $O(\log^3 n)$ such special vertices. Hence there are at most $O(\log^3 n)$ such components in the graphs G_{i-2} and G_{i-1}. Recalling that $K \geq 3$, we see that i cannot be as described above (i.e., creating a component of G_i with no unmatched vertices which remains a separate component of $G_{i'}$ *for* $i' > i$) unless both the edges of M incident with $i - 1$ and i join to components with less than K unmatched vertices. There being at most $O(K \log^3 n)$ unmatched vertices in such components, the probability of hitting them twice is $O(K^2 \log^6 n/(n - i)^2)$. Summing this over all $i \leq n - j$ gives $O(K^2 \log^6 n/j) = o(1)$, so the expected number of times that a component with no unmatched vertices is created in this fashion is $o(1)$. Again, by Markov's inequality we conclude that a.a.s. every component of G_{n-j} contains at least one unmatched vertex.

We finally turn to the vertices $n - j + 1, \dots, n$. Probabilities are conditioned on the occurrence of the events in Claims 1 and 2. Whether or not these hold is determined as soon as the vertex $n - j$ has been treated in the algorithm, since the event in Claim 2 holds if and only if the number of unmatched vertices at this point is j. We may then complete the perfect matching M on these vertices by matching them u.a.r. with the j previously unmatched vertices. Continuing the algorithm, it suffices to show that no subset R of the components of G_{n-j} remains isolated from the rest when the process terminates. We prove this by showing that the expected number of such subsets is $o(1)$. Without loss of generality, choose R so that it contains $t \leq j/2$ unmatched vertices in total. Let u denote the number of components of G_{n-j} having less than K unmatched vertices. Then $u = O(\log^3 n)$ by Claim 1, and the number of ways of choosing the components in R is $O(1)\binom{j/K+u}{t/K+u}$. The factor $O(1)$ accounts for the fact that the binomial coefficients are unimodal and symmetric about the center, and, even though $t/K < j/2K$, it may be that $t/K + u > (j/K + u)/2$. However, $u = O(\log^3 n)$.

Now consider the rest of the algorithm, and color the remaining vertices in $\{n - j + 1, \ldots, n\}$ *red* and *blue,* where a vertex is red if it matches to one in a component in R, and blue otherwise. A red vertex cannot be adjacent to blue ones on both of its sides along the Hamilton cycle H, for then a component in R would join to one outside R. So we can restrict the remaining part of the matching M to one in which the components (paths) of the subgraph of H induced by red vertices all have length at least 2. Call such components *red strings,* and denote by r the number of them. Then there are $\binom{t-r}{r}$ ways to choose the sequence of lengths of red strings, and $\binom{j-t+1}{r}$ ways to choose those of blue strings such that each has length at least 2. For a bound on the number of matchings under consideration, we multiply these two binomials together and divide by the number of ways of choosing the t red vertices, i.e., divide by $\binom{j}{t}$. Finally, we must sum over r. (We should also multiply by 2 to account for the two ways to interleave the red and blue strings.)

Now use

$$\binom{j-t+1}{r}\binom{t-r}{r} = \binom{j-t+1}{r}\binom{t-r}{t-2r} \le \binom{j}{t-r} \le \binom{j}{3t/4}$$

for $r > t/4$ and

$$\binom{j-t+1}{r}\binom{t-r}{r} \le \binom{j}{2r} \le \binom{j}{3t/4}$$

for $r \le t/4$. Use the estimates

$$\binom{j}{3t/4}\Big/\binom{j}{t} \le (t/j)^{t/4}$$

and

$$\binom{j/K+u}{t/K+u} \le j^u\binom{j/K}{t/K} \le (ej/t)^{t/K}$$

and multiply by t to account for summing over r. The result is $o(1)$. This completes the proof of Theorem 1.1. ∎

4. UPPER BOUNDS FOR $d \ge 4$

In this section we prove the upper bounds in Theorem 1.2.

