Cleaning Random d-regular Graphs with Brushes Using a Degree-greedy Algorithm

Margaret-Ellen Messinger[1*], Paweł Prałat[1*], Richard J. Nowakowski[1⋆], and Nicholas Wormald[2⋆⋆]

[1] Department of Mathematics and Statistics,
Dalhousie University, Halifax NS, Canada,
{messnger, pralat, rjn}@mathstat.dal.ca
[2] Department of Combinatorics and Optimization,
Waterloo University, Waterloo ON, Canada,
nwormald@uwaterloo.ca

Abstract. In the recently introduced model for *cleaning* a graph with brushes, we use a degree-greedy algorithm to clean a random d-regular graph on n vertices (with dn even). We then use a differential equations method to find the (asymptotic) number of brushes needed to clean a random d-regular graph using this algorithm. As well as the case for general d, interesting results for specific values of d are examined. We also state various open problems.

Key words: cleaning process, random d-regular graphs, degree–greedy algorithm, differential equations method

1 Introduction

The cleaning model, introduced in [5, 6], considers a network of pipes that must be periodically cleaned of a contaminant that regenerates, for example, algae in water pipes. This is accomplished by having cleaning agents, colloquially, 'brushes', assigned to some vertices. To reduce the recontamination, when a vertex is 'cleaned', a brush must travel down each contaminated edge. Once a brush has traversed an edge, that edge has been *cleaned*. A graph G has been *cleaned* once every edge of G has been cleaned. McKeil [5] considered the model where more than one brush can travel down an edge and brushes can travel down cleaned edges. In [6] and this paper only one brush is allowed to travel along an edge and a brush is not allowed to travel down an edge that has already been cleaned.

Explicitly, every edge and vertex of a graph is initially *dirty* and a fixed number of brushes start on a set of vertices. At each step, a vertex v and all its incident edges which are dirty may be *cleaned* if there are at least as many

⋆ Research partially supported by NSERC and MITACS.
⋆⋆ Research partially supported by the Canada Research Chairs Program and NSERC.

brushes on v as there are incident dirty edges. When a vertex is cleaned, every incident dirty edge is traversed (i.e. cleaned) by one and only one brush, moreover, brushes cannot traverse a clean edge. This cleaning process is a combination of the chip-firing game and edge-searching on a simple finite graph. The approach in [6], and taken here, is that a *graph is cleaned when every vertex, and hence every edge, has been cleaned.* This may result in vertices with no dirty edges being cleaned in which case no brushes move but this approach simplified much of the analysis in [6]. See Figure 1 for an example of this cleaning process. The initial configuration has only 2 brushes, both at a. The solid edges are dirty and the dotted edges are clean. The circle indicates which vertex is cleaned next.

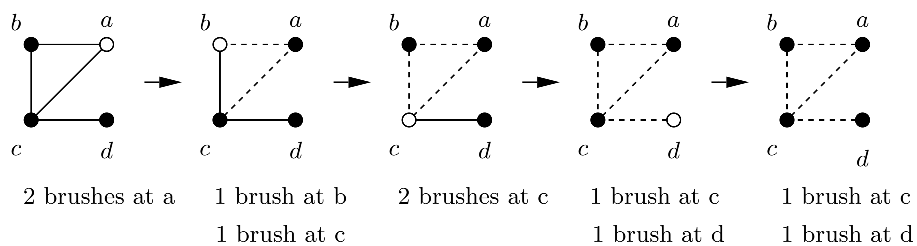


Fig. 1. An example of the cleaning process for graph G.

One condition that this model has, like chip-firing but not searching, is that the cleaning process is to be automatic, i.e. a union of 'vertex firing' sequences where each sequence cleans the graph, continuing on for the lifetime of the network. Therefore, the problems to solve are: firstly, a brush configuration and corresponding vertex firing sequence that cleans the graph; and secondly, having the final configuration of brushes be a starting configuration for another vertex firing sequence that also cleans the graph; and so on. In [6], we show that the final configuration of any cleaning sequence is a valid starting configuration of another cleaning sequence.

In this paper, we are interested in the asymptotic number of brushes needed to clean random d-regular (finite, simple) graphs. At one extreme, the graph could consist of disjoint copies of K_{d+1}. From [6], K_{d+1} requires essentially $d^2/4$ brushes so that the whole graph requires approximately $nd/4$. At the lower end, if d is even then a ring of bipartite graphs $K_{d/2,d/2}$ chained together (see Figure 2 for the case $d = 4$) require only $d^2/4$ brushes regardless of the number of vertices (by working around the ring). If d is odd then every vertex has at least one brush in either the original or final configuration (see [6] for more details) so that a graph on n vertices requires at least $n/2$ brushes.

