Minimum Independent Dominating Sets of Random Cubic Graphs

W. Duckworth[*] and N.C. Wormald[†]

Department of Mathematics & Statistics,
University of Melbourne, Australia

Abstract

We present a heuristic for finding a small independent dominating set, $\mathcal{D}$, of cubic graphs. We analyse the performance of this heuristic, which is a random greedy algorithm, on random cubic graphs using differential equations and obtain an upper bound on the expected size of $\mathcal{D}$. A corresponding lower bound is derived by means of a direct expectation argument. We prove that $\mathcal{D}$ asymptotically almost surely satisfies $0.2641n \leq |\mathcal{D}| \leq 0.2794n$.

1 Introduction

A dominating set, S, of a graph, G, is a subset of the vertices of G such that for every vertex $v \in V(G)$, S contains either v itself or some neighbour of v in G. An *independent* dominating set, $\mathcal{D}$, of a graph, G, is a dominating set such that no two vertices of $\mathcal{D}$ are connected by and edge of G. We are interested in finding independent dominating sets of small cardinality.

The problem of finding a *minimum* independent dominating set of a graph is one of the core, well-known, NP-hard graph-theoretic optimisation problems [4]. Halldórsson [5] showed that, for general n-vertex graphs, this problem is not approximable within $n^{1-\epsilon}$ for any $\epsilon > 0$. Kann [6] showed that the same problem, when restricted to graphs of bounded degree, is APX-complete. Note that, for d-regular graphs, it is simple to verify that this problem is approximable within $(d+1)/2$.

Molloy and Reed [8] showed that, for a *random* n-vertex cubic graph G, the the size of a smallest *dominating set*, $D(G)$, asymptotically almost surely satisfies $0.2636n \leq |D(G)| \leq 0.3126n$. The algorithm they use to prove their upper bound finds a minimum dominating set in the random 3-regular multigraph formed by taking the union of a Hamilton cycle on n vertices and a uniformly selected perfect matching on the same vertex set. The simplistic approach would be to construct the dominating set by choosing every third vertex around the Hamilton cycle. Molloy and Reed prove their upper bound by modifying this simplistic approach and considering the probability that a vertex is adjacent to a member of the dominating set across a matching edge. A result of Robinson and Wormald [10] ensures that the result obtained translates to uniformly distributed random cubic graphs. Their lower bound is obtained by means of a direct expectation argument.

[*] Currently in the Department of Computing, Macquarie University, Australia

[†] Supported by the Australian Research Council

Reed [9] showed that the size of a minimum dominating set of an n-vertex cubic graph is at most $3n/8$ and gave an example of a cubic graph on eight vertices with no dominating set of size less than 3, demonstrating the tightness of this bound. Lam, Shiu and Sun [7] recently showed that the size of a minimum independent dominating set of an n-vertex cubic graph is at most $2n/5$ and gave an example of a cubic graph on ten vertices with no independent dominating set of size less than 4.

In this paper, we present a heuristic for finding a small independent dominating set of cubic graphs. We analyse the performance of this heuristic, which is a random greedy algorithm, on random n-vertex cubic graphs using differential equations and obtain an upper bound on the expected size of the independent dominating set, $\mathcal{D}$, returned by the algorithm. A corresponding lower bound is calculated by means of a direct expectation argument. We show that $\mathcal{D}$ asymptotically almost surely satisfies $0.2641n \leq |\mathcal{D}| \leq 0.2794n$.

A deterministic version of the randomised algorithm that we present in this paper was analysed in [3] using linear programming. It was shown that, given an n-vertex cubic graph, the deterministic algorithm returns an independent dominating set of size at most $29n/70 + O(1)$ and there exist infinitely many n-vertex cubic graphs for which the algorithm only attains this bound. In the same paper, it was also shown that there exist infinitely many n-vertex cubic graphs that have no independent dominating set of size less than $3n/8$.

Throughout this paper we use the notation $\mathbf{P}$ (probability), $\mathbf{E}$ (expectation), u.a.r. (uniformly at random) and a.a.s. (asymptotically almost surely) (see, for example, Bollobàs [1] for these and other random graph theory definitions). When discussing any cubic graph on n vertices, we assume n to be even to avoid parity problems.

In the following section we introduce the model used for generating cubic graphs u.a.r. and in Section 3 we describe the notion of analysing the performance of algorithms on random graphs using a system of differential equations. Section 4 gives the randomised algorithm and Section 5 gives its analysis showing the a.a. sure upper bound. In Section 6 we give a direct expectation argument showing the a.a. sure lower bound.

2 Generating Random Cubic Graphs

The model used to generate a cubic graph u.a.r. (see, for example, [1]) may be summarised as follows. For an n-vertex cubic graph

- take $3n$ points in n buckets labelled $1 \dots n$ with three points in each bucket and
- choose u.a.r. a disjoint pairing of the $3n$ points.

If no pair contains two points from the same bucket and no two pairs contain four points from just two buckets, this represents a cubic graph on n vertices with no loops and no multiple edges. With probability bounded below by a positive constant, loops and multiple edges do not occur (see, for example, [12, Section 2.2]). The buckets represent the vertices of the randomly generated cubic graph and each pair represents an edge whose end-points are given by the buckets of the points in the pair.

We may consider the generation process as follows. Initially, all vertices have degree 0. Throughout the execution of the generation process, vertices will increase in degree until the generation is complete and all vertices have degree 3. During this process, we refer to the graph being generated as the *evolving graph*.

