Minimum connected dominating sets of random cubic graphs*

W. Duckworth
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
billy@ics.mq.edu.au

Submitted: August 14, 2001; Accepted: February 14, 2002.
MR Subject Classifications: 05C80, 05C69

Abstract

We present a simple heuristic for finding a small connected dominating set of cubic graphs. The average-case performance of this heuristic, which is a randomised greedy algorithm, is analysed on random n-vertex cubic graphs using differential equations. In this way, we prove that the expected size of the connected dominating set returned by the algorithm is asymptotically almost surely less than $0.5854n$.

1 Introduction

A dominating set of a graph, G, is a subset, $\mathcal{D}$, of the vertices of G such that for every vertex v of G, either $v \in \mathcal{D}$ or there exists a vertex $u \in \mathcal{D}$ incident with v in G. A *connected* dominating set, $\mathcal{C}$, of a graph, G, is a dominating set such that the subgraph induced by the vertices of $\mathcal{C}$ in G is connected. We are interested in finding connected dominating sets of small cardinality. For other basic graph theory definitions not defined here, the reader is referred to [2].

The problem of finding a *minimum* connected dominating set of a graph is polynomially equivalent to finding a *maximum leaf spanning tree* of the graph. This well-known, NP-hard, optimisation problem [6, Problem ND2] is defined as follows. A spanning tree of a graph, G, is a connected spanning subgraph, T, of G that does not contain a cycle. Vertices of degree 1 in T are called leaves and we are interested in finding a spanning tree with a set of leaves of large cardinality. Note that the non-leaf vertices of T form a connected dominating set of G.

*This research was carried out whilst the author was in The Department of Mathematics & Statistics, The University of Melbourne, VIC 3010, Australia

Solis-Oba [11] showed that the maximum leaf spanning tree problem is approximable with approximation ratio 2, improving the previous best known approximation ratio of 3 by Lu and Ravi [9]. Galbiati, Maffioli and Morzenti [5] showed that the same problem does not exhibit a Polynomial Time Approximation Scheme, unless P=NP.

A graph, G, is said to be d-regular if every vertex of G has degree d. In this paper we consider simple, connected, cubic (i.e. 3-regular) graphs. Also, when considering any such graph on n vertices, we assume n to be even to avoid parity problems. Note that for such graphs, it is simple to show that the minimum connected dominating set problem is approximable with approximation ratio 2.

For a graph, G, define $L(G)$ to be the maximum number of leaves in any spanning tree of G. Storer [10] showed that for an n-vertex connected cubic graph, G, $L(G) \geq \lceil (n/4) + 2 \rceil$. This worst-case bound is the best possible since there exists infinitely many n-vertex connected cubic graphs that have no more than $\lceil (n/4) + 2 \rceil$ leaves. Griggs, Kleitman and Shastri [7] presented and analysed an algorithm that constructs a spanning tree of an n-vertex connected cubic graph with at least $\lceil (n/4) + 2 \rceil$ leaves. They also showed that for an n-vertex connected cubic graph, G, that has no subgraph isomorphic to "$K_4 - e$" (K_4 with one edge removed), $L(G) \geq \lceil (n/3) + (4/3) \rceil$.

Duckworth and Wormald [4] gave a new derivation, at least to within an additive constant, of the main result of [10]. They also showed that the size of a connected dominating set of an n-vertex cubic graph of girth at least 5 is at most $2n/3 + O(1)$. The linear programming technique that was used to analyse the performance of the algorithms that were presented, also demonstrated the existence of infinitely many cubic graphs for which the algorithms only achieve these bounds. An example was given of an infinite family of n-vertex cubic graphs of girth at least 5 that have no connected dominating set of size less than $4n/7 - O(1)$.

As we consider regular graphs that are generated u.a.r. (uniformly at random), we need some notation. We use the notation $\mathbf{P}$ (probability), $\mathbf{E}$ (expectation) and say that a property, $\mathcal{B} = \mathcal{B}_n$, of a random regular graph on n vertices holds a.a.s. (asymptotically almost surely) if $\lim_{n\to\infty} \mathbf{P}(\mathcal{B}_n)$=1. For other basic random graph theory definitions not defined here, the reader is referred to [8].

The algorithms of [4, 7], that find a small connected dominating set of cubic graphs, guarantee that the size of the connected dominating set returned is at most $3n/4 + O(1)$ in the worst-case. In this paper we consider the average-case behaviour of a randomised version of these algorithms. We analyse the performance of this randomised algorithm on random n-vertex cubic graphs using differential equations. In this way, we prove that the expected size of the connected dominating set returned by the algorithm is a.a.s. less than $0.5854n$.

