Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 91 (2004) 43–55
www.elsevier.com/locate/entcs

Small Edge Dominating Sets of Regular Graphs

W. Duckworth[1]

Department of Computing
Macquarie University
Sydney NSW 2109
Australia

Abstract

An *edge dominating set* $\mathcal{F}$ of a graph G is a subset of $E(G)$ such that every edge in $E(G) \setminus \mathcal{F}$ is incident with at least one vertex that is an end-point of an edge in $\mathcal{F}$. Edge dominating sets of small cardinality are of interest. We refer to the size of a smallest edge dominating set of a graph G as the edge domination number of G and denote this by $\beta(G)$. In this paper we improve all current known upper bounds on $\beta(G)$ when G is a random d-regular graph, $d \geq 3$. This is achieved by analysing a simple greedy heuristic on random regular graphs using differential equations. Our results compare favourably with known lower bounds on $\beta(G)$ when G is a random regular graph.

Keywords: random regular graph, edge dominating set, greedy algorithm

1 Preliminaries

Throughout this paper we consider connected graphs that are undirected, unweighted and contain no loops or multiple edges. A graph G is said to be d-regular if every vertex in $V(G)$ has degree d (i.e. each vertex has precisely d incident edges). When discussing any graph G, we let n denote the cardinality of $V(G)$ and for d-regular graphs on n vertices, we assume dn to be even to avoid parity problems. For other basic graph-theoretical definitions we refer the reader to Diestel [3].

[1] billy@ics.mq.edu.au

doi:10.1016/j.entcs.2003.12.005

An *edge dominating set* $\mathcal{F}$ of a graph G is a subset of $E(G)$ such that every edge in $E(G) \setminus \mathcal{F}$ is incident with at least one vertex that is an end-point of an edge in $\mathcal{F}$. Edge dominating sets of small cardinality are of interest. We refer to the size of a smallest edge dominating set of a graph G as the edge domination number of G and denote this by $\beta(G)$.

For an arbitrary graph G, the problem of determining $\beta(G)$ was shown to be NP-hard by Yannakakis and Gavril [12]. In fact, Zito [13] showed that determining $\beta(G)$ is NP-hard to solve within some constant factor of the optimal solution even for cubic (i.e. 3-regular) graphs. Horton and Kilakos [7] showed that determining $\beta(G)$ remains NP-hard for planar bipartite graphs. More recently, Zito [13] extended these NP-hardness results to include bipartite $(ks, 3s)$-graphs for every integer $s > 0$ and for $k \in \{1, 2\}$. In [7] a polynomial time algorithm was given for determining $\beta(G)$ in various classes of chordal graphs.

We consider random d-regular graphs that are generated *uniformly at random* (u.a.r.), and need some associated notation. We say that a property $\mathcal{B} = \mathcal{B}_n$ of a random graph on n vertices holds asymptotically almost surely (a.a.s.) if the probability that $\mathcal{B}$ holds tends to 1 as n tends to infinity. When d-regular graphs are the objects of consideration, this is modified so that n is restricted to even numbers if d is odd. For other basic random graph theory definitions we refer the reader to Bollobás [2] and to Janson et al. [8].

Zito [15] considered determining $\beta(G)$ for random graphs and in [14] presented upper and lower bounds on $\beta(G)$ when G is a random d-regular graph. Explicit values for these bounds were given when $3 \leq d \leq 7$. The lower bounds were calculated by means of a direct expectation argument whilst the upper bounds were calculated by using differential equations to analyse the performance of a randomised algorithm that is based on repeatedly choosing vertices of a particular degree and deleting edges.

In this paper we analyse the average-case performance of a simple heuristic, which is a random greedy algorithm, that gives upper bounds on $\beta(G)$ when G is a random d-regular graph, $d \geq 3$. This algorithm is an extension of that for $d = 3$ presented in [1] and improves all upper bounds presented in [14]. Note that the upper bound for $d = 3$ in [14] is larger than the corresponding bound presented in [1].