3 Analysis Using Differential Equations

One method of analysing the performance of a randomised algorithm is to use a system of differential equations to express the expected changes in variables describing the state of the algorithm during its execution. Wormald [13] gives an exposition of this method and Duckworth [2] applies this method to various other graph-theoretic optimisation problems.

The algorithm we use to find an independent dominating set, $\mathcal{D}$, of cubic graphs is a greedy algorithm based on selecting vertices of given degree. We say that our algorithm proceeds as a series of *operations*. For each operation, a vertex v is chosen u.a.r. from those of current minimum degree. A vertex is chosen to be added to $\mathcal{D}$ from v and its neighbours based on the degree(s) of the neighbour(s) of v. If v has a neighbour of degree strictly greater than that of v, a vertex is chosen to be added to $\mathcal{D}$ u.a.r. from those of maximum degree amongst the neighbours of v. Otherwise, we add v to $\mathcal{D}$. The edges incident with the chosen vertex and its neighbours are then deleted in order to ensure that the dominating set remains independent. Any isolated vertices created, which were not neighbours of the chosen independent dominating set vertex, are then added to $\mathcal{D}$. We refer to these vertices as *accidental isolates*.

In order to analyse our algorithm using a system of differential equations, we incorporate the algorithm as part of a pairing process that generates a random cubic graph. In this way, we generate the random graph in the order that the edges are examined by the algorithm.

During the generation of a random cubic graph, we choose the pairs sequentially. The first point, p_i, of a pair may be chosen by any rule, but in order to ensure that the cubic graph is generated u.a.r., the second point, p_j, of that pair must be selected u.a.r. from all the remaining free (i.e. unpaired) points. The freedom of choice of p_i enables us to select it u.a.r. from the vertices of given degree in the evolving graph. Using $B(p_k)$ to denote the bucket that the point p_k belongs to, we say that the edge $(B(p_i), B(p_j))$ is *exposed*. Note that we may then determine the degree of the vertex represented by the bucket $B(p_j)$, without exposing any further edges incident with that vertex.

Incorporating our algorithm as part of a pairing process that generates a random cubic graph, we select a vertex, v, u.a.r. from those of maximum degree in the evolving graph and expose its incident edge(s). A vertex is selected to be added to $\mathcal{D}$ based on the degree(s) of the new neighbour(s) of v. Further edges are then exposed in order to ensure the dominating set remains independent. More detail is given in the following section.

In what follows, we denote the set of vertices of degree i of the evolving graph, at time t, by $V_i = V_i(t)$ and let $Y_i = Y_i(t)$ denote $|V_i|$. We can express the state of the evolving graph at any point during the execution of the algorithm by considering Y_0, Y_1 and Y_2. In order to analyse our randomised algorithm for finding an independent dominating set, $\mathcal{D}$, of cubic graphs, we calculate the expected change in this state over one unit of time (a unit of time is defined more clearly in Section 5) in relation to the expected change in the size of $\mathcal{D}$. Let $D = D(t)$ denote $|\mathcal{D}|$ at any stage of the algorithm (time t) and let $\mathbf{E}\Delta X$ denote the expected change in a random variable X conditional upon the history of the process. We then regard $\mathbf{E}\Delta Y_i / \mathbf{E}\Delta D$ as the derivative dY_i/dD, which gives a system of differential equations. The solutions to these equations describe functions which represent the behaviour of the variables Y_i. There is a general result which guarantees that the solutions of the differential equations almost surely approximate the variables Y_i. The expected size of the independent dominating set may be deduced from these results.

4 The Algorithm

In this section we present the algorithm incorporated with the pairing process, for finding an independent dominating set of a random cubic graph. It is noteworthy that relaxing the independence condition does not suggest any alternative approach along similar lines, so in some sense we get independence for free.

We denote the set of all free points in the evolving graph by P and use $q(b)$ to denote the set of free points in a given bucket b. The combined algorithm and pairing process, RANDMIDS, is given in Figure 1; a description is given below.

```
select $u$ u.a.r. from $V_0$;
$\mathcal{D} \leftarrow \{u\}$;
$E \leftarrow \{\}$
**isolate**$(u)$;
Add any accidental isolates to $\mathcal{D}$;
**while** $(Y_1 + Y_2 > 0)$
{
  **if** $(Y_2 > 0)$
            select $v$ u.a.r. from $V_2$;
            $\{p_1\} \leftarrow q(v)$;
            select $p_2$ u.a.r. from $P$;
            $u \leftarrow B(p_2)$;
            add the edge $uv$ to $E$;
  **else**
            select $v$ u.a.r. from $V_1$;
            $\{p_1, p_2\} \leftarrow q(v)$;
            select $p_3$ u.a.r. from $P$;
            $j \leftarrow b(p_3)$;
            select $p_4$ u.a.r. from $P$;
            $k \leftarrow b(p_4)$;
            add $vj$ and $vk$ to $E$;
            **if**       $(j \in V_2 \wedge k \in V_1)$ $u \leftarrow k$;
            **else if**  $(j \in V_1 \wedge k \in V_2)$ $u \leftarrow j$;
            **else if**  $(j \in V_2 \wedge k \in V_2)$ $u \leftarrow v$;
            **else**     select $u$ u.a.r from $\{j, k\}$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{u\}$;
  **isolate**$(u)$;
  Add any accidental isolates to $\mathcal{D}$;
}
```

Figure 1: Algorithm RANDMIDS

When the algorithm terminates, as we see below, $\mathcal{D}$ is the independent dominating set of the graph, whose edge set is E. The function **isolate**(b) involves the process of exposing all edges incident with b and its neighbours (including adding those edges to E). This is achieved by randomly selecting a mate for each free point of b and then exposing all edges incident with free points in the buckets of these selected mates. Note that when **isolate**(b) is applied, the accidental isolates are just all vertices which enter V_3 but are not b or its neighbours.