We propose a linear time algorithm to clean d-regular graphs and an a.a.s. upper bound u_d on the number of brushes required by the algorithm. The asymp-

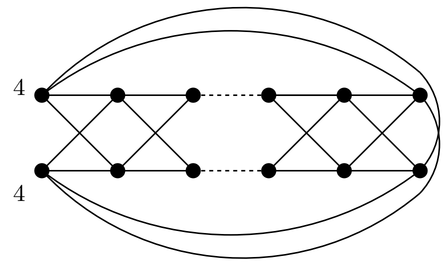


Fig. 2. An example of the cleaning process for a 4-regular graph requiring 8 brushes.

totically almost sure lower bound l_d follows from the fact that a.a.s. all sets of size $\lfloor n/2 \rfloor$ have at least l_d edges going to its complement.

In Section 4 we observe that if $d = 2$, then the brush number (asymptotically) is $(1+o(1))\log n$; for $d = 3$, the brush number is equal to $n/2+2$ a.a.s.; for $d = 4$, $(1+o(1))n/3$ brushes are enough to clean a graph a.a.s.; and for $d = 5$, roughly $0.644n$. For larger d, numerical evidence suggests that each brush on average cleans between 2 and 2.5 edges. In order to get an asymptotically almost sure upper bound on the brush number we use a degree-greedy algorithm, [9], to clean the graph and then use the differential equation method, studied in [12] to find the asymptotic number of brushes required.

In Section 2 we introduce the formal definitions for the cleaning process and a description of the pairing model of random regular graphs which is used instead of working directly with in the uniform probability space.

2 Definitions

The following cleaning algorithm and terminology was recently introduced in [6].

Formally, at each step t, $\omega_t(v)$ denotes the number of brushes at vertex v ($\omega_t : V \to \mathbb{N} \cup \{0\}$) and D_t denotes the set of dirty vertices. An edge $uv \in E$ is dirty if and only if both u and v are dirty: $\{u, v\} \subseteq D_t$. Finally, let $D_t(v)$ denote the number of dirty edges incident to v at step t:

$$D_t(v) = \begin{cases} |N(v) \cap D_t| & \text{if } \; v \in D_t \\ 0 & \text{otherwise.} \end{cases}$$

Definition 1. *The* ***cleaning process*** $\mathfrak{P}(G, \omega_0) = \{(\omega_t, D_t)\}_{t=0}^{T}$ *of an undirected graph* $G = (V, E)$ *with an* ***initial configuration of brushes*** ω_0 *is as follows:*

(0) *Initially, all vertices are dirty:* $D_0 = V$*; Set* $t := 0$

(1) *Let* α_{t+1} *be any vertex in* D_t *such that* $\omega_t(\alpha_{t+1}) \geq D_t(\alpha_{t+1})$*. If no such vertex exists, then stop the process* ($T = t$)*, return the* ***cleaning sequence*** $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$*, the* ***final set of dirty vertices*** D_T*, and the* ***final configuration of brushes*** ω_T

(2) *Clean α_{t+1} and all dirty incident edges by moving a brush from α_{t+1} to each dirty neighbour. More precisely, $D_{t+1} = D_t \setminus \{\alpha_{t+1}\}$, $\omega_{t+1}(\alpha_{t+1}) = \omega_t(\alpha_{t+1}) - D_t(\alpha_{t+1})$, and for every $v \in N(\alpha_{t+1}) \cap D_t$, $\omega_{t+1}(v) = \omega_t(v) + 1$ (the other values of ω_{t+1} remain the same as in ω_t)*

(3) *$t := t + 1$ and go back to* **(1)**

Note that for a graph G and initial configuration ω_0, the cleaning process can return different cleaning sequences and final configurations of brushes; consider, for example, an isolated edge uv and $\omega_0(u) = \omega_0(v) = 1$. It has been shown (see Theorem 2.1 in [6]), however, that the final set of dirty vertices is determined by G and ω_0. Thus, the following definition is natural.

Definition 2. *A graph $G = (V, E)$ **can be cleaned** by the initial configuration of brushes ω_0 if the cleaning process $\mathfrak{P}(G, \omega_0)$ returns an empty final set of dirty vertices ($D_T = \emptyset$).*

Let the brush number, $b(G)$, be the minimum number of brushes needed to clean G, that is,

$$b(G) = \min_{\omega_0 : V \to \mathbb{N} \cup \{0\}} \Big\{ \sum_{v \in V} \omega_0(v) : G \text{ can be cleaned by } \omega_0 \Big\}.$$

Similarly, $b_\alpha(G)$ is defined as the minimum number of brushes needed to clean G using the cleaning sequence α.

It is clear that for every cleaning sequence α, $b_\alpha(G) \geq b(G)$ and $b(G) = \min_\alpha b_\alpha(G)$. (The last relation can be used as an alternative definition of $b(G)$.) In general, it is difficult to find $b(G)$, but $b_\alpha(G)$ can be easily computed. For this, it seems better not to choose the function ω_0 in advance, but to run the cleaning process in some order, and compute the initial number of brushes needed to clean a vertex. We can adjust ω_0 along the way

$$\omega_0(\alpha_{t+1}) = \max\{2D_t(\alpha_{t+1}) - \deg(\alpha_{t+1}), 0\}, \qquad \text{for } t = 0, 1, \ldots, |V| - 1, \qquad (1)$$

since that is how much brushes we have to add over and above what we get for free.