In the following section we give a description of our algorithm and in Section 3 we outline the method used for its analysis. Our analysis uses a theorem of Wormald [11] which we restate in Section 3. The results of this paper are encompassed by the following theorem, the proof of which is given in Section 4.

Theorem 1.1 *Let $d \geq 3$ be fixed. Then for a random d-regular graph on n*

vertices, the size of a minimum edge dominating set is asymptotically almost surely less than $\mathcal{F}_u^d n$, where the constant $\mathcal{F}_u^d$ is given in Table 1.

The constant $\mathcal{F}_u^d$ referred to in Theorem 1.1 arises from the solution of particular sets of differential equations which are derived later in the paper. These constants enable us to give our upper bounds. In Table 1, corresponding lower bounds are also given by evaluating constants $\mathcal{F}_\ell^d$ from the argument in [14] (the details of this standard expectation argument are restated in Section 5). These provide lower bounds as a comparison to our upper bounds.

Table 1
Bounds on $\beta(G)$ when G is a random d-regular graph on n vertices.

d	$\mathcal{F}_\ell^d n$	$\mathcal{F}_u^d n$
03	$0.31581n$	$0.34622n$
04	$0.31499n$	$0.35688n$
05	$0.31766n$	$0.36647n$
06	$0.32125n$	$0.37482n$
07	$0.32500n$	$0.38208n$
08	$0.32866n$	$0.38844n$
09	$0.33214n$	$0.39404n$
10	$0.33542n$	$0.39903n$
15	$0.34899n$	$0.41756n$

2 Prioritised algorithm

Consider the following algorithm that greedily finds an edge dominating set of a graph G. Repeatedly choose an edge e randomly to add to a set $\mathcal{F}$. After each edge is chosen, remove e and all edges incident with its end-points from G. Once no edges remain, the set $\mathcal{F}$ is an edge dominating set in G. It is not difficult to see that the set $\mathcal{F}$ is also an *independent-edge dominating set* of G which is also referred to as a *maximal matching*.

We modify this algorithm slightly by the way in which each subsequent edge of $\mathcal{F}$ is chosen. We assign a priority to vertices of minimum degree and add to $\mathcal{F}$, not always an edge incident with such a vertex, but sometimes an edge incident with a neighbour of such a vertex. Our algorithm is presented in Figure 1. It takes a d-regular n-vertex graph G as input and returns an edge dominating set $\mathcal{F}$ for G. We use the notation $N(v)$ to denote the set of neighbours of a vertex v and $deg(v)$ to denote the current degree of the vertex v in G. Also, we use V_i to denote the set of vertices of current degree i in G, $0 \leq i \leq d$.

```
select $u$ u.a.r. from $V(G)$;
select $v$ u.a.r. from $N(u)$;

$\mathcal{F} \leftarrow uv$;

remove all edges incident with $u$ from $E(G)$;
remove all edges incident with $v$ from $E(G)$;

while ( $|E(G)| > 0$ )
do
    $k \leftarrow \underset{v \in \{V(G) \setminus V_0\}}{\texttt{MIN}} [deg(v)]$;
    select $w$ u.a.r. from $V_k$;
    while ( $deg(w) > 0$ )
    do
        if ( $deg(w) = 1 \wedge deg(N(w)) = 1$ )
        then
            $u \leftarrow w$;
            $v \leftarrow N(w)$;
        else
            select $u$ u.a.r. from $N(w)$;
            $j \leftarrow \underset{v \in \{N(u) \setminus w\}}{\texttt{MAX}} [deg(v)]$;
            select $v$ u.a.r. from $\{N(u) \setminus w\} \cap V_j$;
        endif
        $\mathcal{F} \leftarrow \mathcal{F} \cup uv$;
        remove all edges incident with $u$ from $E(G)$;
        remove all edges incident with $v$ from $E(G)$;
    enddo
enddo
```

Fig. 1. Edge dominating set algorithm

The algorithm proceeds in a series of *operations*. In each operation, one or more edges are sequentially added to $\mathcal{F}$ and each time an edge is added to $\mathcal{F}$, all edges incident with the end-points of that edge are removed from

G. The first operation randomly selects the first edge of $\mathcal{F}$ and removes from G all edges incident with its end-points. All subsequent operations start by selecting a vertex of current degree strictly less than d. This is possible as the input graph is a d-regular graph and assumed to be connected (any induced subgraph of a connected d-regular simple graph may not be d-regular).