It is straightforward to verify that $\mathcal{D}$ is, in the end, an independent dominating set of the graph. To see that it is dominating, note that all vertices enter V_3 eventually. Those entering V_3 other than as accidental isolates, are either a vertex being added to $\mathcal{D}$ or one of its neighbours. All accidental isolates are also placed in $\mathcal{D}$, so $\mathcal{D}$ is dominating. For independence, note that vertices can only enter $\mathcal{D}$ from V_0, V_1 or V_2. The function **isolate**(b) ensures that for any vertex b to which it is applied, all neighbours of b enter V_3 without entering $\mathcal{D}$. So the only possible edges between vertices in $\mathcal{D}$ are those between accidental isolates. However, as every edge added to the graph is incident with a vertex entering V_3 inside the function **isolate**(b) (either b or one of its neighbours), accidental isolates cannot be adjacent to each other.

The algorithm terminates when there are no vertices of degree 1 or 2 remaining, which means that a connected component has been completely generated and an independent dominating set has been found in that component. It is well known that a random cubic graph is a.a.s. connected, so the result is a.a.s. an independent dominating set in the whole graph.

The first operation of the algorithm is the operation that randomly selects the first vertex of the independent dominating set. We split the remainder of the algorithm into two distinct phases. We informally define Phase 1 as the period of time where any vertices in V_2 that are created are used up almost immediately and Y_2 remains small. Once the rate of generating vertices in V_2 becomes larger than the rate that they are used up, the algorithm moves into Phase 2 and all operations involve selecting a vertex from V_2. The transition point between phases is not obvious but arises in our analysis.

There are two types of operation performed by the algorithm. Figure 2 shows the preferred selection of independent dominating set vertex for a typical operation when a vertex is chosen from V_1 and we shall call this Type 1. Similarly we have Figure 3 when a vertex is chosen from V_2 and we shall call this Type 2. In both cases the independent dominating set vertex is chosen as to maximise the number of edges exposed.

Figure 2: Type 1 operations

Figure 3: Type 2 operations

The larger circles represent buckets with the points of that bucket represented by smaller circles. Points that were (without a doubt) free (respectively used up) at the start of an operation are coloured black (respectively white). Other points are shaded. In all cases, the selected vertex is labelled v and the independent dominating set vertex chosen is labelled u. Vertices of unknown degree at the start of an operation are labelled wither w or p. We refer to these vertices as *rems* (for "remove") and *incs* (for "increase") respectively. An *inc* will have its degree increased by 1 for the next operation of the algorithm. A *rem* will have all its incident edges exposed. Should any *rem* be incident with another vertex of unknown degree, these vertices will be *incs*. Edges that are certain to be exposed are represented by solid lines. For vertices of unknown degree, incident dotted lines indicate that an edge is to be exposed if the point that the edge is incident with is free.

5 The Upper Bound

We analyse the combined algorithm and pairing process using differential equations and in this way prove the following theorem.

Theorem 1 *A random cubic graph on n vertices asymptotically almost surely has a minimum independent dominating set with less than $0.2794n$ vertices.*

Proof We define a *clutch* to be a series of operations in Phase i from an operation of Type i up to but not including the next operation of Type i. We proceed with an examination of each of the two phases, before giving a formal definition of the distinction between the phases. Initially, one only needs to assume that the process begins in Phase 1 and that in Phase 2 there are no operations of Type 1.

5.1 Preliminary Equations For Phase 1

The initial operation of Phase 1 is of Type 1 (at least a.a.s.). A vertex v is chosen u.a.r. from V_1 and all edges incident with v are exposed. Once the degrees of the neighbours of v are known, a vertex is chosen to be added to the independent dominating set based on the criteria shown by Figure 2. The next operation of the algorithm may be of Type 1 or Type 2 depending on the size of the set V_2. For simplicity, we consider operations of Type 2 first and then combine the equations given by these operations with those given by the operations of Type 1.

Operations of Type 2 involve the selection of a vertex v from V_2 (which has been created from processing a vertex from V_1). Let $s = s(t)$ denote the number of free points available in all buckets at a given stage (time t). Note that

$$s = \sum_{i=0}^{2}(3-i)Y_i.$$

For our analysis it is convenient to assume that $s > \epsilon n$ for some arbitrarily small but fixed $\epsilon > 0$. Later we discuss the last operations of the algorithm, when $s \leq \epsilon n$.

The expected change in Y_i due to changing the degree of an *inc* from i to $i+1$ by exposing one of its incident edges (at time t) is $\rho_i + o(1)$ where

$$\rho_i = \rho_i(t) = \frac{(i-3)Y_i + (4-i)Y_{i-1}}{s}, \quad 0 \leq i \leq 2$$

and this equation is valid under the assumption that Y_{-1}=0.

