Analysis of greedy algorithms on graphs with bounded degrees

Nicholas C. Wormald*
Department of Combinatorics and Optimization
University of Waterloo
Waterloo ON
Canada N2L 3G1
nwormald@uwaterloo.ca

Abstract

We give a general result on the average-case performance of certain greedy algorithms. These are derived from algorithms in which the possible operations performed at each step are prioritised. This type of prioritisation has occurred in previously studied heuristics for finding large subgraphs with special properties in random regular graphs, such as maximum independent sets and minimum dominating sets. The approach in this paper eliminates some of the complications caused by prioritisation. The main results apply in general to random graphs with a given degree sequence.

1 Introduction

The following greedy algorithm for finding independent sets was analysed for random d-regular graphs in [9] (and also in the particular case $d = 3$ by Frieze and Suen [6]). Start with a d-regular graph G. Pick any vertex v at random from those of minimum degree, and put v into the independent set S. Delete v and its neighbours from G (as they cannot be added to S). Repeat this step until there are no vertices left, at which stage S is an independent set of the original graph G.

The asymptotic size of the independent set produced by this algorithm was determined a.a.s. (*asymptotically almost surely*, denoting that the probability tends to 1 as $n \to \infty$). This gives the best asymptotically almost sure lower bound on the size of the largest independent set in a random d-regular graph which is known for small d, and possibly for all d. (The known bounds are not easily compared, due to the difficulty of calculating them.)

*Research supported by the Australian Research Council and by the Centre de Recerca Matematica, Bellatera, Spain. This research was carried out while the author was affiliated with the Department of Mathematics and Statistics, University of Melbourne

Actually, the procedure produces an independent dominating set, and the analysis therefore gives, additionally, an asymptotically almost sure upper bound on the size of a minimum dominating set, and moreover independent dominating set. For the case $d = 3$, a much smaller bound is obtained by the algorithm used in [4]. That algorithm similarly repeatedly chooses v of minimum degree, but v is not always placed in the dominating set; depending on the degrees of the neighbours of v, in some cases a neighbour of v is placed in the dominating set. In either case, we refer to the chosen vertex v as the one being *processed*. These algorithms are called *degree-greedy* because the vertex processed is chosen from those with the lowest degree.

Independent dominating sets are commonly known as *maximal independent sets*. We are interested here in their dominating property, and it is just a bonus that they are also independent. So we call the smallest one in a given graph a minimum independent dominating set, rather than minimum maximal independent set.

The two algorithms mentioned above are analysed in similar ways. Letting Y_i denote the number of vertices of degree i ($i = 1, \ldots, d$), 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. In both [9] and [4], the analysis has major complications, discussed further in Section 2, arising from the fact that priority is given to vertices currently of minimum degree. We call such an algorithm *prioritised* because vertices in some class are given priority of selection as v.

The main object of the present paper is to examine a rather different approach to the problems addressed by these algorithms and related ones. The new approach is to analyse associated algorithms which entirely avoid prioritising, by using a randomised mixture of choices of the processed vertex degree in the general step. The particular mixture used for any step will be prescribed in advance but will change over the course of the algorithm in order to approximate the prioritised algorithm. This approach was in some sense inspired by the proposal of Zito [13], to prescribe deterministically at the beginning of the algorithm the degree of the vertex to be processed at each step. However, unlike that approach, we attempt to use the randomised choice of degree to obtain the same result as a degree-greedy algorithm, since the best-known algorithms for these problems are degree-greedy.

We call the algorithms being introduced here *deprioritised* since there is no prioritisation of the vertex degree, and yet they are designed to approximate the results of a given prioritised algorithm. A more precise definition is in Section 3. Note that many deprioritised algorithms can be derived by using the operations of the original (prioritised) algorithm. The first use of randomised choices between operations, in a related setting, was by Achlioptas [1]. His algorithm "mTT" contains some deprioritisation, but it also contains a prioritised choice. The net effect is quite different from what is achieved with the fully deprioritised algorithms in the present paper.

This paper has objectives at several levels. One is to show that the intuitive idea referred to above—averaging the operations according to how often they are performed in the degree-greedy algorithm—gives the correct answer. The main object is to investigate the requirements of making this intuitive idea rigorous. Another goal is to actually reduce the number of conditions required to check. This might be hard to see, in a cursory comparison of this paper with [9] and [4], but the justifications in those papers

require checking more complex conditions regarding derivatives. A further objective is to present a more general result. This will simplify the issues when analysing further algorithms of this type; such arguments in [9] and [4] require steps involving branching processes and large deviation inequalities, and some of the details given are quite sketchy. Finally, the approach taken shows indications of being very useful elsewhere, even when the main results given here do not apply. This is explained more fully in Section 5. The new results can also be applied to random graphs with given degrees which are uniformly bounded.

In Section 2 we analyse the degree-greedy algorithms for independent and dominating sets, using a result obtained via prioritised algorithms. This is proved in Section 3 from a more general result which in is in turn proved in Section 4.

2 Theorem and examples for d-regular graphs

Throughout this section we assume $d \geq 3$ is fixed (though the basic definitions apply for $d \geq 1$), with dn even. For all asymptotics we take $n \to \infty$. We consider here, in more detail, the greedy algorithms on random d-regular graphs for independent and dominating sets described in Section 1. This gives motivation for the setting of Theorem 1, which is tailored for algorithms like these. It provides the same asymptotic result that we expect a degree-greedy algorithm to produce. We then apply it to the independent and dominating set algorithms, and obtain the same lower bounds on maximum independent sets, and upper bound on minimum independent dominating sets, as obtained in [9, 4]. Its proof, given in the rest of the paper, uses analysis of the deprioritised algorithm which we associate with a degree-greedy algorithm.