Each operation after the first starts by selecting a vertex w u.a.r. from those vertices of minimum positive current degree in G and the operation is split into a series of *sub-steps*. Each sub-step chooses a neighbour u of w u.a.r. If the degree of both w and u is 1, this edge is isolated from the rest of the edges in G and we simply add the edge uw to $\mathcal{F}$, remove the edge from G and start a new operation.

In all other cases, a sub-step chooses a neighbour v of u (not w) u.a.r. from those vertices of current maximum degree amongst the set $\{N(u) \setminus w\}$. The edge uv is added to $\mathcal{F}$ and all edges incident with u and v are removed from G. Sub-steps are then repeated until the degree of w becomes zero. Once this occurs, more operations are performed and this continues until no edges remain in G.

The algorithm above for finding a small edge dominating set of d-regular graphs is a modified and extended version of the algorithm in [1] that finds a small edge dominating set of cubic graphs. The algorithm in [1] is analysed as follows. Letting variables Y_i $(i = 0, \ldots, 3)$ denote the number of vertices of current degree i, the expected values of Y_i are estimated throughout the algorithm for each i using differential equations. It is shown that with high probability the variables are concentrated near their expected values. The analysis in [1] has major complications arising from the fact that priority is given to vertices of minimum current degree. We call such an algorithm *prioritised*.

The analysis of the prioritised algorithm presented in this section will be carried out using a technique introduced by Wormald [11]. This approach approximates the performance of a prioritised algorithm by analysing associated *deprioritised* algorithms. These algorithms entirely avoid prioritising by using a randomised mixture of operations. The particular mixture used for any sequence of operations is prescribed in advance but changes over the course of the algorithm in order to approximate the prioritised algorithm. One of the main objectives of using this new technique is to reduce the number of conditions that are required to be checked. Arguments in [1] involve branching processes and large deviation inequalities and the justifications of those require checking complex conditions regarding derivatives.

3 Use of deprioritised algorithms

The operations and priorities described in the prioritised algorithm given in Section 2 may be analysed using [11, Theorem 1]. This provides us with a set of differential equations whose solution describes the state of a deprioritised version of the algorithm during its execution. From this, we deduce asymptotically almost sure bounds on the size of the edge dominating set returned.

The standard model for random d-regular graphs is as follows. Take a set of dn points in n buckets labelled $1, 2, \ldots, n$, with d points in each bucket, and choose u.a.r. a pairing $P = p_1, \ldots, p_{dn/2}$ of the points such that each p_i is an unordered pair of points and each point is in precisely one pair p_i. The resulting probability space of pairings is denoted by $\mathcal{P}_{n,d}$. Form a d-regular pseudograph on n vertices by placing an edge between vertices i and j for each pair in P having one point in bucket i and one in bucket j. In order to prove that a property is a.a.s. true of a uniformly distributed random d-regular (simple) graph, it is enough to prove that it is a.a.s. true of the pseudograph corresponding to a random pairing (see Bollobás [2] and Wormald [9]).

As in [10], we redefine this model slightly by specifying that the pairs are chosen sequentially. The first point in a random pair may be selected using any rule whatsoever, as long as the second point in that random pair is chosen u.a.r. from all the remaining free (unpaired) points. This preserves the uniform distribution of the final pairing.

When a pair has been determined in the sequential process, we say that it has been *exposed*. By exposing pairs in the order which an algorithm requests their existence, the generation of the random pairing may be combined with the algorithm (as in [4,5,6]). In this way, the algorithm, such as the one in the previous section, which deletes edges, may be described in terms of operations incorporated into the pairing generation. The definition of the operations may be extended to do whatever other tasks the algorithm needs to carry out (such as exposing pairs based on the degrees of particular vertices).