The idea of Algorithm Cubic can be used to generate a decycling set of any random regular graph: Walk along the Hamilton cycle guaranteed by Theorem 2.1, and skip the vertex being visited when it creates a cycle with the unskipped vertices so far. After all vertices have been visited, the set of skipped vertices gives rise to a decycling set. As seen in the previous section for the special case where $d = 3$, to analyze this algorithm we have to keep track of the sizes of all components of the forest induced by the unskipped vertices. However, this seems to be an impossible task for larger d. Instead one might be tempted to use a relaxation of this algorithm which keeps track of some manageable information and places vertices in the decycling set whenever there is some possibility of forming a cycle otherwise. By using the standard differential equation method [26], we have analyzed the size of the decycling set generated by such an algorithm and obtained an asymptotic upper bound for $\phi(G)/n$. (The particular algorithm we used was the following, which the reader may like to verify is valid and can easily be implemented. The vertex w being visited is included in the decycling set if at least two matching edges at w go backwards, or if only one matching edge at w goes backwards but it creates a cycle with the vertices so far placed in the growing forest, or if the last vertex before w which had at least one backward matching edge was *not* placed in the decycling set.) However, for $d \geq 4$, we found that this upper bound is weaker than one obtained by using a natural algorithm on the pairing model. The purpose of this section is to prove this stronger upper bound, which is the right-hand side of (3). We will use the terminology set at the beginning of Section 2, and we will use the differential equation method for the main analysis of the algorithm.

Proof of Theorem 1.2. Recall that in the pairing model $\mathscr{P}_{n,d}$, as described in Section 2, the vertices of a random d-regular multigraph are cells each containing d points. We first give an algorithm which outputs a decycling set and an induced forest (usually a tree) simultaneously with generating a random pairing u.a.r. Simultaneous generation of random structures and running of algorithms has been used many times (for example, see [25]). The point which is paired with a particular point is called its *mate.* The method of deferred decisions will be used, to the extent that when vertices are added to the decycling set, the points which join back to the growing tree are determined, and their mates are chosen, but the mates of the other points are not determined. We will prove a property of the random pairing a.a.s. It then follows that the uniformly random d-regular graphs have the same property a.a.s. (see [6] or [27, Corollary 2.3]).

Algorithm Pairing

Input Integers $n \geq 5$ and $d \geq 4$ with dn even.

Output A random pairing P in $\mathscr{P}_{n,d}$, a decycling set S of the corresponding multigraph $G = G(P)$ and an induced forest T of G.

1. Set $S_1 = \varnothing$ and $P = \varnothing$, and let T_1 be the graph containing just the vertex 1. Set $t = 1$.
2. Let U_t denote the set of unpaired points in the vertices of T_t. Choose a point x u.a.r. from U_t. (If there is no such point, choose any vertex not in T_t or S_t and just add it to T_t to form T_{t+1}.) Select its mate, y, u.a.r. from the points in the vertices not in T_t or S_t. We will call these the *untreated* vertices. Let u denote the vertex containing y. For each of the other points $z_1, \ldots, z_{d-1} \neq y$ in u, decide whether the mate of z_i is in U_t or not. (This must be done with the correct probability, given that the pairing is uniform subject to all of U_t being paired with untreated points. This probability is estimated below.) If no mate of any z_i is in U_t, set $S_{t+1} = S_t$ and let T_{t+1} be the forest obtained from T_t by adding u together with the edge joining u and the vertex containing x. Otherwise, set $S_{t+1} = S_t \cup \{u\}$ and $T_{t+1} = T_t$ and select mates for those z_i which, as determined above, have mates in U_t. These are of course selected u.a.r. from $U_t \backslash \{x\}$. In the second case, the mates of those points which lie in the untreated vertices are left undetermined.

2. All new pairs determined in this step are added to the pairing P.
3. If $t + 1 = dn/2$ then stop and output $S = S_{dn/2}$ and $T = T_{dn/2}$ and the pairing P; otherwise set $t = t + 1$ and go to Step 2.