The standard model for random d-regular graphs is as follows. See Bollobás [2] and the author's survey [10] for a thorough discussion of this model and the assertions made about it here, as well as other properties of random regular graphs. Take a set of dn points in n buckets labelled $1, 2, \ldots, n$, with d points in each bucket, and choose uniformly at random 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. This pseudograph is a simple graph (i.e. has no loops or multiple edges) if no pair contains two points in the same bucket, and no two pairs contain four points from just two buckets. The d-regular simple graphs on n vertices graphs all occur with equal probabilities.

With probability asymptotic to $e^{(1-d^2)/4}$, the pseudograph corresponding to the random pairing in $\mathcal{P}_{n,d}$ is simple. It follows that, 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.

As in [9] and [11], we redefine this model slightly by specifying that the pairs are chosen sequentially. The first point in a random pair can be selected using any rule whatsoever, as long as the second is chosen uniformly at random from the remaining 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 can be combined with the algorithm (as in [4, 9, 11]). This can be explained alternatively as follows. Suppose that the pairing generation consists of a sequence of operations $\mathrm{op}_0, \mathrm{op}_1, \ldots$, each exposing at least one of the pairs. An algorithm which examines edges in the same order as for the pairing generation can be incorporated into the pairing generation by extending the definition of the operations to do whatever other tasks the algorithm needs to carry out. (An example is below.)

The algorithm being referred to acts upon the final (pseudo)graph of the generation process. It is convenient to regard the operations as sequentially *deleting* the exposed pairs (edges) from this graph. For this reason, we refer to it as the *deletion* algorithm being carried out, to distinguish it from the pairing generation. At each point, the graph in the deletion algorithm contains all the edges of the final graph which have *not* yet been exposed.

For example, the degree-greedy independent set algorithm from [9] and [11] can be described in terms of operations incorporated into the pair generation, as follows. A set S and the pairing are initially empty. Then for integer $t \geq 0$, the operation op_t randomly selects a bucket v with the maximumum degree, the degree of a bucket being the number of points in that bucket which are in exposed pairs. (This is equivalent to a vertex of *minimum* degree in the graph in the deletion algorithm.) It then adds v to S and exposes all the pairs involving points in v, and next exposes the pairs of all points in all neighbouring buckets (i.e. buckets containing pairs in common with v). Any other bucket attaining degree d is also added to S. These correspond to vertices which become isolated in the deletion algorithm without ever being v or one of its neighbours. We call these vertices, and the corresponding buckets, *accidental* isolates.

The setting of Theorem 1 requires a number of general definitions. The initial pairing is empty, denoted by G_0. We consider processes in which each operation op_t is one of Op_i, $i = 1, \ldots, d$, where Op_i consists of selecting a bucket (vertex) v of degree $d - i$ in G_t uniformly at random, and then applying some specified set of (usually randomised) tasks, to obtain G_{t+1}. (We use Op_i for buckets of degree $d-i$ because these correspond to vertices of degree i in the corresponding deletion algorithm.) For the general setting, we do not insist on always performing an operation on the bucket of maximum degree: the tasks specified in Op_i must be such that the operation can always be completed, provided there is some bucket of degree at least $d - i$. A subset S of $V(G) \cup E(G)$ is selected during the operations, with $S_0 = \emptyset$ initially, and $S = S_t$ for the pairing G_t.

For $0 \leq i \leq d$, let $Y_i = Y_i(t)$ denote the number of buckets of degree $d - i$ in G_t (in agreement with the convention of naming Op_i). This is the number of vertices of degree i in the graph in the deletion algorithm. The number of buckets of degree d (vertices of degree 0) is

$$Y_0 = n - \sum_{i=1}^{d} Y_i, \tag{2.1}$$

so this does not need to be calculated (but is useful in discussions). Also let Y_{d+1} denote cardinality of the set S_t.

Let j denote $d+1$. (If we wished to keep track of other sets at the same time, we could define $j > d+1$ and other variables $Y_{d+2}, \ldots, Y_j$.) The process $\mathbf{Y}(t) = (Y_1(t), \ldots, Y_j(t))$ need not be Markovian; i.e., the probability distribution of $\mathbf{Y}(t)$ need not be determined

from the vector $\mathbf{Y}(t-1)$. But we require it to be so, in an asymptotic sense. Accordingly, assume that the expected change in Y_i, in going from G_t to G_{t+1}, conditional upon G_t and op_t, is determined approximately, depending only upon t, op_t, and $Y_1(t), \ldots, Y_j(t)$. In some sense, this is a measure of the rate of change of Y_i. We express the assumption by asserting that for some fixed functions $f_{i,r}(x, \mathbf{y}) = f_{i,r}(x, y_1, \ldots, y_j)$,

$$\mathbf{E}\Big(Y_i(t+1) - Y_i(t) \mid G_t \wedge \{\mathrm{op}_t = \mathrm{Op}_r\}\Big) = f_{i,r}(t/n, Y_1/n, \ldots, Y_j/n) + o(1) \qquad (2.2)$$

for $i = 1, \ldots, j$, $r = 1, \ldots, d$ such that $Y_r(t) > 0$. The convergence in $o(1)$ is uniform over all appropriate choices of t and G_t as functions of n with certain restrictions on G_t which will be specified in the forthcoming theorem. Uniformity over r and i then follows, since there are finitely many possibilities for these two variables.