The algorithm proper acts upon the final (pseudo)graph of the generation process. During the course of the generation process, the set of exposed pairs builds up this final graph and the generation process incorporates the algorithm. The order in which the edges are deleted from the final (pseudo)graph corresponds to the order in which the pairs were exposed by the generation process. For example, in the deletion algorithm, choosing a vertex of current minimum degree and selecting a neighbour of current maximum degree, would correspond to choosing a vertex of maximum degree in the evolving graph, exposing its remaining incident edges and choosing a vertex of minimum degree.

The setting of [11, Theorem 1] requires a number of definitions, and may be described as follows. It concerns a class of processes applied to the random pairing. As described above, this may be defined in terms of the generation algorithm which exposes pairs. The generation algorithm begins with the empty pairing G_0. The pairing G_{t+1} is obtained from G_t by applying an operation which may expose some of the pairs. The operation, op_t, which is applied to G_t must be one of some prespecified set of operations, Op_i, $i = 1, \ldots, d$, where Op_i consists of selecting a bucket u of degree $d - i$ in G_t u.a.r. (the *degree* of a bucket is the number of points it contains in exposed pairs) and then applying some specified set of tasks, resulting in G_{t+1}. A subset F of $E(G)$ is selected during the operations, with $F_0 = \emptyset$ initially, and $F = F_t$ for the pairing G_t.

For $1 \leq i \leq d$, let $Y_i = Y_i(t)$ denote the number of buckets of degree $d - i$ in G_t, and let $Y_{d+1} = Y_{d+1}(t)$ denote cardinality of the set F_t. Put $\mathbf{Y}(\mathbf{t}) = (Y_1(t), \ldots, Y_{d+1}(t))$ and let $\mathbf{y}$ denote $(y_1(x), \ldots, y_{d+1}(x))$. We refer the reader to [11, Theorem 1] for the motivation of the following definitions and provide a little explanation below. Given functions $f_{i,r}\left(x, \mathbf{y}\right)$, define

$$\alpha_k(x, \mathbf{y}) = f_{d-k-1,d-k}\left(x, \mathbf{y}\right),$$

(1)

$$\tau_k(x, \mathbf{y}) = -f_{d-k-1,d-k-1}\left(x, \mathbf{y}\right),$$

where

$$x = \tfrac{t}{n} \qquad \text{and}$$

(2)

$$\mathbf{y}(x) = \tfrac{\mathbf{Y}(t)}{n}.$$

We will consider the equations

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = F\left(x, \mathbf{y}, i, k\right) \tag{3}$$

where

$$F\left(x, \mathbf{y}, i, k\right) = \begin{cases} \frac{\tau_k}{\tau_k+\alpha_k} f_{i,d-k}\left(x, \mathbf{y}\right) + \frac{\alpha_k}{\tau_k+\alpha_k} f_{i,d-k-1}\left(x, \mathbf{y}\right) & k \leq d-2 \\ f_{i,1}\left(x, \mathbf{y}\right) & k = d-1 \end{cases} \tag{4}$$

and work with the parameters of $f_{i,\ell}$ in the domain

$$\mathcal{D}_\epsilon = \{(x, \mathbf{y}) : 0 \leq x \leq d,\ 0 \leq y_i \leq d \text{ for } 1 \leq i \leq d+1,\ y_d \geq \epsilon\} \tag{5}$$

for some pre-chosen value of $\epsilon > 0$. The behaviour of the process will be described in terms of the function $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}(x) = (\tilde{y}_1(x), \ldots, \tilde{y}_{d+1}(x))$ defined as follows, with reference to an initial value $x = x_0 = t_0/n$ of interest:

(6) $\tilde{y}_i(x_0) = Y_i(t_0)/n$, $i = 1, \ldots, d+1$, and inductively for $k \geq 1$, $\tilde{\mathbf{y}}$ is the solution of (3) with initial conditions $\mathbf{y}(x_{k-1}) = \tilde{\mathbf{y}}(x_{k-1})$, extending to all $x \in [x_{k-1}, x_k]$, where x_k is defined as the infimum of those $x > x_{k-1}$ for which at least one of the following holds: $\tau_k \leq 0$ and $k < d-1$; $\tau_k + \alpha_k \leq \epsilon$ and $k < d-1$; $\tilde{y}_{d-k} \leq 0$; or the solution is outside $\mathcal{D}_\epsilon$ or ceases to exist.