The algorithm is similar to that [10] for generating an independent dominating set of a random cubic graph (see [26] also). Let $X(t)$ denote the number of untreated vertices at time t. Let $Z(t)$ be the number of vertices of T_t, and $Y(t) = |U_t|$, the number of unpaired points in such vertices. During the algorithm the probability that one of the points in the vertex u being treated is paired with an unpaired point of T_t is in a general step (i.e., one far from the very end of the algorithm, by which we mean, say, $X(t) > \log n$) asymptotically $Y(t)/(X(t)d)$. This is because the pairing is uniform subject to all $Y(t)$ unpaired points in T_t being paired with the $X(t)d$ points in the untreated vertices. During the processing of the unpaired points in the vertex u, this probability changes only by $o(1)$. Hence the probability that u is added to the growing forest T_t is

$$P(t)^{d-1} + o(1), \qquad \text{where} \quad P(t) = 1 - \frac{Y(t)}{X(t)d}. \tag{5}$$

Consequently, the expected change in $Z(t)$ in one step is asymptotically $P(t)^{d-1}$.

Unlike many other problems of this type, we will need to examine the first ϵn steps of the algorithm separately. Note that, in the first step, $Y(1)$ becomes d. Until $Y(t)$ drops to 0, the growing forest will be a tree. It is this tree property we need to pay attention to, and which requires careful consideration of these early stages of the algorithm.

We will break the "time" t up into three intervals, for suitably small $\epsilon > 0$: $t < \epsilon n$, $\epsilon n < t < (\tau_0 - \epsilon)n$, where τ_0 will be defined later, and $t > (\tau_0 - \epsilon)n$. We examine these intervals in the natural order, but the reader interested in the definition of τ_0 and in the main part of the analysis may wish to skip to the main (middle) interval first.

For $t < \epsilon n$, the probability that the new vertex chosen is not added to the forest is at most $td/(n - t)$ by (5), since clearly $Y(t) \leq td$ and $X(t) \geq n - t$. If the new vertex is added to the forest, $Y(t)$ increases by $d - 2$. Otherwise, $Y(t)$ decreases by at most d. Standard arguments now show that a.a.s. $Y(t) \geq t/2$ for all $t < \epsilon n$. (We sketch the proof of this as follows. We claim that a.a.s., for all such t, the number of times the new vertex was not added to the forest, from step 1 up until step t, is at most $\frac{1}{3}t(d - 2)/d$. It then follows that $Y(t) \geq \frac{1}{3}t(d - 2)$. The claim can be verified by defining Q_t to be the indicator of the event that the new vertex is not added to the forest in step t, and then proceeding for t in three ranges. First, $\sum_{1 \leq t < n^{2/5}} \mathbf{E}Q_t = o(1)$, so by linearity of expectation and Markov's inequality, a.a.s. $Q_t = 0$ for all such t. Secondly, $\mathbf{E} \sum_{n^{2/5} \leq t \leq n^{2/3}} Q_t = O(n^{1/3})$. Standard large deviation inequalities such as [26, Lemma 4.2] (see also McDiarmid [19) show that a.a.s. this sum exceeds $n^{1/3}$ by at most $n^{1/3+\alpha}$ for all $\alpha > 0$. Hence a.a.s. $\sum_{n^{2/5} \leq t \leq n^{2/3}} Q_t = O(n^{1/3+\alpha})$. In this event, the claim holds for all t in this range. The interval $n^{2/3} < t \leq \epsilon n$ is similar, where the expected value is $O(\epsilon n)$ and the deviation above this is essentially a.a.s. $O(n^{1/2+\alpha})$. This gives the claim for sufficiently small $\epsilon > 0$.)

We may now assume that at the beginning of the second interval, when $t = \lfloor \epsilon n \rfloor$, we have $Y(t) > \epsilon/2$. In the analysis of the second interval, we assume $Y(t) > 0$ at each step. (When we come to apply a theorem approximating the process by differential equations, it is only deduced that the variables are well approximated by the solution of a differential equation until that solution approaches a boundary of a domain in which a variable which approximates $Y(t)$ is positive. Numerical computations reveal when this occurs. Until this point, we will only need to know about the behaviour of the process conditional upon $Y(t) > 0$.)