Our results refer to the probability space of random d-regular graphs with uniform probability distribution. This space is denoted $\mathcal{G}_{n,d}$, and asymptotics (such as "asymptotically almost surely", which we abbreviate to a.a.s.) are for $n \to \infty$ with $d \geq 2$ fixed, and n even if d is odd.

Instead of working directly in the uniform probability space of random regular graphs on n vertices $\mathcal{G}_{n,d}$, we use the *pairing model* of random regular graphs, first introduced by Bollobás [1], which is described next. Suppose that dn is even, as in the case of random regular graphs, and consider dn points partitioned into n labeled buckets $v_1, v_2, \ldots, v_n$ of d points each. A *pairing* of these points is a perfect matching into $dn/2$ pairs. Given a pairing P, we may construct a multigraph $G(P)$, with loops allowed, as follows: the vertices are the buckets $v_1, v_2, \ldots, v_n$, and a pair $\{x, y\}$ in P corresponds to an edge $v_i v_j$ in $G(P)$ if x

and y are contained in the buckets v_i and v_j, respectively. It is an easy fact that the probability of a random pairing corresponding to a given simple graph G is independent of the graph, hence the restriction of the probability space of random pairings to simple graphs is precisely $\mathcal{G}_{n,d}$. Moreover, it is well known that a random pairing generates a simple graph with probability asymptotic to $e^{(1-d^2)/4}$ depending on d, so that any event holding a.a.s. over a probability space of random pairings also holds a.a.s. over the corresponding space $\mathcal{G}_{n,d}$. For this reason, asymptotic results over random pairings suffice for our purposes. The advantage of using this model is that the pairs may be chosen sequentially so that the next pair is chosen uniformly at random over the remaining (unchosen) points. For more information on this model, see [10].

3 Some lower bounds

When a graph G is cleaned using the cleaning process described in Definition 1, each edge of G is traversed exactly once and by exactly one brush.

Definition 3. *Given some initial configuration ω_0 of brushes, suppose $G = (V, E)$ admits a cleaning sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$ which cleans G. As each edge in G is traversed exactly once and by exactly one brush, an orientation of the edges of G is permitted such that for every $\alpha_i \alpha_j \in E(G)$, $\alpha_i \to \alpha_j$ if and only if $i < j$.*

The ***brush path*** *of a brush b is the oriented path formed by the set of edges cleaned by b (note that a vertex may not be repeated in a brush path). Then G can be decomposed into $b_\alpha(G)$ oriented brush paths (note that no brush can stay at its initial vertex in the minimal brush configuration).*

The minimum number of paths into which graph G can be decomposed yields a lower bound for $b(G)$; only a lower bound because some path decompositions would not be valid in the cleaning process. For example, K_4 can be decomposed into two edge-disjoint paths, but $b(K_4) = 4$.

Following Definitions 1 and 3, every vertex of odd degree in a graph G will be the endpoint of (at least) one brush path. This leads to a natural lower bound for $b(G)$ since any graph with d_o odd vertices, can be decomposed into a minimum of $d_o/2$ paths (see [6] for more details).

Theorem 1. *Given initial configuration ω_0, suppose G can be cleaned yielding final configuration ω_T. Then for every vertex v in G with odd degree, either $\omega_0(v) > 0$ or $\omega_T(v) > 0$. In particular, $b(G) \geq d_o(G)/2$ where $d_o(G)$ denotes a number of vertices of odd degree.*

The result can be improved a little if there is a lower bound on the vertex degrees (see Section 4.3 for details).

Another general lower bound for d-regular graphs can be obtained as follows. By [6, Theorem 3.2],

$$b(G) \geq \max_j \min_{S \subseteq V, |S|=j} \{jd - 2|E(G[S])|.\}$$

(The proof is simply to observe that the minimum is a lower bound on the number of edges going from the first j vertices cleaned to elsewhere in the graph.) So, suppose that x and y are such that the expected number of sets S of $xn + o(n)$ vertices in $G \in \mathcal{G}_{n,d}$ with $yn + o(n)$ edges to the complementary $V(G) \setminus S$ is $o(1)$. Then this theorem, together with the first moment principle, gives that the brush number is a.a.s. at least $yn + o(n)$. Some standard calculations using the pairing model then give us the following lower bounds a.a.s.: $0.220n$ for $d = 4$, $0.365n$ for $d = 5$ (although this can easily be improved to the lower bound of 0.5n), $0.52n$ for $d = 6$, and $0.687n$ for $d = 7$. We omit further details from this paper.

4 Cleaning random d-regular graphs

The differential equations method (described in [12]) is used here to find an upper bound on the number of brushes needed to the clean the graph using a degree-greedy algorithm. We consider $d = 2$ first, then state some general results, and apply them to the special cases of $3 \leq d \leq 5$ before discussing higher values of d.