To motivate the remaining general definitions, it is helpful to consider the typical behaviour of the degree-greedy independent set algorithm. We now discuss this in terms of the deletion algorithm described above. The initial graph is a d-regular graph on n vertices. The first step must apply $\mathrm{op}_0 = \mathrm{Op}_d$. Typically G_1 has some lower degree vertices, so the next step is determined by its minimum degree. Both Op_d and Op_{d-1} typically produce vertices of degree $d-1$ but none of lower degrees when Y_{d-1} is small (say $o(n)$), so the second step normally inolves Op_{d-1}, as does the next, and this remains so until a vertex of lower degree, say $d-2$, is produced. This causes a temporary hiccup, with an Op_{d-2}, followed by more steps of Op_{d-1}. When vertices of degree $d-1$ become plentiful, vertices of lower degree are more commonly created, and the hiccups occur more often. In this way, the prioritisation causes a rather complicated situation.

Suppose that at some step t in the process, an Op_{d-1} creates, in expectation, α vertices of degree $d-2$, and an Op_{d-2} decreases the number of vertices of degree $d-2$, in expectation, by τ. Then we expect each Op_{d-1} to be followed by (on average) α/τ steps of Op_{d-2}. At some stage τ may fall below 0, at which point the vertices of degree $d-2$ begin to build up and do not decrease under repeated applications of Op_{d-2}. Then vertices of degeree $d-2$ take over the role of vertices of degree $d-1$, and we say *informally* that the first phase of the process has finished and the second has begun. (The definition is necessarily imprecise because the endpoint of a phase is hard to pin down when observing the performance of one run of the algorithm.) The process may continue through further phases; typically, the kth phase begins with an increasing abundance of vertices of degree $d-k$. Note that by the assumptions above, the asymptotic values of α and τ in the first phase are the cases $k = 1$ of the general definitions

$$\begin{aligned} \alpha_k(x, \mathbf{y}) &= f_{d-k-1,d-k}(x, \mathbf{y}), \\ \tau_k(x, \mathbf{y}) &= -f_{d-k-1,d-k-1}(x, \mathbf{y}), \end{aligned} \qquad (2.3)$$

where

$$x = \frac{t}{n}, \qquad \mathbf{y}(x) = \frac{\mathbf{Y}(t)}{n}. \qquad (2.4)$$

During phase k, what is the expected trajectory of the variables Y_i? The limiting behaviour of the scaled version Y_i/n will (we shall see) be independent of n. Since each Op_{d-k} is followed (on average) by α_k/τ_k steps of Op_{d-k-1}, we expect the proportion of steps involving an operation of the former type to be $1/(1+\alpha_k/\tau_k) = \tau_k/(\tau_k+\alpha_k)$, and

of the latter type to be $\alpha_k/(\tau_k+\alpha_k)$. This *suggests* that, if y_i as prescribed in (2.4) were a differentiable function of a real variable, its derivative would satisfy

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = F\left(x, \mathbf{y}, i, k\right) \tag{2.5}$$

where

$$F\left(x, \mathbf{y}, i, k\right) = \begin{cases} \frac{\tau_k}{\tau_k+\alpha_k} f_{i,d-k}\left(x, \mathbf{y}\right) + \frac{\alpha_k}{\tau_k+\alpha_k} f_{i,d-k-1}\left(x, \mathbf{y}\right) & k \leq d-2 \\ f_{i,1}\left(x, \mathbf{y}\right) & k = d-1. \end{cases} \tag{2.6}$$

Our assumptions will ensure that the phases proceed in an orderly fashion, and that the last possible phase is $k = d-1$, in which all operations are Op_1.

Having described the typical behaviour of the degree-greedy independent set algorithm, we may point out some of the difficulties of its analysis. One problem is that the expected change in the variables depends on what this minimum degree is, and this follows a random process itself. It would require justification to argue as above about the proportion of steps involving an operation of a given type. The other problem is that the analysis in between phases does not proceed very smoothly.

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

$$\mathcal{D}_\epsilon = \{(x, \mathbf{y}) : 0 \leq x \leq d,\ 0 \leq y_i \leq d \text{ for } 1 \leq i \leq j,\ y_d \geq \epsilon\} \tag{2.7}$$

for some prechosen value of $\epsilon > 0$. (The upper bounds on x and y_i are chosen so as to contain all conceivably relevant $(x, \mathbf{y})$. A positive lower bound on y_d is included to avoid the singularities of the functions $f_{i,r}$ in the applications we will consider.) The behaviour of the process will be described in terms of the function $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}(x) = (\tilde{y}_1(x), \ldots, \tilde{y}_j(x))$ defined as follows, with reference to an initial value $x = x_0 = t_0/n$ of interest:

> $\tilde{y}_i(x_0) = Y_i(t_0)/n$, $i = 1, \ldots, j$, and inductively for $k \geq 1$, $\tilde{\mathbf{y}}$ is the solution of (2.5) 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. (2.8)