The interval $[x_{k-1}, x_k]$ is called Phase k. This inductive definition of $\tilde{\mathbf{y}}$ continues for phases $k = 1, 2, \ldots, m$, where

(7) m denotes the smallest k for which either $k = d-1$, or any of the termination conditions for Phase k in (6) hold at x_k apart from $x_k = \inf\{x \geq x_{k-1} : \tau_k \leq 0\}$.

It turns out that the intervals called phases have nonempty interior provided

$$\tau_k > 0 \text{ and } \tau_k + \alpha_k > \epsilon \text{ at } (x_{k-1}, \tilde{\mathbf{y}}(x_{k-1})) \quad (1 \leq k \leq \min\{d-2, m\}), \tag{8}$$

$$\begin{aligned} f_{d-1,d-1} &> 0 \text{ at } (x_0, \tilde{\mathbf{y}}(x_0)), \\ f'_{d-k,d-k}\tau_k + f_{d-k,d-k-1}f'_{d-k-1,d-k} &> 0 \text{ at } (x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))^+ \\ &\quad (1 < k \leq \min\{d-2, m\}), \\ f'_{d-k,d-k} &> 0 \text{ at } (x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))^- \\ &\quad (1 < k \leq m), \\ f'_{1,1} &> 0 \text{ at } (x_{d-2}, \tilde{\mathbf{y}}(x_{d-2}))^+ \\ &\quad (\text{if } m = d-1), \end{aligned} \tag{9}$$

with f' denoting $\frac{\mathrm{d}f(x,\tilde{\mathbf{y}}(x))}{\mathrm{d}x}$ and $(x, \tilde{\mathbf{y}}(x))^+$ and $(x, \tilde{\mathbf{y}}(x))^-$ referring to the right-hand and left-hand limits as functions of x.

We may now restate [11, Theorem 1] which we will use in the following section in connection with the edge dominating set algorithm.

Theorem 3.1 ([11]) *Let $d \geq 3$. Assume that for some fixed $\epsilon > 0$ the operations Op_r satisfy*

$$\mathbf{E}\Big(Y_i(t+1) - Y_i(t) \mid G_t \wedge \{\mathrm{op}_t = \mathrm{Op}_r\}\Big) = f_{i,r}(t/n, Y_1/n, \ldots, Y_{d+1}/n) \tag{10}$$

(within $o(1)$) for some fixed functions $f_{i,r}(x, y_1(x), \ldots, y_{d+1}(x))$ and for $i = 1, \ldots, d+1$, $r = 1, \ldots, d$, with the convergence in $o(1)$ uniform over all t and G_t for which $Y_r(t) > 0$ and $Y_d(t) > \epsilon n$. Assume furthermore that

(i) there is an upper bound, depending only upon d, on the number of pairs

exposed, and on the number of elements added to F (i.e. $|F_{t+1}| - |F_t|$), during any one operation;

(ii) the functions $f_{i,r}$ are rational functions of x, $y_1, \ldots, y_{d+1}$ with no pole in $\mathcal{D}_\epsilon$ defined in (5);

(iii) there exist positive constants C_1, C_2 and C_3 such that for $1 \le i < d$, everywhere on $\mathcal{D}_\epsilon$, $f_{i,r} \ge C_1 y_{i+1} - C_2 y_i$ when $r \ne i$, and $f_{i,r} \le C_3 y_{i+1}$ for all r.