4.1 2-regular graphs

Let $Y = Y_n$ be the total number of cycles in a random 2-regular graph on n vertices. Since exactly two brushes are needed to clean one cycle, we need $2Y_n$ brushes in order to clean a 2-regular graph.

We know that the random 2-regular graph is a.a.s. disconnected; by simple calculations we can show that the probability of having a Hamiltonian cycle is asymptotic to $\frac{1}{2}e^{3/4}\sqrt{\pi}n^{-1/2}$ (see, for example, [10]).

We also know that the total number of cycles Y_n is sharply concentrated near $(1/2)\log n$. It is not difficult to see this by generating the random graph sequentially using the pairing model. The probability of forming a cycle in step i is exactly $1/(2n-2i+1)$, so the expected number of cycles is $(1/2)\log n + O(1)$. The variance can be calculated in a similar way. So we get that a.a.s. the brush number for a random 2-regular graph is $(1 + o(1))\log n$.

4.2 d-regular graphs ($d \geq 3$) — the general setting

In this subsection, we assume $d \geq 3$ is fixed with dn even. In order to get an asymptotically almost sure upper bound on the brush number, we study an algorithm that cleans random vertices of minimum degree. This algorithm is called *degree-greedy* because the vertex being cleaned is chosen from those with the lowest degree.

We start with a random d-regular graph $G = (V, E)$ on n vertices. Initially, all vertices are dirty: $D_0 = V$. In every step t of the cleaning process, we clean a random vertex α_t, chosen uniformly at random from those vertices with the lowest degree ($D_t = D_{t-1} \setminus \{\alpha_t\}$) in the induced subgraph $G[D_{t-1}]$. In the first step, d brushes are needed to clean random vertex α_1 (we say that this is "phase zero"). Note that this is a.a.s. the only vertex whose degree in D_t is d at the time of cleaning. Indeed, if α_t ($t \geq 2$) has degree d in $G[D_{t-1}]$, then $G[D_{t-1}]$ consists of a connected component(s) of G and thus G is disconnected. It was proven independently in [2, 11] that G is disconnected with probability $o(1)$ and later extended to d growing with n in [4]. The induced subgraph $G[D_1]$ now has d vertices of degree $d-1$ and $n-d-1$ vertices of degree d.

In the second step, $d-2$ extra brushes are needed to clean a random vertex α_2 of degree $d-1$. Typically, in the third step, a vertex of degree $d-1$ is cleaned and in each subsequent step, a vertex of degree $d-1$ is cleaned until some vertex of degree $d-2$ is produced in the subgraph induced by the set of dirty vertices. After cleaning the first vertex of degree $d-2$, we typically return to cleaning vertices of degree $d-1$, but after a some more steps of this type we may clean another vertex of degree $d-2$. When vertices of degree $d-1$ become plentiful, vertices of lower degree are more commonly created and these hiccups occur more often. When vertices of degree $d-2$ take over the role of vertices of degree $d-1$, we say (informally!) that the first phase finishes and we begin the second phase. In general, in the kth phase a mixture of vertices of degree $d-k$ and $d-k-1$ are cleaned.

It is usually difficult to study the behaviour of a greedy algorithm at the end of the process. Fortunately, in this case we need to study the first $\lfloor (d-1)/2 \rfloor$ phases since the rest of vertices are cleaned 'for free'. The details have been omitted, but can be found in [9].

For $0 \leq i \leq d$, let $Y_i = Y_i(t)$ denote the number of vertices of degree i in $G[D_t]$. (Note that $Y_0(t) = n - t - \sum_{i=1}^{d} Y_i(t)$ so $Y_0(t)$ does not need to be calculated, but it is useful in the discussion.) Let $S(t) = \sum_{l=1}^{d} lY_l(t)$ and for any statement A, let δ_A denote the Kronecker delta function

$$\delta_A = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

It is not difficult to see that

$$\begin{aligned} \mathbb{E}(Y_i(t) - Y_i(t-1) \mid G[D_{t-1}] \ \wedge \ \deg_{G[D_{t-1}]}(\alpha_t) = r) \\ &= f_{i,r}((t-1)/n, Y_1(t-1)/n, Y_2(t-1)/n, \ldots, Y_d(t-1)/n) \\ &= -\delta_{i=r} - r\frac{iY_i(t-1)}{S(t-1)} + r\frac{(i+1)Y_{i+1}(t-1)}{S(t-1)}\delta_{i+1 \leq d} \end{aligned} \qquad (2)$$

for $i, r \in [d]$ such that $Y_r(t) > 0$. Indeed, α_t has degree r, hence the term $-\delta_{i=r}$. When a pair of points in the pairing model is exposed, the probability that the other point is in a bucket of degree i (that is, the bucket contains i unchosen points) is asymptotic to $iY_i(t-1)/S(t-1)$. Thus $riY_i(t-1)/S(t-1)$ stands for the expected number of the r buckets found adjacent to α_t which have degree i. This contributes negatively to the expected change in Y_i, whilst buckets of degree $i+1$ which are reached contribute positively (of course, only if this type of vertices (buckets) exist in a graph; thus $\delta_{i+1 \leq d}$). This explains (2).