Since the forest is a tree, of the $Z(t)d$ points in its vertices, $2Z(t) + O(1)$ are used by the edges within the forest T_t, and $Y(t)$ are unpaired. So $(d - 2)Z(t) - Y(t)$ have been paired to vertices in S_t, which therefore has [up to additive error $O(1)$]

$$\begin{aligned} W(t) &= d(n - Z(t) - X(t)) - ((d - 2)Z(t) - Y(t)) \\ &= d(n - X(t) - 2Z(t)) + Y(t) + 2Z(t) \end{aligned}$$

unpaired points. In Step 2 of the algorithm, the point x counted by $Y(t)$ is used up. For each of the $d - 1$ points in u other than y, the probability that its mate is a point counted by $Y(t)$ is $1 - P(t) + o(1)$ [see (5)]. Hence the expected number of such points in u paired to T_t is $(d - 1)(1 - P(t)) + o(1)$. Thus, in the event that u is added to S_t [which happens with probability $1 - P(t)^{d-1} + o(1)$], the expected change in $Y(t)$ is [up to an additive term $o(1)$]

$$\begin{aligned} &(1 - P(t)^{d-1})\{-1 - (d - 1)(1 - P(t))/(1 - P(t)^{d-1})\} \\ &\quad = -1 + P(t)^{d-1} - (d - 1)(1 - P(t)). \end{aligned}$$

On the other hand, in the event that u is added to T_t [which happens with probability $P(t)^{d-1}$], the probability that any particular one of these $d - 1$ points is paired with one of the points counted by $X(t)$ [and not with one of the $W(t)$ unpaired points in S_t] is $(X(t)d - Y(t))/(X(t)d - Y(t) + W(t)) + o(1)$. [This is because all $Y(t)$ unpaired points in T_t must be paired with points in the untreated vertices u.a.r., leaving $X(t)d - Y(t)$ points among these vertices, plus the $W(t)$ unpaired points in $S(t)$, for a random perfect matching. Here we are assuming that, in this particular step, the denominator $X(t)d - Y(t) + W(t)$ is at least some constant times n, with the constant depending on ϵ. We discuss later why this can be assumed.] So the expected change in $Y(t)$ due to this event is [up to $+o(1)$]

$$\left\{(d - 1)\left(\frac{X(t)d - Y(t)}{X(t)d - Y(t) + W(t)}\right) - 1\right\}P(t)^{d-1}.$$

Putting together the expected change in $Y(t)$ in one step is

$$(d - 1)\left(\frac{X(t)d - Y(t)}{X(t)d - Y(t) + W(t)}\right)P(t)^{d-1} - (d - 1)(1 - P(t)) - 1. \tag{6}$$

On the other hand, the expected change in $X(t)$ per step is exactly -1, since each untreated vertex is used in every step. Recall that the expected change in Z per step is given in (5). We use $\tau = t/n$ to denote the "scaled time," and, as usual for the differential equation method, we use $x(\tau)$, $y(\tau)$, $z(\tau)$, and $w(\tau)$ to model $X(t)/n$, $Y(t)/n$, $Z(t)/n$, and $W(t)/n$, respectively. Then $P(t)$ and $W(t)/n$ can be modeled by $p(\tau) = 1 - y(\tau)/(x(\tau)d)$ and

$w(\tau) = d(1 - x(\tau) - 2z(\tau)) + y(\tau) + 2z(\tau)$, respectively. The expected changes in $X(t)$, $Y(t)$, and $Z(t)$ suggest the following system of differential equations:

$$\begin{cases} \dfrac{dx}{d\tau} = -1 \\ \dfrac{dy}{d\tau} = (d-1)\left(\dfrac{xd-y}{xd-y+w}\right)p^{d-1} - (d-1)(1-p)(1-p^{d-1}) - 1 \\ \dfrac{dz}{d\tau} = p^{d-1} \end{cases}$$