The interval $[x_{k-1}, x_k]$ represents phase k, and the termination condition $\tilde{y}_{d-k} = 0$ is necessary to ensure that the process does not revert to the conditions of phase $k-1$. (This, if it did occur, could still be analysed by similar methods, but to permit this would make the descriptions of the phases difficult; it does not seem to occur in practice for the algorithms of interest here.) Typically it will eventuate that $\tilde{y}_{d-k}(x_{k-1}) = 0$ but $\tilde{y}_{d-k}(x) > 0$ for x greater than, but close to, x_{k-1}, which permits phase k to endure for a non-empty interval $[x_{k-1}, x_k]$, provided τ_k stays positive on such an interval. We require this inductive definition of $\tilde{\mathbf{y}}$ to continue for phases $k = 1, 2, \ldots, m$, where

> m denotes the smallest k for which either $k = d-1$, or any of the termination conditions for phase k in (2.8) hold at x_k apart from $x_k = \inf\{x \geq x_{k-1} : \tau_k \leq 0\}$. (2.9)

To simplify the discussion, we will impose conditions to ensure that the intervals representing phases $1, 2, \ldots, m$ in the definition of $\tilde{\mathbf{y}}$ are nonempty. These conditions are

$$\tau_k > 0 \text{ and } \tau_k + \alpha_k > \epsilon \text{ at } (x_{k-1}, \tilde{\mathbf{y}}(x_{k-1})) \quad (1 \le k \le \min\{d-2, m\}), \tag{2.10}$$

$$\begin{aligned} f_{d-1,d-1} &> 0 \text{ at } (x_0, \tilde{\mathbf{y}}(x_0)), \\ f'_{d-k,d-k}\tau_k + f_{d-k,d-k-1}f'_{d-k-1,d-k} &> 0 \text{ at } (x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))^+ \quad (1 < k \le \min\{d-2, m\}), \\ f'_{d-k,d-k} &> 0 \text{ at } (x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))^- \quad (1 < k \le m), \\ f'_{1,1} &> 0 \text{ at } (x_{d-2}, \tilde{\mathbf{y}}(x_{d-2}))^+ \quad \text{if } m = d-1, \end{aligned} \tag{2.11}$$

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 state the first main result of this paper, to be proved later using deprioritised algorithms.

Theorem 1 *Let $d \ge 3$, for $1 \le i \le d$ let $Y_i(t)$ denote the number of buckets of degree $d-i$ in G_t, and let $Y_{d+1}(t)$ denote $|S_t|$. Assume that for some fixed $\epsilon > 0$ the operations Op_r satisfy (2.2) 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 S (i.e. $|S_{t+1}| - |S_t|$), 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 (2.7);

(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 (2.8), set $x_0 = 0$, define m as in (2.9), and assume that (2.10) and (2.11) both hold. Then there is a randomised algorithm on $\mathcal{P}_{n,d}$ for which a.a.s. there exists t such that $|S_t| = n\tilde{y}_{d+1}(x_m) + o(n)$ 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 $x_{k-1} \le x \le x_k$, $1 \le i \le d-k-1$ $(1 \le k \le m)$.

Note The hypotheses of the theorem relate to the behaviour of all the operations, but it is really only the behaviour of Op_δ which matters, where δ is the minimum vertex degree of G_t. This should be expected, since in the prioritised algorithm, it is only the behaviour of the operation with highest priority which counts; the other operations may not even be defined! (See the discussion of Example 2 below.) Nevertheless, we insist on having Op_r defined for $r > \delta$ (provided $Y_r > 0$); otherwise, it would be much more awkward to state the hypotheses which would correctly lead to an equivalent conclusion.

Example 1. Independent sets in random d-regular graphs.

Consider the degree-greedy algorithm for independent sets from [9, 11] described above. Here, in the specification of Op_r (which first selects a random bucket of degree

$d-r$), the set of randomised tasks consists of exposing all the pairs involving points in v and points in all neighbouring buckets, and adding v and any accidental isolates to S.

We may verify the hypotheses of Theorem 1. It is shown in [9] that (2.2) holds when $Y_d(t) > \epsilon n$ (for any $\epsilon > 0$) with the definition

$$f_{i,r} = -\delta_{ir} - r\frac{iy_i}{s} + r((i+1)y_{i+1}\delta_{i+1\le d} - iy_i)\sum_{\ell=2}^{d}\frac{(\ell-1)\ell y_\ell}{s^2} \tag{2.12}$$

for $0 \le i \le d$, where y_0 is defined as $1 - \sum_{\ell=1}^{d} y_\ell$ in accordance with (2.1) and (2.4). Here, for any statement Q, define $\delta_{\mathrm{Q}} = 1$ if Q is true and 0 otherwise, and let $\delta_{i\ell}$ denote $\delta_{i=\ell}$. Also $s = \sum_{\ell=1}^{d} \ell y_\ell$ (noting that Y_i here was denoted by Y_{d-i} in [9]). It follows that (2.2) also holds for $i = d+1$ with $f_{d+1,r}$ defined as $1 + f_{0,r}$, since in each Op_r, an extra vertex is added to the independent set S and the expected number of accidental isolates is $f_{0,r}$ as defined in (2.12).