Define $\tilde{\mathbf{y}}$ as in (6), set $x_0 = 0$, define m as in (7), and assume that (8) and (9) both hold. Then there is a randomised algorithm on $\mathcal{P}_{n,d}$ for which a.a.s. there exists t such that $|F_t| = n\tilde{y}_{d+1}(x_m) + o(n)$ and $Y_i(t) = n\tilde{y}_i(x_m) + o(n)$ for $1 \le i \le d$. Also $\tilde{y}_i(x) \equiv 0$ for $x_{k-1} \le x \le x_k$, $1 \le i \le d - k - 1$ (where $1 \le k \le m$).

Some of these definitions may be easily explained. The algorithm in Section 2 works by deleting edges; the edges deleted are equivalent to pairs exposed in the corresponding pairing generation algorithm as described above. In particular, a vertex of degree i in the original algorithm corresponds to a bucket of degree $d - i$ in the pairing version; we use *vertex degree* and *bucket degree* to distinguish these complementary measures. The algorithm gives higher priority to the buckets of highest degree (vertices of lowest degree). The phase is determined by the set of bucket degrees which are reasonably common (meaning, roughly, more than cn buckets have that degree for some $c > 0$). Phase k corresponds to a period in which the smallest such common vertex degree is $d-k$ (i.e. largest common bucket degree is k). At such a time, vertices of degree $d - k - 1$, when created, will immediately be used up, by being chosen for w in the subsequent steps, until the minimum positive vertex degree returns to $d-k$. So Phase k basically consists of a mixture of two operations: Op_{d-k} and Op_{d-k-1}. The functions α and τ represent respectively the expected net increase in Y_{k+1} in an Op_{d-k}, and the expected net decrease in Y_{k+1} in an Op_{d-k-1}. From these quantities, one may estimate the proportions of these operations being performed at any stage. The randomised algorithm referred to in the theorem uses roughly the same mixture of operations. This in turn allows us to calculate the expected changes in the variables, and the result is (4), which leads to the differential equation (3) analogous to the equations derived in [1]. In fact, for $d = 3$, the algorithm described above and the algorithm in [1] are identical.

4 Upper bounding the edge domination number

Consider the degree-greedy algorithm for finding a small edge dominating set as described in Section 2. Here, in the specification of Op_r (which first selects a random vertex, w, of degree r), the set of randomised tasks consists of choosing a neighbour u of w u.a.r. and then (possibly) choosing a different neighbour of u u.a.r. from those vertices of current maximum degree in the set $\{N(u) \setminus w\}$. An edge is then chosen to be added to $\mathcal{F}$ and edges are deleted. This sub-step is repeated until the degree of w reaches zero.

As random regular graphs a.a.s. contain few small cycles [8, Theorem 9.5] the structure of a random graph around a vertex up to a constant distance is a.a.s. a tree. Therefore, when an Op_r is performed, r edges are chosen to be part of $\mathcal{F}$. We may verify the hypotheses of Theorem 3.1. First we will show that (10) holds when $Y_d(t) > \epsilon n$ (for any $\epsilon > 0$). From here onwards in this description, *v-degree* refers to vertex degree, so v-degree i means bucket degree $d - i$. Let y_i denote Y_i/n.

An Op_r starts by selecting a vertex, w, u.a.r. from the vertices of v-degree r and chooses a neighbour u of w u.a.r. The probability that u has v-degree x, $1 \le x \le d$, is P_x where

$$\mathrm{P}_j = \frac{j y_j}{s} \qquad \text{and} \qquad s = \sum_{i=1}^{d} i y_i.$$

The probability that the neighbour v of u chosen has v-degree q is $\chi + o(1)$ which is given by

$$\chi = (\mathrm{S}_1^q)^{x-1} - (\mathrm{S}_1^{q-1})^{x-1}$$

where $\mathrm{S}_a^b = \sum_{x=a}^{b} \mathrm{P}_x$.