The following section gives a brief description of our algorithm. In Section 3 we describe the model we use for generating cubic graphs u.a.r. and describe the notion of analysing the performance of algorithms on random graphs using systems of differential equations. Details of our algorithm are given in Section 4 and its analysis is presented in Section 5 proving our a.a. sure upper bound.

2 A Simple Heuristic

The heuristic we describe is a randomised greedy algorithm that is based on repeatedly selecting vertices of given current degree from an ever-shrinking subgraph of the input graph. At the start of our algorithm all vertices have degree 3. Throughout the execution of the algorithm edges are deleted and the algorithm terminates when all vertices have degree 0.

For a cubic graph, G, the algorithm constructs a subset, C, of the vertices of G in a series of *steps*. Each step starts by selecting a vertex u.a.r. from those vertices of a particular current degree. The first step is unique in the sense that it is the only step in which a vertex is selected u.a.r. from the vertices of degree 3. We select such a vertex u.a.r. to add to C and delete all of its incident edges. Note that, as G is assumed to be connected, after the first step and before the completion of the algorithm, there always exists a vertex of current degree 1 or 2.

For each step after the first, if there exists vertices of current degree 2, such a vertex, u, is chosen u.a.r. Otherwise we select u u.a.r. from those vertices of current degree 1. We then choose a vertex, v, u.a.r. from the neighbours of u and add u to C based on the current degree of v. If v has degree 3, we add u to C and delete all edges incident with u. Otherwise, we complete the step by deleting the edge between u and v. Note that for each step, vertices other than that chosen for possible addition to C have their degree decreased by at most 1. Each time such a vertex has its degree decreased from 3 to 2, the vertex u is added to C. This ensures that C is dominating in G at the end of the algorithm. As each vertex selected for possible addition to C (after the first) is chosen from those vertices of current degree 1 or 2, the subgraph induced by the vertices of C in G is always connected.

3 Random Graphs and Differential Equations

3.1 Generating Random Cubic Graphs

The model we use to generate a cubic graph u.a.r. (see, for example, Bollobás [1]) may be summarised as follows. For an n-vertex cubic graph: take $3n$ points in n buckets labelled $1 \ldots n$ (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. 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. With probability bounded below by a positive constant, loops and multiple edges do not occur (see, for example, Wormald [13, Section 2.2]).

Generating a random cubic graph in this way may be considered 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. We refer to the graph being generated throughout this process as the *evolving graph*.

3.2 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 [14] gives an exposition of this method and Duckworth [3] applies this method to various other graph-theoretic optimisation problems.

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. We refer to selecting p_j as choosing a *mate* for p_i. The freedom of choice of p_i enables us to select it u.a.r. from the vertices of given current 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 from $B(p_i)$ to $B(p_j)$ is *exposed*. Note that we may then determine the current degree of the vertex represented by the bucket $B(p_j)$ without exposing any further edges.

The incorporated algorithm and pairing process may be loosely summarised as follows. Repeatedly select a vertex, u, u.a.r. from those vertices of given current degree in the evolving graph and expose an edge incident with u. This is achieved by selecting a point, p_1, u.a.r. from the free points in the bucket corresponding to u and selecting a mate, p_2, for p_1 u.a.r. from all the remaining free points in the evolving graph. The choice of whether to add u to the set under construction will depend on the current degree of the vertex represented by the bucket that the point p_2 belongs to. Further edges incident with u may then be exposed. More detail is given in the following section.

In what follows, we denote the set of vertices of current 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 a connected dominating set, $\mathcal{C}$, 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 $\mathcal{C}$. Let $C = C(t)$ denote $|\mathcal{C}|$ 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. The equations representing $\mathbf{E}\Delta Y_i$ and $\mathbf{E}\Delta C$ are then used to derive a system of differential equations. The solutions to the differential equations describe functions which represent the behaviour of the variables Y_i. Wormald [14, Theorem 6.1] describes a general result which guarantees that the solutions of the differential equations almost surely approximate the variables Y_i. The expected size of the connected dominating set may be deduced from these results.