It may help to sketch, nonrigorously, the derivation of (2.12). (See [9] for the full story). The bucket v has degree $d-r$ before the operation and 0 afterwards, hence the term $-\delta_{ir}$. The probability that when a pair is exposed, the other point is in a bucket of degree $d-i$, is asymptotic to iY_i/σ where $\sigma = \sum_{\ell=1}^{d} \ell Y_\ell$. Thus riy_i/s stands for the expected number, of the r buckets found adjacent to v, which have degree i. The rest of the formula comes from the expected changes due to the buckets of "distance" 2 from v. For each bucket of degree $d-\ell$ adjacent to v (the expected number of which is $\ell Y_\ell/\sigma$) there are $(\ell-1)$ pairs exposed (we can ignore pairs exposed from both ends — they are rare) and the expected number of buckets of degree $d-i$ each of these reaches is iY_i/σ. This contributes negatively to the expected change in Y_i, whilst buckets of degree $d-i-1$ which are reached contribute positively. This explains (2.12) (omitting justification of the omission of some insignificant terms).

Hypothesis (i) of the theorem is immediate since in any operation only the pairs involving points in one bucket and its neighbours are exposed, and a bounded number of vertices are added to S (as there are certainly less than d^2 accidental isolates). The functions $f_{i,r}$ satisfy (ii) because from (2.12) their (possible) singularities satisfy $s = 0$, which lies outside $\mathcal{D}_\epsilon$ since in $\mathcal{D}_\epsilon$, $s \ge y_d \ge \epsilon$. Hypothesis (iii) follows from (2.12) again using $s \ge y_d \ge \epsilon$ and the boundedness of the functions y_i (which follows from the boundedness of $\mathcal{D}_\epsilon$). Thus, defining $\tilde{\mathbf{y}}$ as in (2.8) with $t_0 = 0$, $Y_d(0) = n$ and $Y_i(0) = 0$ for $i \ne d$, we may solve (2.5) numerically to find m, verifying (2.10) and (2.11) at the appropriate points of the computation. It turns out that these hold for each d which was treated numerically in [9], and that in each case $m = d-2$, for sufficiently small $\epsilon > 0$. For such ϵ, the value of $\tilde{y}_{d+1}(x_m)$ may be computed numerically, and then by Theorem 1, this is the asymptotic value of the size of the independent set S at the end of some randomised algorithm. So the conclusion is that a random d-regular graph a.a.s. has an independent set of size at least $n\tilde{y}_{d+1}(x_m) + o(n)$. Note also that (by the theorem) $\tilde{y}_i(x) \equiv 0$ in phase k for $1 \le i \le d-k-1$, and by the nature of the differential equation, $\tilde{y}_i(x)$ will be strictly positive for $i > d-k$. So by (2.8) and (2.9), the end of the process (for ϵ arbitrarily small) occurs in phase $d-2$ when either $\tau_k + \alpha_k \le \epsilon$ or $\tilde{y}_2$ becomes 0. Numerically, we find it is the latter. This is numerically more stable as a check for the end of the process than checking when $\tilde{y}_d$ reaches 0, since the derivative of the latter is very small.

In [9], the differential equations computed were actually different (but equivalent) and only the algorithm with priority constraints was considered. The resulting bounds for independent sets were the same as the argument above gives. However, the method of analysis required verification of more complicated conditions than the use of Theorem 1.

Note that in applying this theorem, the type of algorithm being used is immaterial. All that is important is the operations and their expected effects. It no longer matters that the differential equations mimic the algorithm which repeatedly selects the vertex v of minimum degree, adds v to the independent set, and then deletes v and its neighbours from the graph.

Example 2. Independent dominating sets in random 3-regular graphs.

A similar randomised greedy algorithm was used in [4] to obtain upper bounds on the size of a minimum independent dominating set (minimum maximal independent set) in a random cubic graph. In this algorithm a random vertex v of minimum degree is selected, and a vertex u is added to the dominating set S where u is either v or a neighbour of v. The determination of u depends on the degrees of v and its neighbours: if $d(v) = 3$, or $d(v) = 2$ and both neighbours have degree 2, then $u = v$; otherwise, u is a neighbour of v of maximum possible degree (randomly chosen if there is a tie). Then u and its neighbours are deleted, any resulting vertices of degree 0 (accidental isolates) are added to S, and the step is repeated.

For the analysis here, the operation is called Op_r if r is the degree of v in the deletion algorithm; i.e. the corresponding bucket has degree $3-r$ in the current pairing. Op_2 was only defined in [4] in the case that the neighbours of v do not have degree 1 (which was sufficient for the purpose there, since this is always true when the degree-greedy priority rule is used). It was shown that under this assumption, for $r = 1$ and 2, $1 \le i \le 3$, (2.2) holds with

$$\begin{aligned} f_{i,1} &= \frac{-iy_i + (6y_3 + 2y_2)\mu_i}{s} - \delta_{i1}, \\ f_{i,2} &= \frac{9y_3^2(-2\delta_{i3} + 2\mu_i) + 12y_3y_2(-\delta_{i3} - 2\delta_{i2} + \delta_{i1} + 2\mu_i) + 4y_2^2(-3\delta_{i2} + 2\rho_i)}{s^2} \end{aligned}$$

where

$$\begin{aligned} \mu_i &= \frac{-iy_i + (6y_3 + 2y_2)\rho_i}{s}, \\ \rho_i &= \frac{-iy_i + (i+1)y_{i+1}\delta_{i+1\le 3}}{s}, \quad 1 \le i \le 3, \end{aligned}$$

and δ and s are as in (2.12). The justification of these equations is similar to those in the previous example. Moreover, it is easy to show that the corrsponding equation for $r = 3$ is