with initial conditions $x(0) = 1$, $y(0) = z(0) = 0$. These can be solved (numerically) up to the first positive τ for which $x = 0$ or $xd - y + w = 0$. We denote this value of τ by τ_0. By [26, Theorem 5.1] (see also [25] or the simplified version [28, Theorem 3]), the solutions $x(\tau)$, $y(\tau)$, $z(\tau)$ to these equations exist, and $X(t)$, $Y(t)$, and $Z(t)$ are a.a.s. approximated by $nx(t/n)$, $ny(t/n)$, and $nz(t/n)$, respectively, with error $o(n)$, for $t < (\tau_0 - \epsilon)n$. (We leave the precise application of these theorems, which is quite routine, to the reader.) Numerical solutions of the differential equations (which we performed for d up to 10) reveals that $x(\tau_0) = 0$ (to within the precision of the numerical computations). By restricting this second interval to $t < (\tau_0 - \epsilon)n$, we have ensured that the approximation is valid through this whole interval. The last interval, $t > (\tau_0 - \epsilon)n$, can only contain a negligible number of steps, so by ignoring it we do not lose anything significant.

Recall that $Z(t)$ is the number of vertices in the induced tree T_t. Now take ϵ very small and consider the end of the second interval. Then, for all $\epsilon > 0$, $z(\tau_0) - \epsilon$ is a.a.s. an asymptotic lower bound on the proportion of vertices in the induced tree, and hence an upper bound on $\phi(G)/n$ is a.a.s. $B(d) = 1 - z(\tau_0) + \epsilon$. Upper bounds $B(d)$ which hold a.a.s. on $1 - z(\tau_0)$ are listed for $4 \leq d \leq 10$ in the right column of Table 1 in the Introduction, as computed by numerical solution of the differential equations. This proves the upper bounds in Theorem 1.2. ∎

Note that, from this proof, it follows that there is a.a.s. an induced tree of the same asymptotic size as the induced forest found (at least, to the precision of the computations).

5. EXPECTED NUMBER OF TREES AND FORESTS

In this section we obtain the lower bounds in Theorem 1.2.

We consider first the expected number of induced trees of order k in the random graph $\mathcal{G}_{n,d}$, and then modify the calculations for induced forests. We use the pairing model described in Section 2, and thus consider a random pairing. Any property obeyed a.a.s. by the random pairing then carries over to $\mathcal{G}_{n,d}$ (see [27]). We calculate $\mathbf{E}X_k$, where X_k denotes the number of trees of order k in (the graph corresponding to) the pairing.

Suppose that T is an induced tree on k vertices the graph corresponding to the pairing. The vertices of T can be chosen in

$$\binom{n}{k}$$

ways. If the degrees of the vertices are $d_1, \ldots, d_k$ then T can be chosen in

$$\frac{(k-2)!}{\prod_{i=1}^{k}(d_i-1)!} \tag{7}$$

ways.

We pause here to justify this. It is well known in tree enumeration theory, but we will have a need to extend the result to a bound for forests. The simple explanation for (7) comes from the Prüfer sequence for the labeled tree, which comes from repeatedly deleting the lowest-labeled leaf and writing down the label *of its adjacent vertex.* Stop when there are only two vertices left. This means that all labels of nonleaves appear somewhere in the sequence, and hence the first vertex is the lowest label not appearing. Thus the first step of the deletion process can be reconstructed from the sequence, by attaching this leaf to the vertex whose number is first in the sequence. By induction, the whole tree can be reconstructed, so each sequence corresponds to at most one tree. Conversely, it is easy to see that each sequence of length $k-2$ from the k labels corresponds to a tree. Thus the number of trees on k vertices is k^{k-2}, but in particular, since the label of a vertex of degree j appears exactly $j-1$ times in the sequence, we obtain (7).

Once the tree T has been chosen, we may choose precisely which points in the pairing are used for the pairs corresponding to its edges. The d_i edges coming into a vertex can be mapped to points in the vertex in $d!/(d-d_i)!$ ways. Hence, collecting the factors above, the number of ways to choose all the pairs corresponding to edges of T is (with square brackets denoting coefficient extraction)