Suppose that at some step t of the phase k, cleaning a vertex of degree $d-k$ creates, in expectation, β_k vertices of degree $d-k-1$ and cleaning a vertex of degree $d-k-1$ decreases, in expectation, the number of vertices of degree $d-k-1$ by τ_k. After cleaning a vertex of degree $d-k$, we expect to then clean (on average) β_k/τ_k vertices of degree $d-k-1$. Thus, in phase k, the proportion of steps which clean vertices of degree $d-k$ is $1/(1+\beta_k/\tau_k) = \tau_k/(\beta_k+\tau_k)$. If τ_k falls below zero, vertices of degree $d-k-1$ begin to build up and do not decrease under repeated cleaning vertices of this type and we move to the next phase.

From (2) it follows that

$$\beta_k = \beta_k(x, y_1, y_2, \ldots, y_d) \;=\; f_{d-k-1,d-k}(x, y_1, y_2, \ldots, y_d) \;=\; f_{d-k-1,d-k}(x, \mathbf{y}),$$

$$\begin{aligned} \tau_k = \tau_k(x, y_1, y_2, \ldots, y_d) \;&=\; -f_{d-k-1,d-k-1}(x, y_1, y_2, \ldots, y_d) \\ &=\; -f_{d-k-1,d-k-1}(x, \mathbf{y}), \end{aligned}$$

where $x = t/n$ and $y_i(x) = Y_i(t)/n$ for $i \in [d]$. This *suggests* (see [12] for more information on the differential equations method) the following system of differential equations

$$\frac{dy_i}{dx} = F(x, \mathbf{y}, i, k)$$

where

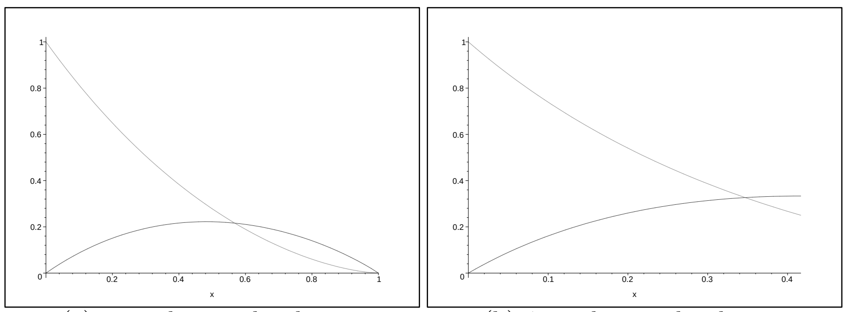
$$F(x, \mathbf{y}, i, k) = \begin{cases} \frac{\tau_k}{\beta_k+\tau_k} f_{i,d-k}(x, \mathbf{y}) + \frac{\beta_k}{\beta_k+\tau_k} f_{i,d-k-1}(x, \mathbf{y}) & \text{for } k \le d-2, \\ f_{i,1}(x, \mathbf{y}) & \text{for } k = d-1. \end{cases}$$

At this point we may formally define the interval $[x_{k-1}, x_k]$ to be phase k, where the termination point x_k is defined as the infimum of those $x > x_k$ for which at least one of the following holds: $\tau_k \le 0$ and $k < d-1$; $\tau_k + \beta_k = 0$ and $k < d-1$; $y_{d-k} \le 0$. Using final values $y_i(x_k)$ in phase k as an initial values for phase $k+1$ we can repeat the argument inductively moving from phase to phase starting from phase 1 with obvious initial conditions $y_d(0) = 1$ and $y_i(0) = 0$ for $0 \le i \le d-1$.

The general result [9, Theorem 1] studies a deprioritized version of degree-greedy algorithms, which means that the vertices are chosen to process in a slightly different way, not always the minimum degree, but usually a random mixture of two degrees. Once a vertex is chosen, it is treated the same as in the degree-greedy algorithm. The variables Y are defined in an analogous manner. The hypotheses of the theorem are straightforward to verify. The conclusion is that, for a certain algorithm using a deprioritized 'mixture' of the steps of the degree-greedy algorithm, with variables Y_i defined as above, we have that a.a.s.

$$Y_i(t) = n y_i(t/n) + o(n)$$

for $1 \le i \le d$ for phases $k = 1, 2, \ldots, m$, where m denotes the smallest k for which either $k = d-1$, or any of the termination conditions for phase k hold at x_k apart from $x_k = \inf\{x > x_{k-1} : \tau_k \leq 0\}$. We omit all details pointing the reader to [9] and the general survey [12] about the differential equations method which is a main tool in proving [9, Theorem 1]. In addition, the theorem gives information on an auxiliary variable such as, of importance to our present application, the number of brushes used. Instead of quoting this precisely, we use it merely as justification for being able to use the above equations as if they applied to the greedy algorithm. (This is no doubt the case, but it is not actually proved in [9].) The solutions to the relevant differential equations for $d = 3$ and 4 are shown in Figure 3.