To justify this, note that when the point in the *inc* was chosen, the number of points in the buckets corresponding to vertices currently of degree i is $(3-i)Y_i$, and s is the total number of points. In this case Y_i decreases; it increases if the selected point is from a vertex of degree $i-1$. These two quantities are added because expectation is additive. The term $o(1)$ comes about because the values of all these variables may change by a constant during the course of the operation being examined. Since $s > \epsilon n$ the error is in fact $O(1/n)$.

The expected change in Y_i due to exposing all edges incident with a *rem* and its incident *incs* (at time t) is $\mu_i + o(1)$ where

$$\mu_i = \mu_i(t) = \frac{(i-3)Y_i}{s} + \frac{(6Y_0+2Y_1)\rho_i}{s}, \quad 0 \le i \le 2.$$

The first term represents the removal of the *rem* from V_i. The expected number of *incs* incident with a *rem* is $(6Y_0+2Y_1)/s + o(1)$ and each of these will have its degree increased by 1 (giving the second term).

The expected change in Y_i for an operation of Type 2 in Phase 1 (at time t) is $\alpha_i + o(1)$ where

$$\alpha_i = \alpha_i(t) = \frac{(i-3)Y_i}{s} + \frac{(6Y_0+2Y_1)\mu_i}{s} - \delta_{i2}, \quad 0 \le i \le 2 \qquad (1)$$

in which δ denotes the Kronecker delta function.

We now consider operations of Type 1. The expected change in Y_i for operation $1h$ given in Figure 2 (at time t) is $\beta_{h,i} + o(1)$ where

$$\begin{aligned}
\beta_{a,i} &= \beta_{a,i}(t) = -2\delta_{i0} + 2\mu_i, \quad 0 \le i \le 2,\\
\beta_{b,i} &= \beta_{b,i}(t) = -\delta_{i0} - 2\delta_{i1} + \delta_{i2} + 2\mu_i, \quad 0 \le i \le 2 \quad \text{and}\\
\beta_{c,i} &= \beta_{c,i}(t) = -3\delta_{i1} + 2\rho_i, \quad 0 \le i \le 2.
\end{aligned}$$

For an operation of Type 1 in Phase 1, neighbours of v (the vertex selected at random from V_1) were in $\{V_0 \cup V_1\}$ at the start of the operation, since $Y_2 = 0$ when the algorithm performs this type of operation. The probability that these neighbours were in V_0 or V_1 are asymptotically $3Y_0/s$ and $2Y_1/s$ respectively. Therefore the probabilities that, given we are performing an operation of Type 1 in Phase 1, the operation is that of type 1a, 1b or 1c are given by

$$\begin{aligned}
\mathbf{P}(1a) &= \tfrac{9Y_0^2}{s^2} + o(1),\\
\mathbf{P}(1b) &= \tfrac{12Y_0Y_1}{s^2} + o(1) \quad \text{and}\\
\mathbf{P}(1c) &= \tfrac{4Y_1^2}{s^2} + o(1)
\end{aligned}$$

respectively.

We define a *birth* to be the generation of a vertex in V_2 by processing a vertex of V_1 or V_2 in Phase 1. The expected number of births from processing a vertex from V_1 (at time t) is $\nu_1 + o(1)$ where

$$\nu_1 = \nu_1(t) = \frac{9Y_0^2}{s^2} \times 2\mu_2 + \frac{12Y_0Y_1}{s^2} \times (1 + 2\mu_2) + \frac{4Y_1^2}{s^2} \times \frac{4Y_1}{s}.$$

Here, for each case, we consider the probability that vertices of degree 1 (in the evolving graph) become vertices of degree 2 by exposing an edge incident with the vertex.

Similarly, the expected number of births from processing a vertex from V_2 (at time t) is $\nu_2 + o(1)$ where

$$\nu_2 = \nu_2(t) = \frac{(6Y_0 + 2Y_1)\mu_2}{s}.$$

Consider the Type 1 operation at the start of the clutch to be the first generation of a *birth-death* process in which the individuals are the vertices in V_2, each giving birth to a number of children (essentially independent of the others) with expected number ν_2. Then, the expected number in the j^{th} generation is $\nu_1 {\nu_2}^{j-1}$ and the expected total number of births in the clutch is

$$\frac{\nu_1}{1 - \nu_2}.$$

For Phase 1, the equation giving the expected change in Y_i for a clutch is therefore given by

$$\mathbf{E}\Delta Y_i = \frac{9Y_0^2 \beta_{a,i}}{s^2} + \frac{12 Y_0 Y_1 \beta_{b,i}}{s^2} + \frac{4Y_1^2 \beta_{c,i}}{s^2} + \frac{\nu_1 \alpha_i}{1 - \nu_2} + o(1). \qquad (2)$$

This assumes $Y_1 + Y_2$ is not zero, an eventuality which will be discussed later.

In Phase 1, since $|V_2|$ is very small and $|V_1|$ is at least a constant times n, the probability of an accidental isolate being created is negligible in any one operation: when two edges are exposed they are highly unlikely to go to the same neighbour. Thus, the expected increase in the size of the independent dominating set is 1 for any single operation in Phase 1. So the equation giving the expected increase in D for a clutch in Phase 1 is given by

$$\mathbf{E}\Delta D = 1 + \frac{\nu_1}{1 - \nu_2} + o(1). \qquad (3)$$

5.2 Preliminary Equations For Phase 2

In Phase 2, all operations are considered to be of Type 2 and therefore a clutch consists of one operation. The expected change in Y_i is given by

$$\mathbf{E}\Delta Y_i = \alpha_i + o(1) \qquad (4)$$

where α_i remains the same as that given for Phase 1. The expected increase in D is given by

$$\mathbf{E}\Delta D = 1 + \left(\frac{6Y_0 + 2Y_1}{s}\right)^2 \frac{Y_2}{s} + o(1)$$

representing an increase of 1 for the chosen vertex plus a further increase depending on the probability that there are accidental isolates created.