$$\begin{aligned}\binom{n}{k}(k-2)!\sum_{\substack{d_1,\ldots,d_k\\ 2k-2=\sum d_i}}\frac{d!}{(d-d_i)!(d_i-1)!} &= \binom{n}{k}(k-2)!\sum_{\substack{d_1,\ldots,d_k\\ 2k-2=\sum d_i}}\binom{d}{d_i}d_i\\ &= \binom{n}{k}(k-2)![x^{2k-2}]\left(\sum_{j=1}^{d}\binom{d}{j}jx^j\right)^k\\ &= \binom{n}{k}(k-2)![x^{2k-2}](g(x))^k \qquad (8)\end{aligned}$$

where

$$g(x) = xd(1+x)^{d-1}.$$

The standard way to estimate the coefficient in (8) is (see [20], for example) to observe that it is bounded above by

$$\alpha^{-2k+2}g(\alpha)^k \tag{9}$$

for all $\alpha > 0$. We may choose α so as to minimize this bound. Since all we require ultimately is an upper bound on $\mathbf{E}X_k$, this suffices for our purposes. In fact, it can be shown that using this bound results in the correct value of X_k to within a polynomial

TABLE 2. Upper Bounds on Size of Induced Trees and Forests, a.a.s.

d	κ for Tree	κ for Forest
5	0.6214756457	0.6215520592
6	0.5768963205	0.5775223167
7	0.5390900048	0.5402738418
8	0.5068847315	0.5085196796
9	0.4790661409	0.4810425927
10	0.4547283832	0.4569554491
11	0.4332035162	0.4356103248
12	0.4139905628	0.4165230118
13	0.3967060215	0.3993230883
14	0.3810506356	0.3837212484
15	0.3667868661	0.3694874682
16	0.3537233114	0.3564360217
17	0.3417036974	0.3444149357
18	0.3305989350	0.3332983963
19	0.3203012819	0.3229811686
20	0.3107199781	0.3133744218

factor, and hence we are not losing anything by this when the final result is considered. Differentiation of the logarithm of (9) [noting that the derivative of log $g(x)$ is $(xd + 1)/(x^2 + x)$] shows that the best α is $1/(d - 2)$. Thus (8) is bounded above by

$$\binom{n}{k} \frac{(k-2)!}{(d-2)^2} \rho^k, \qquad \text{where} \quad \rho = \frac{d(d-1)^{d-1}}{(d-2)^{d-2}}. \tag{10}$$

This is a bound on the number of ways to form the pairs which give the induced tree. The rest of the pairing is formed by first pairing each unused point in the tree's vertices with points in vertices not used by the tree—in $[dn - dk]_{dk-2k+2}$ ways, where $[r]_i$ denotes the falling factorial—and then choosing a perfect matching of all remaining points—in $M(nd - 2kd + 2k - 2)$ ways, where $M(2i) = (2i)!/(i!2^i)$. Multiplying these three factors together and dividing by $M(nd)$, and using Stirling's formula for factorials, neglecting polynomial factors [so that, for example, $M(nd) \approx (nd)^{nd/2}$] gives

$$\mathbf{E}X_k < (f(d, \kappa) + o(1))^n \tag{11}$$

where $\kappa = k/n$ and

$$f(d, \kappa) = \frac{\rho^\kappa d^{d-d\kappa}(1-\kappa)^{(d-1)(1-\kappa)}}{(d - 2d\kappa + 2\kappa)^{d/2 - d\kappa + \kappa} d^{d/2}}.$$

For $k = 3$, (1) implies that $\mathbf{E}X_k = 0$ if $\kappa > 3/4$. We find $f(3, 3/4) = \sqrt{2}$, so there is no new information gained for $d = 3$. Similarly, for $d = 4$, $f(4, 2/3) \approx 1.1906$, which permits many induced trees of size $2n/3$, corresponding to the upper bound obtained from (1). For larger d, we obtain new upper bounds on the size of the largest induced tree in $\mathscr{G}_{n,d}$ as shown in the middle column of Table 2, from the point κ at which $f(d, \kappa)$ dips below 1, since then $\mathbf{E}X_k$ becomes exponentially small.

However, the largest induced forest may be substantially larger than the largest induced tree. We next consider an upper bound on $\mathbf{E}Y_k$, where Y_k is the number of induced forests of k vertices in $G \in \mathcal{G}_{n,d}$. The bounds we will obtain are shown in the third column of Table 2.