The probability that u has b neighbours (other than w) of v-degree q, (assuming that q is the maximum current v-degree amongst the neighbours of u) is $\nu/\chi + o(1)$ where

$$\nu = (\mathrm{P}_q)^b \binom{x-1}{b} (\mathrm{S}_1^{q-1})^{x-b-1}.$$

The expected number of neighbours of u (other than w) that have v-degree j, $1 \le j \le q-1$ is $\gamma/\nu + o(1)$ where

$$\gamma = (\mathrm{P}_q)^b \binom{x-1}{b} (\mathrm{S}_1^{q-1})^{x-b-2} (x-b-1)\mathrm{P}_{\mathrm{j}}.$$

The $o(1)$ terms in the statements above are due to the fact that the values of all these variables may change by a constant during an operation. Here we assume $s \geq \epsilon n$ for some fixed $\epsilon > 0$ which means that the error is in fact $O(1/n)$. Later we discuss operations where $s < \epsilon n$.

So we have that (10) holds with

$$\begin{aligned} f_{i,r} = -\delta_{i=r} - r\mathrm{P}_i &+ r\left(\textstyle\sum_{x=2}^{d} \mathrm{P}_x \sum_{q=1}^{d} \left[\chi((q-1)\rho_i - \delta_{i=q}) + \right]\right) \\ &+ r\left(\textstyle\sum_{x=2}^{d} \mathrm{P}_x \sum_{b=1}^{x-1} \left[\nu(b-1)\Phi_i^q + \sum_{j=1}^{q-1} \gamma \Phi_i^j\right]\right) \end{aligned} \tag{11}$$

where

$$\begin{aligned} \Phi_i^j &= \delta_{i=j-1} - \delta_{i=j}, \\ \rho_i &= \mathrm{P}_{i+1} - \mathrm{P}_i, \quad 1 \leq i \leq d-1, \\ \rho_d &= -\mathrm{P}_d \end{aligned} \tag{12}$$

and for any statement Q, $\delta_Q = 1$ if Q evaluates to `TRUE` and $\delta_Q = 0$ if Q evaluates to `FALSE`.

The first term in (11) denotes the change in v-degree of w from r to 0 and the second term denotes the change in the v-degrees of the r neighbours of w from i to 0 (one for each of the sub-steps). The remaining terms denote the expected change due to finding r edge dominating set edges and removing the edges incident with their end-points. The expected change in the size of $\mathcal{F}$ per operation is simply r.

We are now ready to verify the hypotheses of Theorem 3.1. Hypothesis (i) of Theorem 3.1 is immediate since in any operation at most $2d^2$ pairs are exposed and at most r edges are added to $\mathcal{F}$. The functions $f_{i,r}$ satisfy (ii) because from (11) their (possible) singularities satisfy $s = 0$, which lies outside $\mathcal{D}_\epsilon$ since in $\mathcal{D}_\epsilon$, $s \geq y_d \geq \epsilon$. Hypothesis (iii) follows from (11) again using $s \geq y_d \geq \epsilon$ and the boundedness of the functions y_i (which follows from the boundedness of $\mathcal{D}_\epsilon$). Thus, defining $\tilde{\mathbf{y}}$ as in (6) with $t_0 = 0$, $Y_d(0) = n$ and $Y_i(0) = 0$ for $i \neq d$, we may solve (3) numerically to find m, verifying (8) and (9) at the appropriate points of the computation.

It turns out that these hold for each d in Table 1, and that in each case $m = d - 1$, for sufficiently small $\epsilon > 0$. For such ϵ, the value of $\tilde{y}_{d+1}(x_m)$ may be computed numerically (the result is shown as the constant $\mathcal{F}_u^d$ in Table 1), and then by Theorem 3.1, this is the asymptotic value of the size of the edge dominating set $\mathcal{F}$ at the end of some randomised algorithm. So the conclusion is that a random d-regular graph a.a.s. has an edge dominating set of size at most $n\tilde{y}_{d+1}(x_m) + o(n)$. Note also that (by the theorem) $\tilde{y}_i(x) \equiv 0$ in Phase k for $1 \leq i \leq d-k-1$, and by the nature of the differential equation, $\tilde{y}_i(x)$ will be strictly positive for $i > d - k$. So by (6) and (7), the end of the process (for ϵ arbitrarily small) occurs in Phase $d-1$ when $\tilde{y}_1$ becomes 0.