4 The Algorithm

In Figure 1 we present our algorithm combined with a pairing process. This combination, RANDCDS, generates an n-vertex cubic graph, G, u.a.r. and, at the same time, finds a subset, $\mathcal{C}$, of the vertices of G.

```
select u u.a.r. from V_0;
C ← {u};
expose all edges incident with u;
while (Y_1 + Y_2 > 0) do
    if (Y_1 > 0)
        select u u.a.r. from V_1;
    else
        select u u.a.r. from V_2;
    endif
    expose an edge incident with u;     \* to a vertex v (say) *\
    if (v ∈ V_1)
        C ← C ∪ {u};
        expose all edges incident with u;
    endif
enddo
```

Figure 1: RANDCDS

Note that all vertices chosen to be part of $\mathcal{C}$ (after the first) were in V_1 or V_2 at the start of the iteration of the loop that that selected them. This ensures that the subgraph induced by the vertices of $\mathcal{C}$ in G is connected. The algorithm terminates when $Y_1 + Y_2 = 0$. At such time, either a connected cubic component has been generated and $Y_0 > 0$, or a dominating set has been found for G. It is well known that cubic graphs are a.a.s. connected, so the result is a.a.s. a connected dominating set in the whole graph.

We select the first element of $\mathcal{C}$ u.a.r. from all of the vertices in the evolving graph and expose all of its incident edges. We say that the remainder of the combined algorithm and pairing process proceeds in *operations* where each operation is denoted by one iteration of the while loop. There are two basic types of operation. A Type 1 operation refers to an operation where $Y_1 > 0$ and a vertex, u, is selected u.a.r. from V_1. Similarly, a Type 2 operation refers to an operation where $Y_1 = 0$ and a vertex, u, is selected u.a.r. from V_2. For both types of operation, once u has been selected, an edge incident with u is exposed. This is achieved by selecting a point, p_1, u.a.r. from the free points in the bucket corresponding to u and selecting a mate, p_2, for p_1 u.a.r. from all the remaining free points in the evolving graph. Let v to denote the vertex corresponding to the bucket that the point p_2 belongs to. If v now has current degree 1, we add u to $\mathcal{C}$ and expose the remaining edges incident with u (if any).

5 Algorithm Analysis

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

Theorem 1 *The size of a minimum connected dominating set of a random n-vertex cubic graph is asymptotically almost surely less than* $0.5854n$.

Proof After the first element of $\mathcal{C}$ has been chosen, we split the remainder of the algorithm into two distinct *phases*. We informally define Phase 1 as the period of time from the first Type 1 operation up to but not including the first Type 2 operation. Phase 2 is informally defined as the remainder of the process from the first Type 2 operation to the end of the algorithm. 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. For a clutch of operations in each phase we develop equations to represent the expected changes in the variables Y_i in relation to the expected change in the size of $\mathcal{C}$. These equations are then formulated as a system of differential equations.

5.1 Preliminary Equations For Phase 1

In Phase 1 all operations are of Type 1 and therefore a clutch consists of just one operation. 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 $s \leq \epsilon n$.

For a Type 1 operation in Phase 1, we select a vertex, u, u.a.r. from V_1 and expose an incident edge by selecting a point, p_1, u.a.r. from the free points of u and selecting its mate, p_2, u.a.r. from all the free points in the evolving graph. Let v denote the bucket that p_2 belongs to.

The expected change in Y_i due to changing the degree of v from i to $i+1$ (at time t) is $\rho_i + o(1)$ where

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

and this equation is valid under the assumption that Y_{-1}=0. To justify this, note that when the point p_2 was chosen, the number of points in the buckets corresponding to vertices currently of degree i is $(3-i)Y_i + o(1)$, 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 probability that v was of degree 0 before the start of the operation is $3Y_0/s+o(1)$. In such an instance, we expose the remaining edge incident with u and add u to $\mathcal{C}$. Otherwise v had degree strictly greater than 0 before the start of the operation. In which case, the degree of u is increased to 2. For both instances, the size of the set V_1 decreases by 1 and a vertex of unknown degree has its degree increased by 1.

The expected change in Y_i for an operation of Type 1 in Phase 1 (and therefore a clutch) is $\beta_i + o(1)$ where

$$\beta_i = \beta_i(t) = -\delta_{i1} + \rho_i + \frac{3Y_0}{s}\rho_i + \left(1 - \frac{3Y_0}{s}\right)\delta_{i2}, \qquad (0 \le i \le 2) \tag{1}$$

in which δ_{ij} denotes the Kronecker delta function.

The expected increase in C for a clutch in Phase 1 is just

$$\mathbf{E}(\Delta C) = \frac{3Y_0}{s} + o(1) \tag{2}$$

as we add u to $\mathcal{C}$ if v had degree 0 at the start of the operation.