5.3 The Differential Equations

Equation (2) representing $\mathbf{E}\Delta Y_i$ for processing a clutch in Phase 1 forms the basis of a differential equation. Write $Y_i(t) = nz_i(t/n)$, $\mu_i(t) = n\tau_i(t/n)$, $\beta_{j,i}(t) = n\psi_{j,i}(t/n)$, $s(t) = n\xi(t/n)$, $\alpha_i(t) = n\chi_i(t/n)$ and $\nu_j(t) = n\omega_j(t/n)$. The differential equation suggested is

$$z_i' = \frac{9z_0^2}{\xi^2}\psi_{a,i} + \frac{12 z_0 z_1}{\xi^2}\psi_{b,i} + \frac{4z_1^2}{\xi^2}\psi_{c,i} + \frac{\omega_1}{1 - \omega_2}\chi_i, \quad 0 \le i \le 2 \qquad (5)$$

where differentiation is with respect to x and xn represents the number, t, of clutches. From the definitions of μ, β, s, α and ν we have

$$\begin{aligned}
\tau_i &= \tfrac{(i-3)z_i}{\xi} + \tfrac{(6z_0+2z_1)((i-3)z_i+(4-i)z_{i-1})}{\xi^2}, \quad 0 \le i \le 2,\\
\psi_{a,i} &= -2\delta_{i0} + 2\tau_i, \quad 0 \le i \le 2,\\
\psi_{b,i} &= -\delta_{i0} - 2\delta_{i1} + \delta_{i2} + 2\tau_i, \quad 0 \le i \le 2,\\
\psi_{c,i} &= -3\delta_{i1} + \tfrac{2((i-3)z_i+(4-i)z_{i-1})}{\xi}, \quad 0 \le i \le 2,\\
\chi_i &= \tfrac{(i-3)z_i}{\xi} + \tfrac{(6z_0+2z_1)\tau_i}{\xi} - \delta_{i2}, \quad 0 \le i \le 2,\\
\omega_1 &= \tfrac{18z_0^2\tau_2}{\xi^2} + \tfrac{12z_0z_1(1+2\tau_2)}{\xi^2} + \tfrac{16z_1^3}{\xi^3},\\
\omega_2 &= \tfrac{(6z_0+2z_1)\tau_2}{\xi}, \quad \text{where}\\
\xi &= \textstyle\sum_{i=0}^{2}(3-i)z_i.
\end{aligned}$$

Using the equation representing the expected increase in the size of $\mathcal{D}$ after processing a clutch in Phase 1 and writing $D(t) = nz(t/n)$ suggests the differential equation for z as

$$z' = 1 + \frac{\omega_1}{1-\omega_2}. \tag{6}$$

For Phase 2 the equation representing $\mathbf{E}\Delta Y_i$ for processing a clutch suggests the differential equation

$$z_i' = \chi_i, \quad 0 \le i \le 2. \tag{7}$$

The equation representing the increase in the size of D after processing a vertex in Phase 2, suggests the differential equation for z as

$$z' = 1 + \left(\frac{6z_0+2z_1}{\xi}\right)^2 \frac{z_2}{\xi}. \tag{8}$$

The solution to these systems of differential equations represents the cardinalities of the sets V_i and $\mathcal{D}$ (scaled by $1/n$) for given t. For Phase 1, the equations are (5) and (6) with initial conditions

$$z_0(0) = 1, \quad z_i(0) = 0 \ \ (i > 0).$$

The initial conditions for Phase 2 are given by the final conditions for Phase 1 and the equations are given by (7) and (8).

We use a result from [13] to show that during each phase, the functions representing the solutions to the differential equations almost surely approximate the variables Y_i/n and D/n with error o(1). For this we need some definitions.

Consider a probability space whose elements are sequences $(q_0, q_1, \ldots)$ where each $q_t \in S$. We use h_t to denote $(q_0, q_1, \ldots, q_t)$, the *history* of the process up to time t, and H_t for its random counterpart. $S^{(n)+}$ denotes the set of all $h_t = (q_0, \ldots, q_t)$ where each $q_i \in S$, $t = 0, 1, \ldots$. All these things are indexed by n and we will consider asymptotics as $n \to \infty$.

We say that a function $f(u_1, \ldots, u_j)$ satisfies a *Lipschitz condition* on $W \subseteq \mathbb{R}^j$ if a constant $L > 0$ exists with the property that

$$|f(u_1, \ldots, u_j) - f(v_1, \ldots, v_j)| \le L \max_{1 \le i \le j} |u_i - v_i|$$

for all $(u_1, \dots, u_j)$ and $(v_1, \dots, v_j)$ in W, and note that $\max_{1 \le i \le j} |u_i - v_i|$ is the distance between $(u_1, \dots, u_j)$ and $(v_1, \dots, v_j)$ in the ℓ^∞ metric.