Once $s < \epsilon n$, the change in variables per operation is bounded by a constant. Hence, letting ϵ tend to 0 sufficiently slowly, in $o(n)$ operations the change in the random variables Y_i and F is $o(n)$.

The equations were solved using a Runge-Kutta method giving accuracy to at least five decimal places. This completes the proof of Theorem 1.1.

5 Lower bounding the edge domination number

Zito [14] gave lower bounds on the size of a minimum maximal matching (minimum independent-edge dominating set) for random d-regular graphs. Using a standard expectation argument, it was shown that for a random d-regular graph on n vertices, where $d \geq 3$, the number of maximal matchings of size λ is at most

$$\frac{d^{2\lambda}(2\lambda(d-1))^{2\lambda(d-1)}(d/2)^{d/2}2^{d(1-2\lambda)}}{\lambda^{\lambda}(1-2\lambda)^{1-2\lambda}(\lambda(2d-1))^{\lambda(2d-1)}d^{d}}.$$

Finding the value of λ above which this expression is strictly greater than 1 gives lower bounds on $\beta(G)$ when G is a random d-regular graph. Evaluating this function for various values of d gives the constants $\mathcal{F}_{\ell}^{d}$ reported in Table 1.

6 Remarks

It is worth noting that the same line of research has already yielded results on other closely related abstract properties of random regular graphs (e.g. independent sets and induced matchings). Whereas, in the majority of other cases, such as that in [14,15], each operation adds one element to the set that is under construction, here we add up to $d-1$ elements per operation. This and exploiting the well known structure of random regular graphs has allowed us to improve upon the known upper bounds for the problem under consideration.

References

[1] Assiyatun H. and W. Duckworth, Small maximal matchings of random cubic graphs. *Submitted.*

[2] Bollobás, B., *Random graphs.* Academic Press Incorporated, London, 1985.

[3] Diestel, R., *Graph theory.* Springer-Verlag, New York, 2000.

[4] Duckworth, W., Maximum 2-independent sets of random cubic graphs. *The Australasian Journal of Combinatorics*, **27**:63-80, 2003.

[5] Duckworth,W. and N. C. Wormald, Minimum independent dominating sets of random cubic graphs. *Random Structures and Algorithms*, **21(2)**:147–161, 2002.

[6] Duckworth, W. and M. Zito. Large 2-independent sets of regular graphs. *Electronic Notes in Theoretical Computer Science*, **78**:1-13(electronic), 2003.

[7] Horton, J. D. and K. Kilakos, Minimum edge dominating sets. *SIAM Journal on Discrete Mathematics*, **6(3)**:375–387, 1993.

[8] Janson, S., T. Łuczak and A. Rucinski, *Random graphs*. Wiley, New York, 2000.

[9] Wormald, N. C., Models of random regular graphs. *Surveys in Combinatorics*, 1999 (Canterbury). London Mathematical Society Lecture Note Series, **267**:239–298, Cambridge University Press, Cambridge, 1999.

[10] Wormald, N. C., The differential equation method for random graph processes and greedy algorithms. *Lectures on Approximation and Randomized Algorithms*, 73–155, PWN, Warsaw, 1999. M. Karoński and H. Prömel (editors).

[11] Wormald, N. C., Analysing greedy algorithms on regular graphs. *Electronic Notes in Discrete Mathematics*, **10**, (electronic), 2001. M.N.J. Nesetril and O. Serra (editors).

[12] Yannakakis, M. and F. Gavril, Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, **38(3)**, 364–372, 1980.

[13] Zito, M., Randomised techniques in combinatorial Algorithmics. *Doctoral Thesis*, The University of Warwick, 1999.

[14] Zito, M., Greedy algorithms for minimisation problems in random regular graphs. *Lecture Notes in Computer Science*, **2161**:524–536, Springer-Verlag, Berlin, 2001. F.M. auf der Heide, (editor).

[15] Zito, M., Small maximal matchings in random graphs. *Theoretical Computer Science*, **297(1-3)**:487–507, 2003.