5.2 Preliminary Equations For Phase 2

The initial operation of Phase 1 is of Type 2. For simplicity, we consider operations of Type 1 first and then combine the equations given by these operations with those given by the operations of Type 2.

For an operation of Type 1 in Phase 2, the expected change in Y_i is the same as that for an operation of Type 1 in Phase 1 and we have

$$\mathbf{E}\Delta Y_i = \beta_i = -\delta_{i1} + \rho_i + \frac{3Y_0}{s}\rho_i + \left(1 - \frac{3Y_0}{s}\right)\delta_{i2}, \qquad (0 \le i \le 2).$$

We now consider operations of Type 2. A vertex, u, is chosen u.a.r. from V_2 and an edge incident with u is exposed to a vertex v (say). If v had degree 0 before the start of the operation, we add u to $\mathcal{C}$. The expected change in Y_i for an operation of Type 2 in Phase 2 is $\alpha_i + o(1)$ where

$$\alpha_i = \alpha_i(t) = -\delta_{i2} + \rho_i, \qquad (0 \le i \le 2).$$

We define a *birth* to be the generation of a vertex in V_1 by performing an operation of Type 1 or Type 2 in Phase 2. The expected number of births from a Type 1 operation (at time t) is $\nu_1 + o(1)$ where

$$\nu_1 = \nu_1(t) = \frac{3Y_0Y_2}{s^2} + \frac{6Y_0Y_1}{s^2} + 2\frac{9Y_0^2}{s^2} = \frac{3Y_0(s+3Y_0)}{s^2}.$$

Here we consider the probability that we expose edges to vertices that were of degree 0 at the start of the operation. Similarly, the expected number of births from a Type 2 operation (at time t) is $\nu_2 + o(1)$ where

$$\nu_2 = \nu_2(t) = \frac{3Y_0}{s}.$$

Consider the Type 2 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_1, each giving birth to a number of children (essentially independent of the others) with expected number ν_1. Then, the expected number in the j^{th} generation is $\nu_2{\nu_1}^{j-1}$ and the expected total number of births in the clutch is $\nu_2/(1-\nu_1)$.

For Phase 2, the expected change in Y_i for a clutch is given by

$$\mathbf{E}(\Delta Y_i) = \alpha_i + \frac{\nu_2}{1-\nu_1}\beta_i + o(1), \qquad (0 \le i \le 2) \tag{3}$$

and the expected increase in the size of $\mathcal{C}$ for a clutch is given by

$$\mathbf{E}(\Delta C) = \frac{3Y_0}{s}\left(1 + \frac{\nu_2}{1-\nu_1}\right) + o(1). \tag{4}$$

The contribution to the increase in the size of $\mathcal{C}$ by the Type 2 operation in a clutch is 1 if v had degree 0 at the start of the operation. As random regular graphs a.a.s. contain few small cycles [8, Theorem 9.5], for each birth we have a Type 1 operation (a.a.s.).

5.3 The Differential Equations

We use the preliminary equations derived in the previous two subsections to formulate a system of differential equations for each phase. Write $Y_i(t) = nz_i(t/n)$, $\rho_i(t) = n\psi_i(t/n)$, $\beta_i(t) = n\chi_i(t/n)$, $\alpha_i(t) = n\tau_i(t/n)$, $s(t) = n\xi(t/n)$ and $\nu_j(t) = n\omega_j(t/n)$. From the definitions of ρ, β, α, s and ν we have

$$\begin{aligned}
\psi_i &= \tfrac{(i-3)z_i+(4-i)z_{i-1}}{\xi}, \qquad (0 \le i \le 2),\\
\xi &= \textstyle\sum_{i=0}^{2}(3-i)z_i, \qquad (0 \le i \le 2),\\
\chi_i &= -\delta_{i1} + \psi_i + \tfrac{3z_0}{\xi}\psi_i + \left(1 - \tfrac{3z_0}{\xi}\right)\delta_{i2}, \qquad (0 \le i \le 2),\\
\tau_i &= -\delta_{i2} + \psi_i, \qquad (0 \le i \le 2),\\
\omega_1 &= \tfrac{3z_0(\xi+3z_0)}{\xi^2} \qquad \text{and}\\
\omega_2 &= \tfrac{3z_0}{\xi}.
\end{aligned}$$

Equation (1) representing the expected change in Y_i for processing a clutch in Phase 1 forms the basis of a differential equation. The differential equation suggested is

$$\frac{dz_i}{dx} = \chi_i, \qquad (0 \le i \le 2). \tag{5}$$

Here, differentiation is with respect to x and xn represents the number, t, of clutches.