For variables $Y_1, \dots, Y_a$ defined on the components of the process, and $W \subseteq \mathbb{R}^{a+1}$, define the *stopping time* $T_W = T_W(Y_1, \dots, Y_a)$ to be the minimum t such that $(t/n, Y_1(t)/n, \dots, Y_a(t)/n) \notin W$.

The following is a restatement of [13, Theorem 6.1]. We refer the reader to that paper for explanations, and to [11] for a similar result with virtually the same proof.

Theorem 2 *Let $\widehat{W} = \widehat{W}(n) \subseteq \mathbb{R}^{a+1}$. For $1 \le l \le a$, where a is fixed, let $y_l : S^{(n)+} \to \mathbb{R}$ and $f_l : \mathbb{R}^{a+1} \to \mathbb{R}$, such that for some constant C_0 and all l, $|y_l(h_t)| < C_0 n$ for all $h_t \in S^{(n)+}$ for all n. Let $Y_l(t)$ denote the random counterpart of $y_l(h_t)$. Assume the following three conditions hold, where in (ii) and (iii) W is some bounded connected open set containing the closure of*

$$\{(0, z_1, \dots, z_a) : \mathbf{P}(Y_l(0) = z_l n, 1 \le l \le a) \ne 0 \textit{ for some } n\} .$$

(i) For some functions $\beta = \beta(n) \ge 1$ and $\gamma = \gamma(n)$, the probability that

$$\max_{1 \le l \le a} |Y_l(t+1) - Y_l(t)| \le \beta,$$

conditional upon H_t, is at least $1 - \gamma$ for $t < \min\{T_W, T_{\widehat{W}}\}$.

(ii) For some function $\lambda_1 = \lambda_1(n) = o(1)$, for all $l \le a$

$$|\,\mathbf{E}(Y_l(t+1) - Y_l(t) \,|\, H_t) - f_l(t/n, Y_1(t)/n, \dots, Y_a(t)/n)\,| \le \lambda_1$$

for $t < \min\{T_W, T_{\widehat{W}}\}$.

(iii) Each function f_l is continuous, and satisfies a Lipschitz condition, on

$$W \cap \{(t, z_1, \dots, z_a) : t \ge 0\},$$

with the same Lipschitz constant for each l.

Then the following are true.

(a) For $(0, \hat{z}_1, \dots, \hat{z}_a) \in W$ the system of differential equations

$$\frac{dz_l}{dx} = f_l(x, z_1, \dots, z_a), \qquad l = 1, \dots, a$$

has a unique solution in W for $z_l : \mathbb{R} \to \mathbb{R}$ passing through

$$z_l(0) = \hat{z}_l,$$

$1 \le l \le a$, and which extends to points arbitrarily close to the boundary of W;

(b) Let $\lambda > \lambda_1 + C_0 n\gamma$ with $\lambda = o(1)$. For a sufficiently large constant C, with probability $1 - O(n\gamma + \frac{\beta}{\lambda}\exp(-\frac{n\lambda^3}{\beta^3}))$,

$$Y_l(t) = nz_l(t/n) + O(\lambda n)$$

uniformly for $0 \le t \le \min\{\sigma n, T_{\widehat{W}}\}$ and for each l, where $z_l(x)$ is the solution in (a) with $\hat{z}_l = \frac{1}{n}Y_l(0)$, and $\sigma = \sigma(n)$ is the supremum of those x to which the solution can be extended before reaching within ℓ^∞-distance $C\lambda$ of the boundary of W.

First, we apply Theorem 2 to the Process within Phase 1. For arbitrary small ϵ, define W to be the set of all (t, z_0, z_1, z_2, z) for which $t > -\epsilon$, $\xi > \epsilon$, $\omega_2 < 1 - \epsilon$, $z > -\epsilon$ and $z_i < 1 + \epsilon$ where $0 \leq i \leq 2$. Also define $\widehat{W}$ to be the vectors for which $z_1 \geq 0$, $z_2 \geq 0$ and $z_1 + z_2 > 0$.

For part (i) of Theorem 2 we must ensure that $Y_i(t)$ does not change too quickly throughout the process. As long as the expected number of births in a clutch is bounded above, the probability of getting say n^ϵ births is $O(n^{-K})$ for any fixed K. This comes from a standard argument as in [13, page 141]. So part (i) of Theorem 2 holds with $\beta = n^\epsilon$ and $\gamma = n^{-K}$. Near the start of the process, operations may be of Type 1 or Type 2. Equations (2) and (3) verify part (ii) for a function λ_1 which goes to zero sufficiently slowly. (Note in particular that since $\xi > \epsilon$ inside W, the assumption that $s > \epsilon n$ used in deriving these equations is justified. Also, since $t < T_{\widehat{W}}$, it follows that $Y_1 + Y_2 > 0$, so that the next operation is of Type 1 or Type 2.) Part (iii) of Theorem 2 is immediate from the form of the functions in equations (2) and (3).

The conclusion of Theorem 2 therefore holds. This implies (taking $\lambda \to 0$ sufficiently slowly) that the random variables Y_i/n and D/n a.a.s. remain within $o(1)$ of the corresponding deterministic solutions to the differential equations (5) and (6) until a point arbitrarily close to where it leaves the set W, or until $t = T_{\widehat{W}}$ if that occurs earlier. Since the latter can only occur when the algorithm has completely processed a component of the graph, and a random cubic graph is a.a.s. connected, we may turn to examining the former.