$$f_{i,3} = -\delta_{i3} - 3\mu_i$$

(assuming the neighbours of the chosen vertex do not have degree less than 3). We now need to remove the assumption that the neighbours of v do not have degree less than r in Op_r ($r = 2$ and 3). An easy way to do this is to extend the definition of the operations so that, if v does have any such neighbours, the operation just exposes all

points in v, and adds any accidental isolates to the dominating set. Then the formulae for $f_{i,2}$ and $f_{i,3}$ above are modified by the addition of rational functions, of a type similar to the present functions, and whose denominator is a power of s. (This much is fairly easy to see; getting the correct functions only a little more work.) In the case of $f_{i,2}$, the extra terms all have y_1 as a factor, accounting for the probability that v has a neighbour of degree 1. Since in the solution $\tilde{y}_1 = 0$ (by the last statement in Theorem 1) and the differential equation does not involve $f_{i,3}$, these extra terms can be ignored. The final variable is y_4, for the size of the dominating set constructed by the algorithm. For each operation, Y_4 gains 1 plus the number of accidental isolates. It is shown in [4] that $f_{4,1} = 1 + (6y_3 + 2y_2)^2 y_1/s^2$ and $f_{4,2} = f_{4,3} = 1$ (again ignoring the extra terms for neighbours of degree less than r, which does not affect the result, or its validity, for the same reasons as before). Verification of the hypotheses of Theorem 1 is much as in the previous example, though condition (2.10) and the first in (2.11) are easily checked directly from the initial conditions, so the only condition requiring a special numerical check is the last in (2.11), that $f'_{1,1} > 0$ at $(x_1, \tilde{\mathbf{y}}(x_1))$. Numerical solution of the differential equation (2.5) (with $\epsilon \approx 10^{-5}$) shows that $m = 2$, with $x_1 \approx 0.1419$, $\tilde{y}_2(x_1) \approx 0.219 > 0$ and $y_3 \approx 0.236$. We may substitute these values into the derivative of $f_{1,1}$ with respect to x along the solution, to see that $f'_{1,1} \approx 3.9$. Further numerical solution of the differential equation gives $\tilde{y}_4(x_2) \approx 0.27941$. Moreover, $\tilde{y}_i(x_2) \approx 0$ for $1 \le i \le 3$, and these variables represent all the vertices not yet dominated. So by Theorem 1, there is a randomised algorithm on $\mathcal{P}_{n,d}$ which at some point has $|S| \approx 0.27941n$ and has dominated virtually all of the vertices of the graph. Numerically, we find that $\tilde{y}_4(x_2) + \sum_{i=1}^{3} \tilde{y}_i(x_2) < 0.27942$. Thus, adding the remaining vertices into the dominating set (or some of them, as required in a greedy fashion) gives the almost sure upper bound $0.27942n$ on the size of the minimum independent dominating set in a random cubic graph. This is, because of the nature of the proof of Theorem 1, the same result as obtained in [4] for the degree-greedy algorithm.

3 Deprioritised algorithms

In this section we introduce the precise deprioritised algorithms to be used to approximate greedy algorithms. We state a general result, Theorem 2, and use it to prove Theorem 1. Theorem 2 is proved in the next section.

First, the algorithms described in the examples in Section 2, as well as some obvious variations, can be put into the following framework. (Most of the assumptions in Section 2 will be used again, as well as some introduced in this section.) Working in this generality enables us to isolate the arguments which do not depend on the pairings introduced in Section 2. Consider for each $n > 0$ a discrete-time Markov process $G_0, G_1, \ldots = \{G_t\}_{t \ge 0}$, so that $G_t = G_t(n)$. Here, G_0 may be a random function of n. For example, it may be a (partial) pairing as in Section 2, or a graph. Assume that each transition from G_t to G_{t+1} is an instance of one of a finite set of d *operations* $\mathrm{Op}_1, \ldots, \mathrm{Op}_d$. We denote the particular one of these applied to G_t to obtain G_{t+1} by op_t. In general, applying Op_i to G_t will itself involve some randomised steps. It many not be possible to apply every operation to every possible object G_t, but we assume there is a set of $j \ge d$ functions of G_t, $Y_1 = Y_1(t) = Y_1(G_t), \ldots, Y_j = Y_j(t) = Y_j(G_t)$, for

$t = 0, 1, \ldots$, such that

$$\text{the application of } \mathrm{Op}_i \text{ to } G_t \text{ is defined if } i \le d \text{ and } Y_i(G_t) > 0. \tag{3.1}$$

(Note that j is no longer set equal to $d+1$ as in Section 2.) We thus define Op_i to be *permissible* for G_t iff $Y_i > 0$. Note that in this setting, $Y_i(t)$ becomes a random variable, though for convenience we sometimes refer to its value on a given trajectory of the process.

Along with this, assume that for all instances of the process, for all $t \ge 0$

$$Y_i(t) \ge 0 \quad (1 \le i \le d). \tag{3.2}$$

The other variables, $Y_{d+1}, \ldots, Y_j$, are unconstrained.