Equation (2) representing the expected increase in the size of $\mathcal{C}$ after processing a clutch in Phase 1 and writing $C(t) = nz(t/n)$ suggests the differential equation for z as

$$\frac{dz}{dx} = \frac{3z_0}{\xi}. \tag{6}$$

For Phase 2, Equation (3) representing the expected change in Y_i for processing a clutch suggests the differential equation

$$\frac{dz_i}{dx} = \tau_i + \frac{\omega_2}{1-\omega_1}\chi_i, \qquad (0 \le i \le 2). \tag{7}$$

Equation (4) representing the increase in the size of $\mathcal{C}$ after processing a clutch in Phase 2 suggests the differential equation

$$\frac{dz}{dx} = \frac{3z_0}{\xi}\left(1 + \frac{\omega_2}{1-\omega_1}\right). \tag{8}$$

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

$$z_0(0) = 1, \quad z_1(0) = 0, \quad z_2(0) = 0 \quad \text{and} \quad z(0) = 0.$$

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 [14] to show that during each phase, the functions representing the solutions to the differential equations almost surely approximate the variables Y_i and C with error $o(n)$. 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 $(u_1, \ldots, u_j)$ and $(v_1, \ldots, v_j)$ in W. Note that $\max_{1 \le i \le j} |u_i - v_i|$ is the distance between $(u_1, \ldots, u_j)$ and $(v_1, \ldots, v_j)$ in the ℓ^∞ metric.

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

$$(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \notin W.$$

The following is a restatement of [14, Theorem 6.1]. We refer the reader to that paper for explanations, and to [12] 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*

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

(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 $l \le a$

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

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 $t > -\epsilon$, $\xi > \epsilon$, $z_1 > \epsilon$, $z > -\epsilon$ and $z_i < 1+\epsilon$ $(0 \leq i \leq 2)$.

For part (i) of Theorem 2 we must ensure that $Y_i(t)$ does not change too quickly throughout the process. As a clutch in Phase 1 consists of just one operation, the expected change in any of the variables Y_i for a clutch is at most 3. So part (i) of Theorem 2 holds with $\beta = 3$ and $\gamma = 0$. Equations (1) and (2) verify part (ii) of Theorem 2 for a function λ_1 which goes to 0 sufficiently slowly. (Note in particular that since $\xi > \epsilon$ inside W, the assumption that $s > \epsilon n$ used in deriving these equations is justified.) Part (iii) of Theorem 2 ensures that the rate of change of the variables does not change too quickly in time. By the definition of the phase and the domain W, it may be verified that the functions derived from equations (1) and (2) are continuous on W and its boundary. This implies that the functions are uniformly continuous. From this, the Lipschitz property of the functions required by Theorem 2 part (iii) may be deduced.

The Lipschitz condition in Theorem 2 part (iii) prevents us from choosing a domain which extends to the natural end of the phase which may occur at some time t_2, say. We choose a domain which the variables will almost surely remain inside until time $t_1 = t_2 - \epsilon n$.

The conclusion of Theorem 2 therefore holds for the process within Phase 1. This implies (taking $\lambda = o(1)$ tending to 0 sufficiently slowly) that with probability

$$1 - O(\lambda^{-1} \exp(-n\lambda^3)),$$

the random variables Y_i and C a.a.s. remain within $O(\lambda n)$ of the corresponding deterministic solutions to the differential equations (5) and (6) until a point arbitrarily close to where it leaves the set W. Choosing, for example, $\lambda = n^{-1/4}$, makes this success probability $1 - o(1)$.

Computing the ratio dz_i/dz gives

$$\frac{dz_i}{dz} = \frac{\xi \chi_i}{3z_0}, \qquad (0 \leq i \leq 2)$$

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 $z_1 = \epsilon$ (for $\epsilon = 0$ this would approximately be when $z \geq 0.5437$). At this point, we may formally define Phase 1 as the period of time from time t_0=0 to the time t_1 such that $z = t_1/n$ is the solution of z_1=ϵ.

From the point in Phase 1 after which Theorem 2 does not apply until the start of Phase 2, the change in each variable per step is bounded by a constant. Hence, letting ϵ tend to 0 sufficiently slowly, in $o(n)$ steps the change in the variables Y_i and C is $o(n)$.