(a) 3-regular graph, phase 1 (b) 4-regular graph, phase 1

Fig. 3. Solution to the differential equations.

In the kth phase a mixture of vertices of degree $d-k$ and $d-k-1$ are cleaned. Since $\max\{2l-d, 0\}$ brushes are needed to clean vertex of degree l (see (1)), we need

$$u_d^k = (1+o(1))n\Bigg(\max\{d-2k,0\} \int_{x_{k-1}}^{x_k} \frac{\tau_k}{\tau_k+\beta_k} dx + \max\{d-2k-2,0\} \int_{x_{k-1}}^{x_k} \frac{\beta_k}{\tau_k+\beta_k} dx \Bigg)$$

brushes in phase k. Thus, the total number of brushes needed to clean a graph using the degree-greedy algorithm is equal to

$$u_d = \sum_{k=1}^{\lfloor (d-1)/2 \rfloor} u_d^k = (1+o(1))n\Bigg(\sum_{k=1}^{\lfloor (d-1)/2 \rfloor} \Bigg((d-2k-2)(x_k - x_{k-1}) + 2\int_{x_{k-1}}^{x_k} \frac{\tau_k}{\tau_k+\beta_k} dx \Bigg) + \delta_{d \text{ is odd}} \int_{x_{k-1}}^{x_k} \frac{\beta_k}{\tau_k+\beta_k} dx \Bigg).$$

4.3 3-regular graphs

Let $G = (V, E)$ be any 3-regular graph on n vertices. The first vertex cleaned must start three brush paths, the last one terminates three brush paths, and all other vertices must start or finish at least one brush path, so the number of brush paths is at least $n/2 + 2$.

The result mentioned above can be shown to result in an upper bound of $n/2 + o(n)$ for the brush number of a random 3-regular (i.e. cubic) graph. We do not provide details because of the following stronger result. It is known [8] that a random 3-regular graph a.a.s. has a Hamilton cycle. The edges not in a Hamilton cycle must form a perfect matching. Such a graph can be cleaned by starting with three brushes at one vertex, and moving along the Hamilton cycle with one brush, introducing one new brush for each edge of the perfect matching. Hence the brush number of a random 3-regular graph with n vertices is a.a.s. $n/2 + 2$.

4.4 4-regular graphs

For 4-regular graphs, we are interested in phase 1 only: we need two brushes to clean vertices of degree 3, but vertices of degree 2 are cleaned 'for free'. Note that $y_1(x) = y_2(x) = 0$. We have the following system of differential equations

$$\begin{aligned} \frac{dy_4}{dx} &= \frac{-6y_4(x)}{3y_3(x) + 2y_4(x)} \\ \frac{dy_3}{dx} &= \frac{-3y_3(x) + 4y_4(x)}{3y_3(x) + 2y_4(x)} \end{aligned}$$

with the initial conditions $y_4(0) = 1$ and $y_3(0) = 0$. The particular solution (see Figure 3 (b)) to these differential equations is

$$\begin{aligned} y_4(x) &= 5 - 4\sqrt{1+3x} + 3x \\ y_3(x) &= \frac{4(-3 + 3\sqrt{1+3x} - 5x + x\sqrt{1+3x})}{2 - \sqrt{1+3x}}\,, \end{aligned}$$

so $\beta_1 = -3 + 3\sqrt{1+3x}$ and $\tau_1 = 3 - 2\sqrt{1+3x}$. Thus phase 1 finishes at time $t_1 = 5n/12$ ($x_1 = 5/12$ is a root of the equation $\tau_1(x) = 0$) and the number of vertices of degree 3 cleaned during this phase is asymptotic to

$$n\int_0^{5/12} \frac{\tau_1}{\tau_1 + \beta_1} dx = n/6\,.$$

Since we need 2 brushes to clean one such vertex we get an asymptotically almost sure upper bound of $u_4 = (1 + o(1))n/3$.

On the other hand, it is true that a.a.s. a random 4-regular graph can be decomposed into two edge-disjoint Hamilton cycles [3], and hence four paths.

Note that the following two problems can be asked in general for any $d \geq 3$.

Problem 1. Is it true that for the random case it is best to clean lowest degree vertices?

In other words, if one is going to choose a random vertex of given degree then one might as well choose a random vertex of minimum degree.

If Problem 1 is proven to be true, then the following problem should be considered. To get the brush number one might (in fact, probably should) choose non-random vertices during the cleaning process. But it might be true that a.a.s. one cannot save more than $o(n)$ brushes compared to the greedy algorithm under consideration.

Problem 2. Is it true that a.a.s. the brush number for a random d-regular graph is $u_d(1 - o(1))$?