For the degree-greedy algorithms mentioned in Section 1, the choice of operations is prioritised. Thus, at each step t, the operation to be performed is determined by the Y_i as follows: Op_δ must be applied to G_t, where $\delta = \min\{i : i \le d, Y_i(t) \ne 0\}$. Under these circumstances, as long as the $f_{i,j}$ are well enough behaved, we would *expect* the rate of change of Y_i to approximately equal $f_{i,\delta(G_t)}(x, \mathbf{y})$ with x and $\mathbf{y}$ as in (2.4). To help ensure good behaviour of the algorithm, we require that the Y_i and $f_{i,k}$ satsify Lipschitz conditions. Here, by f being *Lipschitz* on a domain $\mathcal{D}$, we mean that for some $C > 0$, for all $\epsilon' > 0$, $|f(\mathbf{x}) - f(\mathbf{x}_0)| < C\epsilon'$ whenever $|\mathbf{x} - \mathbf{x}_0| < \epsilon'$ with $\mathbf{x}$, $\mathbf{x}_0 \in \mathcal{D}$. We are primarily interested only in the following closed domain $\mathcal{D}_{\epsilon,M} \subseteq \mathbb{R}^{j+1}$ where ϵ and M are fixed positive constants to be defined in the particular applications:

$$\mathcal{D}_{\epsilon,M} = \{(x, \mathbf{y}) : 0 \le x \le M,\ |y_i| \le M\,(1 \le i \le j),\ 0 \le y_i\,(1 \le i < d),\ y_d \ge \epsilon\}. \tag{3.3}$$

Note that under the change of variables (2.4), $Y_i(t) \le Mn$ inside $\mathcal{D}_{\epsilon,M}$. The following restricts the Y_i rather more strongly than is necessary (see [11, Theorem 5.1]) but suffices for many purposes and makes for simpler proofs. The assumptions we use are:

(A1) for $1 \le i \le j$ and for some $C > 0$, $Y_i(t) < 9Mn/10$ always and $|Y_i(t+1) - Y_i(t)| < C$ always;

(A2) for all $1 \le r \le d$ and all i, $f_{i,r}$ is Lipschitz on $\mathcal{D}_{\epsilon,M}$.

We will also be assuming

$$\begin{gathered}\text{the derivative of each } f_{i,j} \text{ along a trajectory of (2.5) is Lipschitz} \\ \text{in some neighbourhood of } (x_k, \tilde{\mathbf{y}}(x_k))\ (1 \le k < m).\end{gathered} \tag{3.4}$$

In conjunction with (2.11), this will ensure that y_{d-k} grows at the beginning of phase k to ensure that the phase lasts for a significant time period, although the reasons for this may not be clear at present.

We remark that in the applications considered here, x_m turns out to be finite, and at this point $\tilde{y}_i$ approaches 0 for each $i \le d$.

Priority constraints cause annoyance during the analysis in [9, 11]. Instead of adhering to the priorities, consider another algorithm in which there is prescribed for each G_t a probability vector $\mathbf{p} = \mathbf{p}(n, x) = (p_1, \ldots, p_d)$ which depends only upon n and x. Here each p_i is nonnegative, and $\sum_i p_i = 1$. Given $\mathbf{p}$, the probability that the next step

applies Op_r to G_t is $p_r = p_r(n,x)$, recalling that $x = t/n$. From (2.2) this implies by linearity of expectation that

$$\mathbf{E}(Y_i(t+1) - Y_i(t) \mid G_t) \sim \sum_{r=1}^{d} p_r f_{i,r}(x, \mathbf{y}). \tag{3.5}$$

We call this new algorithm the *deprioritised* $\mathbf{p}$*-algorithm* corresponding to the original algorithm.

At each step until the algorithm terminates, $\mathbf{p}$ must be such that

$$\mathrm{Op}_i \text{ is permissible for } G_t \text{ for each } i \text{ such that } p_i > 0. \tag{3.6}$$

If this condition is ever violated for a particular t, we may define $G_{t+1} = G_t$ and say that the algorithm has become *stuck*.

To make this approach workable, we impose some extra conditions on the functions $f_{i,r}$. For the applications of present concern, the following is satisfied in a natural way, for the simple reason that when a vertex is removed from a random graph with given degree sequence, the degrees of its neighbours are determined approximately from the numbers of vertices of given degree. (It is still a stronger assumption than necessary to obtain useful results.)

(B) There exist positive constants C_1, C_2 and C_3 (which may depend on ϵ and M) such that $f_{i,r} \geq C_1 y_{i+1} - C_2 y_i$ on $\mathcal{D}_{\epsilon,M}$ for $1 \leq i < d$ and $r \neq i$, and $f_{i,r} \leq C_3 y_{i+1}$ on $\mathcal{D}_{\epsilon,M}$ for $1 \leq i < d$ and all r.

The following theorem gives a useful result even for algorithms applied to non-regular graphs (or other more general applications), since the error terms depend only on the maximum vertex degree of the initial graph. The key features of the proof are only required when $d \geq 3$, so this condition is imposed at the outset.

Theorem 2 *Let $\epsilon, M > 0$ and $d \geq 3$, and define $\mathcal{D}_{\epsilon,M}$ as in (3.3). Assume that the Markov process $\{G_t\}_{t\geq 0}$ has operations Op_i and variables $Y_i(G_t)$ satisfying (3.1) and (3.2), and that (2.2) holds whenever $(t/n, Y_1(t)/n, \ldots, Y_j(t)/n) \in \mathcal{D}_{\epsilon,M}$. Also assume the conditions (A1), (A2) and (B), as well as that $Y_i(0) = 0$ for $i < d$ and that*

$$Y_d(0) > c_0 n \text{ for some constant } c_0 > C\epsilon,\ c_0 < M \tag{3.7}$$