Forests with given degree sequence and given number of vertices and edges do not seem to have been counted in the literature, though Britikov [8] found asymptotic formulae without regard to degree sequence. Without too much trouble we can obtain quite useful upper bounds. First, for simplicity, consider forests with no isolated vertices, with k vertices and j components. Consider constructing the Prüfer sequence for such a forest, as described above for a tree. This time, it is possible that the lowest-labeled leaf is adjacent to another leaf. If it is, do not write down the label of that leaf, but simply enter a special character (which we may call 0) and delete the two adjacent leaves. Again, stop when there are two (adjacent) vertices left. Then the length of the sequence is reduced by $j - 1$ as compared to the Prüfer sequence for a tree of k vertices, and so the number of sequences is

$$\frac{(k-j-1)!}{(j-1)!\,\Pi_{i=1}^{k}\,(d_i-1)!}.$$

Each sequence corresponds to at most $[k-1]_j$ forests, since the full identity of the forest is not revealed unless the labels of the vertices adjacent to the ones which enter "0"s are revealed. For these there are at most $[k-1]_j$ possibilities, as the lowest-labeled leaf in the forest cannot occur here. (This is quite an overcount, and is the only source of error in our overestimate for $\mathbf{E}Y_k$.)

Of course, if a forest has j_0 isolated vertices out of k, their labels can be chosen separately. Thus, since $\Sigma_i\, d_i = 2k - 2j$, an upper bound on the ways to choose pairs corresponding to a forest with k vertices, j components and with j_0 isolates is, corresponding to (10),

$$\binom{n}{k}\frac{(k-1)!}{j_0!(j-1)!}\,\rho^{k-j_0-j},$$

with ρ as in (10). Following the argument as for trees, this results in

$$\mathbf{E}Y_k < (h(d,\,\kappa,\,\lambda,\,\lambda_0) + o(1))^n \tag{12}$$

where $\kappa = k/n$ and

$$h(d,\,\kappa,\,\lambda,\,\lambda_0) = \frac{\rho^{\kappa-\lambda-\lambda_0}d^{d-d\kappa}(1-\kappa)^{(d-1)(1-\kappa)}}{\lambda_0^{\lambda_0}\lambda^{\lambda}(d-2d\kappa+2\kappa-2\lambda-2\lambda_0)^{d/2-d\kappa+\kappa-\lambda-\lambda_0}d^{d/2}}.$$

Fixing $\lambda + \lambda_0$, the first and second derivatives show that the maximum occurs at $\lambda_0 = \lambda$. Making this substitution, we find the second partial derivative of log h with respect to λ is

$$\frac{2(-2\kappa - d + 2d\kappa)}{\lambda(d - 2d\kappa + 2\kappa - 4\lambda)}.$$

The expression in the denominator occurs in the denominator of $h(d, \kappa)$ and has a physical meaning as the cardinality of a set of points. Hence the second derivative is negative, and putting $\partial \log h/\partial\lambda = 0$ will reveal the unique maximum. Solving this equation to yield λ_1 and solving $h(d, \kappa, \lambda_1, \lambda_1) = 1$ for κ (using Maple) gives the results in the third column of Table 2.

We conclude this section with our opinion on the question of whether the decycling number of random 4-regular graphs is a.a.s. equal to the bound given in (1). Our calculation above shows that the expected number of induced trees of the complementary size, $(2n - 1)/3$, is exponentially large.

Conjecture 5.1. *For $G \in \mathcal{G}_{n,4}$, a.a.s.*

$$\phi(G) = \left\lceil \frac{|G|}{3} + \frac{1}{3} \right\rceil.$$

REFERENCES

[1] N. Alon, D. Mubayi, and R. Thomas, Large induced forests in sparse graphs, J Graph Theory 38 (2001), 113–123.

[2] V. Bafna, P. Berman, and T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problems, SIAM J Discrete Math 12 (1999), 289–297.

[3] S. Bau and L. W. Beineke, The decycling number of graphs, Austr J Combinat 25 (2002), 285–298.