We compute the ratio $dz_i/dz = z_i'(x)/z'(x)$ and we have

$$\frac{dz_i}{dz} = \frac{\frac{9z_0^2}{\xi^2}\psi_{a,i} + \frac{12z_0z_1}{\xi^2}\psi_{b,i} + \frac{4z_1^2}{\xi^2}\psi_{c,i} + \frac{\omega_1}{1-\omega_2}\chi_i}{1 + \frac{\omega_1}{1-\omega_2}}, \quad 0 \leq i \leq 2, \tag{9}$$

where differentiation is with respect to z and all functions can be taken as functions of z.

By solving (numerically) this system of differential equations, we find that the solution hits a boundary of the domain at ω_2=$1 - \epsilon$ (for ϵ=0 this would be at $z \geq 0.1375$). At this point, we may formally define Phase 1 as the period of time from time t=0 to the time t_0 such that $z = t_0/n$ is the solution of ω_2=1.

Our next aim is to show that by the time $\epsilon' n$ operations after the start of Phase 2 (for some $\epsilon' > 0$), the variable Y_2 is a.a.s. at least some constant times n. For this, the main requirement is that the variable ν_2 increases significantly above 1, since $\nu_2 - 1$ is the expected increase in Y_2 when processing a vertex of in V_2.

Unfortunately, the expected increase in ν_2 due to processing a vertex from V_1 right near the end of Phase 1 is negative. So instead we consider the variable $\widehat{\nu_2}$ defined by setting $Y_2 = 0$ in the definitions of all variables; that is,

$$\widehat{\nu_2} = \widehat{\nu_2}(t) = \frac{(6Y_0 + 2Y_1)\widehat{\mu_2}}{\widehat{s}}$$

where

$$\begin{aligned} \widehat{\mu_2} &= \widehat{\mu_2}(t) = \frac{(6Y_0 + 2Y_1)\widehat{\rho_2}}{\widehat{s}}, \\ \widehat{\rho_2} &= \widehat{\rho_2}(t) = \frac{2Y_1}{\widehat{s}} \text{ and} \\ \widehat{s} &= 3Y_0 + 2Y_1. \end{aligned}$$

Regarding $\widehat{\nu_2}$ as a function of Y_0 and Y_1 only, we may compute the expected increase in $\widehat{\nu_2}$ due to an operation of Type 1 as

$$\frac{\partial \widehat{\nu_2}}{\partial Y_0} E_0 + \frac{\partial \widehat{\nu_2}}{\partial Y_1} E_1 \tag{10}$$

where E_i is the expected increase in Y_i in such an operation. The latter can be computed from the first three terms on the right side of (2). Plugging in the values of Y_0 and Y_1 at the end of Phase 1 gives a positive quantity, approximately 3.86. For a Type 2 operation, the same calculation is used, but the values of E_1 and E_2 come from α_i as seen in (2). The result is 3.93.

Since the formula given by (10) is Lipschitz, it must remain positive for at least $\epsilon_1 n$ operations after reaching time $t_0 - \epsilon n$, for ϵ_1 sufficiently small. Subject to the choice of ϵ_1, we may take ϵ arbitrarily small. It now follows by the usual large deviation argument that the increase in $\widehat{\nu_2}$ between time $t_0 - \epsilon n$ and a time t_1 when $\epsilon_1 n$ operations have occurred in Phase 2 is a.a.s. at least c for some positive constant c. By choosing ϵ sufficiently small, ν_2 is a.a.s. arbitrarily close to 1 at time $t_0 - \epsilon n$, and so the same goes for $\widehat{\nu_2}$ since Y_0 is a.a.s. very small in Phase 1. Thus $\widehat{\nu_2} > 1 + c_1$ a.a.s. at time t_1 for some $c_1 > 0$.

Once this value of $\widehat{\nu_2}$ is attained, since $\widehat{\nu_2} = \nu_2$ when $Y_2 = 0$ we can choose a $c > 0$ such that either $Y_2 > cn$ or $\nu_2 > 1+c$. In the former case we are well into Phase 2 in the informal sense. In the latter case, due to the Lipschitz property of ν_2, for the next $\epsilon_2 n$ operations, processing a vertex from V_2 produces an expected $1 + c/2$ new vertices of V_2. Again, using the usual large deviation argument, this ensures that with high probability the process moves in the next $\epsilon_2 n$ operations into a state where $V_2 > c_2 n$, and is thus, again, firmly entrenched in Phase 2 in the informal sense. Thus, in either case, there will be some time t_2 which is followed by $c_2 n$ consecutive operations of Type 2, which means that the equations for Phase 2 are valid.

For Phase 2 and for arbitrary small ϵ, define W' to be the set of all (t, z_0, z_1, z_2, z) for which $t > t_2 - \epsilon$, $\xi > \epsilon$, $z > -\epsilon$ and $z_i < 1 + \epsilon$ where $0 \leq i \leq 2$. Theorem 2 applies as in Phase 1 (with time shifted by subtracting t_2) except that here, a clutch consists of just one operation of Type 2. Note also that the starting point of the process is randomised, which is permitted in Theorem 2. Computing the ratio $dz_i/dz{=}z_i'(x)/z'(x)$ gives

$$\frac{dz_i}{dz} = \frac{\chi_i}{1 + \left(\frac{6z_0+2z_1}{\xi}\right)^2 \frac{z_2}{s}}, \qquad 0 \leq i \leq 2.$$

By solving this we see that the solution hits a boundary of W' at $\xi = \epsilon$ (for $\epsilon = 0$ this would be approximately $0.2794n$).