For Phase 2 and for arbitrarily small $\epsilon > 0$, define W' to be the set of all (t, z_0, z_1, z_2, z) for which $t > t_1 + \epsilon$, $\xi > \epsilon$, $\omega_1 < 1 - \epsilon$, $z > -\epsilon$ and $z_i < 1 + \epsilon$ $(0 \leq i \leq 2)$. Also define $\widehat{W'}$ to be the vectors for which $z_1 \geq 0$, $z_2 \geq 0$ and $z_1 + z_2 > 0$. W' defines a domain for the variables t, z_i and z so that Theorem 2 may be applied to the process within Phase 2 (with time shifted by subtracting t_2).

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 [14, page 141]. So part (i) of Theorem 2 holds with $\beta = n^\epsilon$ and $\gamma = n^{-K}$. Equations (3) and (4) verify part (ii) for a function λ_1 which goes to 0 sufficiently slowly. (Note that 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.) The Lipschitz property of the functions required by Theorem 2 part (iii) may be deduced in a similar manner to those for Phase 1.

The conclusion of Theorem 2 therefore holds for the process within Phase 2. This implies that with probability

$$1 - O(n^{1-K} + n^\epsilon \lambda^{-1} \exp(-n^{1-3\epsilon}\lambda^3)),$$

the random variables Y_i and C a.a.s. remain within $O(\lambda n)$ of the corresponding deterministic solutions to the differential equations (7) and (8) until a point arbitrarily close to where it leaves the set W', or until $t = T_{\widehat{W}}$ if that occurs earlier. Note that the latter may only occur when the algorithm has completely processed a component of the graph and a random cubic graph is a.a.s. connected. Choosing $K = 2$ and $\lambda = n^{\epsilon - 1/4}$, say, leads to a success probability of $1 - o(1)$.

We compute the ratio dz_i/dz, and we have

$$\frac{dz_i}{dz} = \frac{\tau_i + \frac{\omega_2}{1-\omega_1}\chi_i}{\frac{3z_0}{\xi}\left(1 + \frac{\omega_2}{1-\omega_1}\right)}, \qquad (i \in \{0,1,2\})$$

where, again, differentiation is with respect to z and all functions can be taken as functions of z. By solving this we find that the solution hits a boundary of W' at $\xi = \epsilon$.

From the point in Phase 2 after which Theorem 2 does not apply until the completion of the algorithm, the change in each variable per step is bounded by a constant. Hence, letting ϵ tend to 0 sufficiently slowly, in $o(n)$ steps the change in the random variables Y_i and C is $o(n)$.

The differential equations were solved using a Runge-Kutta method, giving $\xi = \epsilon$ at $z < 0.5854$. This corresponds to the size of the connected dominating set (scaled by $1/n$) when all vertices are used up, thus proving the theorem. □

References

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

[2] R. Diestel. *Graph Theory.* Springer-Verlag, 1997.

[3] W. Duckworth. *Greedy Algorithms and Cubic Graphs.* Doctoral Thesis, Department of Mathematics and Statistics, The University of Melbourne, 2001.

[4] W. Duckworth and N.C. Wormald. Linear Programming and the Worst-Case Analysis of Greedy Algorithms on Cubic Graphs. *Submitted.*

[5] G. Galbiati, F. Maffioli, and A. Morzenti. A Short Note on the Approximability of the Maximum Leaves Spanning Tree Problem. *Information Processing Letters*, **52**(1):45–49, 1994.

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

[7] J.R. Griggs, D.J. Kleitman, and A. Shastri. Spanning Trees with Many Leaves in Cubic Graphs. *Journal of Graph Theory*, **13**(6):669–695, 1989.

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

[9] Hsueh-I Lu and R. Ravi. Approximating Maximum Leaf Spanning Trees in Almost Linear Time. *Journal of Algorithms*, **29**(1):132–141, 1998.

[10] J.A. Storer. Constructing Full Spanning Trees for Cubic Graphs. *Information Processing Letters*, **13**(1), 8–11, 1981.

[11] R. Solis-Oba. 2-Approximation Algorithm for finding a Spanning Tree with Maximum Number of leaves. In *Proceedings of the Sixth European Symposium on Algorithms*, Venice, 1998. Lecture Notes in Computer Science, **1461**:441–452, Springer, 1998.

[12] N.C. Wormald, Differential Equations for Random Processes and Random Graphs. *Annals of Applied Probability*, **5**:1217–1235, 1995.

[13] N.C. Wormald. Models of Random Regular Graphs. In *Surveys in combinatorics*, Canterbury, 1999, pages 239–298. Cambridge University Press, 1999.

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