[4] S. Bau, L. W. Beineke, G.-M. Du, Z. S. Liu, and R. C. Vandell, Decycling cubes and grids, Utilitas Math 59 (2001), 129–137.

[5] L. W. Beineke and R. C. Vandell, Decycling graphs, J Graph Theory 25 (1997), 59–77.

[6] B. Bollobás, Random graphs, Academic Press, London, 1985.

[7] J. A. Bondy, G. Hopkins, and W. Staton, Lower bounds for induced forests in cubic graphs, Can Math Bull 30 (1987), 193–199.

[8] V. E. Britikov, Asymptotics of the number of forests made up of nonrooted trees (Russian), Mat Zametki 43 (1988), 672–684, 703 [transl Math Notes 43(5–6) (1988), 387–394].

[9] R. Diestel, Graph theory, Springer-Verlag, New York, 1997.

[10] W. Duckworth and N. C. Wormald, Minimum independent dominating sets of random cubic graphs, Random Struct Alg, in press.

[11] P. Erdős, M. Saks, and V. T. Sós, Maximum induced trees in graphs, J Combinat Theory 41B (1986), 61–79.

[12] R. M. Karp, "Reducibility among combinatorial problems," Complexity of computer computations, Eds. R. E. Miller and J. W. Thatcher, 85–103, Plenum Press, New York, London, 1972.

[13] J. H. Kim and N. C. Wormald, Random matchings which induce Hamilton cycles and hamiltonian decompositions of random regular graphs, J Combinat Theory Ser B 81 (2001), 20–44.

[14] G. Kirchhoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird, Ann Phys Chem 72 (1847), 497–508.

[15] D.-M. Li and Y.-P. Liu, A polynomial algorithm for finding the minimum feedback vertex set of a 3-regular simple graph, Acta Math Sci (Engl Ed) 19(4) (1999), 375–381.

[16] Y. D. Liang, On the feedback vertex set problem in permutation graphs, Inf Process Lett 52 (1994), 123–129.

[17] Y. D. Liang and M.-S. Chang, Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs, Acta Inf 34 (1997), 337–346.

[18] J.-P. Liu and C. Zhao, A new bound on the feedback vertex sets in cubic graphs, Discrete Math 148 (1996), 119–131.

[19] C. McDiarmid, "Concentration," Probabilistic methods for algorithmic discrete mathematics, Eds. M. Habib et al., 195–248, Springer-Verlag, Berlin, 1998.

[20] A. M. Odlyzko, "Asymptotic enumeration methods," Handbook of Combinatorics, Vol. II, Eds. R. L. Graham, M. Grotschel, and L. Lovasz, Elsevier, Amsterdam, 1995.

[21] R. W. Robinson and N. C. Wormald, Almost all cubic graphs are Hamiltonian, Random Struct Alg 3 (1992), 117–125.

[22] R. W. Robinson and N. C. Wormald, Almost all regular graphs are Hamiltonian, Random Struct Alg 5 (1994), 363–374.

[23] A. Ruciński and N. C. Wormald, Connectedness of graphs generated by a random d-process, J Austr Math Soc 72 (2002), 67–85.

[24] S. Ueno, Y. Kajitani, and S. Gotoh, On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three, Discrete Math 72 (1988), 355–360.

[25] N. C. Wormald, Differential equations for random processes and random graphs, Ann Appl Probab 5 (1995), 1217–1235.

[26] N. C. Wormald, "The differential equation method for random graph processes and greedy algorithms," Lectures on approximation and randomized algorithms, Eds. M. Karoński and H. J. Prömel, 73–155. PWN, Warsaw, 1999.

[27] N. C. Wormald, "Models of random regular graphs," Surveys in combinatorics, 1999, Eds. J. D. Lamb and D. A. Preece, London Mathematical Society Lecture Notes Series, Vol. 267, 239–298. Cambridge University Press, Cambridge, 1999.

[28] N. C. Wormald, Analysis of greedy algorithms on graphs with bounded degrees, Discrete Math, to appear.

[29] M. Zheng and X. Lu, On the maximum induced forests of a connected cubic graph without triangles, Discrete Math 85 (1990), 89–96.