The differential equations were solved using a Runge-Kutta method, giving $\omega_2{=}1$ at $z \geq 0.1375$ and in Phase 2, $z_2{=}0$ at $z > 0.2794$. This corresponds to the size of the independent dominating set (scaled by $1/n$) when all vertices are used up, thus proving the theorem. □

6 The Lower Bound

We now establish a lower bound on the size of a minimum independent dominating set of a random cubic graph.

Theorem 3 *For a random cubic graph on n vertices the size of a minimum independent dominating set is asymptotically almost surely greater than* $0.2641n$.

Proof Consider a random cubic graph G on n vertices. Let $D(G,k)$ denote the number of independent dominating sets of G of size k. We calculate $\mathbf{E}D(G,k)$ and show that when $k = 0.2641n$, $\mathbf{E}D(G,k) = o(1)$, thus proving the theorem.

We denote the number of vertices of $V(G) \setminus D(G)$ that are dominated by 1 vertex in $D(G)$ by a and the number of vertices of $V(G) \setminus D(G)$ that are dominated by 2 vertices in $D(G)$ by b. The remaining $n-k-a-b$ vertices of $V(G) \setminus D(G)$ are dominated by 3 vertices in $D(G)$. Clearly $b = 3n - 6k - 2a$. For this situation the expected number of independent dominating sets of G of cardinality k is given by

$$\begin{aligned}\mathbf{E}D(G,k) &= \frac{\binom{n}{k}\,(3k)!\,\binom{n-k}{a}\,3^a\,\binom{n-k-a}{b}\,3^b\,(3n-6k)!\,\left(\frac{3n}{2}\right)!\,2^{\left(\frac{3n}{2}\right)}}{\left(\frac{3n}{2}-3k\right)!\,2^{\frac{3n}{2}-3k}\,(3n)!} \\ &= \frac{n!\,3^{a+b}\,(3k)!\,(3n-6k)!\,\left(\frac{3n}{2}\right)!\,2^{3k}}{k!\,a!\,b!\,(n-k-a-b)!\,\left(\frac{3n}{2}-3k\right)!\,(3n)!}.\end{aligned}$$

Approximate using Stirling's formula and rewrite using $f(x) = x^x$, $\alpha = a/n$, $\beta = b/n$ and $\kappa = k/n$ and we have

$$(\mathbf{E}D(G,k))^{\frac{1}{n}} \sim \frac{3^{\alpha+\beta}\,f(3\kappa)\,f(3-6\kappa)\,f(\frac{3}{2})\,2^{3\kappa}}{f(\kappa)\,f(\alpha)\,f(\beta)\,f(1-\kappa-\alpha-\beta)\,f(\frac{3}{2}-3\kappa)\,f(3)}.$$

Substitute for β and we have

$$\mathbf{E}D(G,k))^{\frac{1}{n}} \sim \frac{3^{3-6\kappa-\alpha}\,f(3\kappa)\,f(3-6\kappa)\,f(\frac{3}{2})\,2^{3\kappa}}{f(\kappa)\,f(\alpha)\,f(3-6\kappa-2\alpha)\,f(\alpha+5\kappa-2)\,f(\frac{3}{2}-3\kappa)\,f(3)}. \tag{11}$$

We can now differentiate the expression on the right with respect to α and use this to find that for $\kappa = 0.2641$ it is strictly less than 1. □

References

[1] B. Bollobás. *Random Graphs.* Academic Press, London, 1985.

[2] W. Duckworth. *Greedy Algorithms and Cubic Graphs.* PhD thesis, Department of Mathematics and Statistics, University of Melbourne, Australia, 2001.

[3] W. Duckworth and N.C. Wormald. Linear Programming and the Worst-Case Analysis of Greedy Algorithms on Cubic Graphs. (*In preparation.*)

[4] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and Company, 1979.

[5] M.M. Halldórsson. Approximating the Minimum Maximal Independence Number. *Information Processing Letters*, 46(4):169–172, 1993.

[6] V. Kann. *On the Approximability of NP-Complete Optimisation Problems.* PhD thesis, Department of Numerical Analysis, Royal Institute of Technology, Stockholm, 1992.

[7] Peter Che Bor Lam, Wai Chee Shiu and Liang Sun. On the Independent Domination Number of Regular Graphs. *Discrete Mathematics*, 202:135–144, 1999.

[8] M. Molloy and B. Reed. The Dominating Number of a Random Cubic Graph. *Random Structures and Algorithms*, 7(3):209–221, 1995.

[9] B. Reed. Paths, Stars and the Number Three. *Combinatorics, Probability and Computing*, 5:277–295, 1996.

[10] R.W. Robinson and N.C. Wormald. Almost All Cubic Graphs are Hamiltonian. *Random Structures and Algorithms*, 3(7):117–125, 1992.

[11] N.C. Wormald, Differential equations for random processes and random graphs, *Annals of Applied Probability* 5:1217–1235, 1995.

[12] N.C. Wormald. Models of Random Regular Graphs. *Surveys in Combinatorics, 1999 (Canterbury)*, 239–298, Cambridge University Press, Cambridge, 1999.

[13] N.C. Wormald. The Differential Equation Method for Random Graph Processes and Greedy Algorithms. In *Lectures on Approximation and Randomized Algorithms*, 73–155, PWN, Warsaw, 1999. Michał Karoński and Hans-Jürgen Prömel (editors).