4.5 5-regular graphs

In order to study the brush number for 5-regular graphs yielded by the degree-greedy algorithm, we cannot consider phase 1 only as before; we need 3 brushes to clean vertices of degree 4 but also 1 brush to clean vertices of degree 3. Thus two phases must be considered.

In phase 1, $y_1(x) = y_2(x) = y_3(x) = 0$ and we have the following system of differential equations

$$\begin{aligned} \frac{dy_5}{dx} &= \frac{-20y_5(x)}{8y_4(x) + 5y_5(x)} \\ \frac{dy_4}{dx} &= \frac{-8y_4(x) + 15y_5(x)}{8y_4(x) + 5y_5(x)} \end{aligned}$$

with the initial conditions $y_5(0) = 1$ and $y_4(0) = 0$. The numerical solution (see Figure 4 (a)) *suggests* that the phase finishes at time $t_1 = 0.1733n$. The number of brushes needed in this phase is asymptotic to (the numerical solution)

$$\begin{aligned} u_5^1 &= (1 + o(1)) \left(3n \int_0^{t_1/n} \frac{\tau_1}{\tau_1 + \beta_1} dx + n \int_0^{t_1/n} \frac{\beta_1}{\tau_1 + \beta_1} dx \right) \\ &= (1 + o(1)) \left(t_1 + 2n \int_0^{t_1/n} \frac{\tau_1}{\tau_1 + \beta_1} dx \right) \ \approx\ 0.3180n\,. \end{aligned}$$

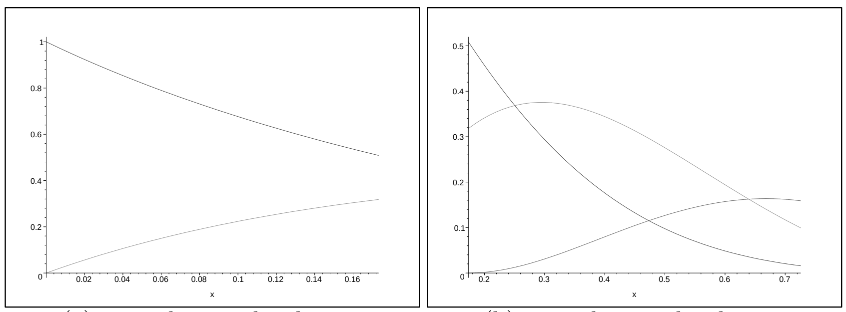
In the phase 2, $z_1(x) = z_2(x) = 0$ and we have another system of differential equations

$$\begin{aligned} \frac{dz_5}{dx} &= \frac{-15z_5(x)}{6z_3(x) + 4z_4(x) + 5z_5(x)} \\ \frac{dz_4}{dx} &= \frac{-3(4z_4 - 5z_5(x))}{6z_3(x) + 4z_4(x) + 5z_5(x)} \\ \frac{dz_3}{dx} &= \frac{-6z_3(x) + 8z_4(x) - 5z_5(x)}{6z_3(x) + 4z_4(x) + 5z_5(x)} \end{aligned}$$

with the initial conditions $z_5(t_1/n) = y_5(t_1/n) = 0.5088$, $z_4(t_1/n) = y_4(t_1/n) = 0.3180$ and $z_3(t_1/n) = 0$. The numerical solution (see Figure 4 (b)) *suggests* that the phase finishes (approximately) at time $t_2 = 0.7257n$. The number of brushes needed in this phase is asymptotic to (the numerical solution)

$$u_5^2 = (1 + o(1))n \int_{t_1/n}^{t_2/n} \frac{\tau_2}{\tau_2 + \beta_2} dx \approx 0.3259n\,.$$

Finally, we get an asymptotically almost sure upper bound of $u_5 = u_5^1 + u_5^2 \approx 0.6439n$.



(a) 5-regular graph, phase 1 (b) 5-regular graph, phase 2

Fig. 4. Solution to the differential equations.

4.6 *d*-regular graphs of higher order

d	$\lim_{n\to\infty} u_d/n$	d	$\lim_{n\to\infty} u_d/n$	d	$\lim_{n\to\infty} u_d/n$	d	$\lim_{n\to\infty} u_d/n$
3	0.5	13	2.078	23	4.159	99	21.422
4	0.334	14	2.248	24	4.358	100	21.653
5	0.644	15	2.482	25	4.589	149	33.169
6	0.684	16	2.661	26	4.791	150	33.404
7	0.949	17	2.893	27	5.022	199	45.036
8	1.057	18	3.079	28	5.227	200	45.273
9	1.305	19	3.311	29	5.457	249	56.979
10	1.444	20	3.502	30	5.664	250	57.217
11	1.684	21	3.733	31	5.895	299	68.975
12	1.842	22	3.928	32	6.104	300	69.215

Table 1. Upper bounds on the brush number for some d values.

Note that the lower bound for $d = 4$ (see Section 3) will be considerably lower than the lower bound of $n/2 + 2$ for $d = 3$, whereas the upper bound we have been discussing is the same degree-greedy algorithm in all cases. However, the upper bound is also sensitive to the parity of d. For the 4-regular case, vertices of degree 2 are processed 'for free' and so one only really worries about degree 3 vertices and there are fewer of those processed than degree 2 vertices when $d = 3$. But it seems that the parity of d does not affect the value of u_d for d big enough (see Figure 5 and Table 1).

Problem 3. Does $\lim_{d\to\infty} \lim_{n\to\infty} u_d/dn$ exist?

In Figure 5, the values of $\lim_{n\to\infty} u_d/dn$ have been presented for all d-values up to 100, although we have only listed the first 30 and a few more values for higher d in Table 1. The computations presented in the paper were performed by using Maple™ [7]. The worksheets can be found at the following address: `http://www.mathstat.dal.ca/~pralat/`.

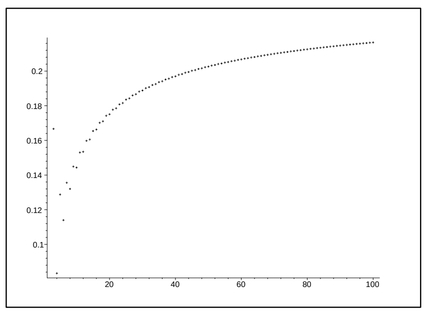


Fig. 5. A graph of $\lim_{n\to\infty} u_d/dn$ versus d (from 3 to 100).

Finally, the most important open question is clearly the following:

Problem 4. Let $G \in G(n, d)$. Is there a constant c such that the brush number is asymptotically cdn?

4.7 Other models

In this section, we present more open problems.

Problem 5. What is the brush number for binomial random graphs $G(n, p)$? What is a lower/upper bound? How about other random graph models, for example models that give power law degree distribution or d-regular graphs generated by the d-process?

Another version of the cleaning process was introduced in [5]. In this version, when a vertex is cleaned multiple brushes are allowed to traverse each dirty edge. Thus, the brush number $B(G)$ of this generalized version is at most the classic one $b(G)$. Using the degree-greedy algorithm to clean a random d-regular graph for d even, no brush 'gets stuck' in the first $\lfloor (d-1)/2 \rfloor$ phases, there is no point to introduce more brushes in the initial configuration, and vertices in the last phases are cleaned 'for free'. So the upper bound obtained is the same as before. For d odd, it is clear that one can save some brushes at phase $(d-1)/2$ but the following is still open.

Problem 6.

- Is it true that for $G \in \mathcal{G}_{n,d}$, d even, $b(G) - B(G) = o(n)$ a.a.s.?
- Is it true that for $G \in \mathcal{G}_{n,d}$, d odd, $b(G) - B(G) = \Theta(n)$ a.a.s.? How far apart are they?

References

1. Bollobás, B.: A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. European Journal of Combinatorics **1** (1980) 311–316
2. Bollobás, B.: Random graphs, Combinatorics (ed. H.N.V. Temperley): London Mathematical Society Lecture Note Series, Vol. 52, Cambridge University Press, Cambridge (1981) 80–102
3. Kim, J.H., Wormald, N.C.: Random matchings which induce Hamilton cycles and hamiltonian decompositions of random regular graphs. Journal of Combinatorial Theory, Series B **81** (2001) 20–44
4. Łuczak, T.: Sparse random graphs with a given degree sequence. In: Frieze, A., Łuczak, T. (eds.): Random Graphs, Vol. 2. Wiley, New York (1992) 165–182
5. McKeil, S.: Chip Firing Cleaning Processes. MSc Thesis, Dalhousie University (2007)
6. Messinger, M. E., Nowakowski, R. J., Prałat, P.: Cleaning a Network with Brushes, Theoretical Computer Science, accepted.
7. Monagan, M. B., Geddes, K. O., Heal, K. M., Labahn, G., Vorkoetter, S. M., McCarron, J., DeMarco, P.: Maple 10 Programming Guide. Maplesoft, Waterloo, Canada (2005)
8. Robinson, R. W., Wormald, N. C.: Almost all cubic graphs are hamiltonian. Random Structures and Algorithms **3** (1992) 117–125
9. Wormald, N. C.: Analysis of greedy algorithms on graphs with bounded degrees. EuroComb '01 (Barcelona), *Discrete Mathematics* **273** (2003) 235–260
10. Wormald, N. C.: Models of random regular graphs. In: Lamb, J.D., Preece, D.A. (eds): Surveys in Combinatorics, Vol. 276. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge (1999) 239–298
11. Wormald, N. C.: The asymptotic connectivity of labelled regular graphs. Journal of Combinatorial Theory, Series B **31** (1981) 156–167
12. Wormald, N. C.: The differential equation method for random graph processes and greedy algorithms. In: Karoński, M., Prömel, H.J. (eds): Lectures on Approximation and Randomized Algorithms. PWN, Warsaw (1999) 73–155