(where C is the constant in (A1)). Setting $\mathcal{D}_\epsilon = \mathcal{D}_{\epsilon,M}$, define $\tilde{\mathbf{y}}$ and x_k as in (2.8) with $x_0 = 0$, and m as in (2.9), and assume that (2.10), (2.11) and (3.4) all hold. Then the x_k are all distinct, and there is a choice of $\mathbf{p} = \mathbf{p}(n,x)$ such that, for the deprioritised $\mathbf{p}$-algorithm, a.a.s. $Y_i(t) = n\tilde{y}_i(t/n) + o(n)$ uniformly for $0 \leq t \leq x'_m n$ and $1 \leq i \leq j$, where $x_m - x'_m = o(1)$. Furthermore, for $1 \leq k \leq m$,

$$\tilde{y}_i(x) \equiv 0 \quad (x_{k-1} \leq x \leq x_k,\ 1 \leq i \leq d-k-1). \tag{3.8}$$

Notes

1 We define the statement that a.a.s. $A(t) = B(t) + o(n)$ to mean that there exists a function $\lambda'(n) = o(n)$ for which $|A(t) - B(t)| < \lambda'(n)$ a.a.s. Similar statements in this paper should be interpreted the same way.

2 If the functions $f_{i,r}$ are not Lipschitz on $\mathcal{D}_{\epsilon,M}$ but on some subset of it which corresponds to all feasible trajectories of the process, a similar theorem holds with virtually the same proof.

3 Bounds on the error in the functions computed when solving the differential equations numerically permit verification of (2.10) and (2.11) because they involve inequalities rather than equations (and in view of (3.4)). This implies that the distinctness of the x_k can be checked in the same way, since this follows from simple inequalities holding at each x_k ensuring that phase k does not immediately terminate. On the other hand, numerical verification of the existence of a particular x_k relies upon checking that the inequality in (2.8) which determines the end of a phase is satisfied sharply shortly after the end of the phase. Luckily, this happens in all applications so far considered.

4 The prioritised algorithm behaves similarly to the **p**-algorithm whose existence is shown in Theorem 2, but this statement does not follow easily from Theorem 2. In any case, for the applications of concern here, it is sufficient to know of *any* algorithm behaving in a given way asymptotically.

Theorem 1 is essentially the specialisation of Theorem 2 to degree-greedy algorithms on the pairing model; that is, algorithms on pairings in which the selection of the operation Op_i involves using the least i for which $Y_i > 0$.

Proof of Theorem 1 For this theorem, we have $j = d+1$. Let us check the hypotheses of Theorem 2, with $\epsilon > 0$ chosen to be suitably small.

First, (3.1) holds by the specification that Op_i can be performed whenever a bucket of degree i exists, and (3.2) is immediate from the definition of Y_i.

Let $M = d$, so that $\mathcal{D}_\epsilon = \mathcal{D}_{\epsilon,M}$ as in (3.3). Equation (2.2) certainly applies inside $\mathcal{D}_\epsilon$. The first part of condition (A1) holds by the fact that $Y_i \le dn/2 + n$ always, noting that $d \ge 3$. (For $i = d+1$ the value is at most the total number of vertices and edges in the graph, and for smaller i it is at most n, the number of vertices.) The second part holds by assumption (i) of the theorem's hypotheses. Condition (A2) holds since it follows from (ii) that the $f_{i,r}$ are analytic and have bounded derivatives on the domain $\mathcal{D}_{\epsilon,M}$, which is bounded. Condition (B) is given by (iii). Initially, $Y_d(0) = n$, so choosing $\epsilon > 0$ sufficiently small, the lower bound (3.7) on $Y_d(0)$ is satisfied. The graph is initially regular so $Y_i(0) = 0$ for $i < d$. Equations, (2.10) and (2.11), required for Theorem 2, are both asserted directly in the hypotheses of Theorem 1. Finally, (3.4) follows easily from (ii), since the derivative of each $f_{i,j}$ along any trajectory of (2.5) is a rational function of x and the y_i and their derivatives along that trajectory. By (2.6), these derivatives themselves are rational functions with no pole in $\mathcal{D}_{\epsilon,M}$ by the definition of x_k in (2.8). So by the conclusion of Theorem 2, there is a deprioritised algorithm for which $Y_{d+1}(t) = n\tilde{y}_{d+1}(t/n) + o(n)$ a.a.s., where $t = \lfloor nx_m \rfloor$. The main statement in the Theorem 1 now follows by the Lipschitz condition (A2), which implies that the $\tilde{y}_i$ have bounded derivatives and so $|\tilde{y}_i(x_m) - \tilde{y}_i(x'_m)| = o(1)$. The last statement is just (3.8). ∎

4 Proof of Theorem 2

The proof of Theorem 2 uses a deprioritised algorithm. For all but but an insignificant part of the time, the randomised mixture of operations used, determined by $\mathbf{p}$, is arbitrarily close to the mixture which the original prioritised algorithm uses on average at the corrsponding stage of its execution. There is also the necessity of building and then maintaining a large number of vertices of each possible degree, so that the required operations can always be carried out.

We will use the fact that for any choice of $\mathbf{p}$ satisfying a Lipschitz condition, the value of $\mathbf{Y} = (Y_1, \ldots, Y_j)$ a.a.s. follows close to the solution of the corresponding differential equation. To establish this, we use the following result which is a simplified version of [11, Theorem 6.1], which is an extension of [11, Theorem 5.1]. (See also [9, Theorem 1].) First we need a few definitions. The real variables $Y_1, \ldots, Y_a$ are defined on any discrete-time random process $G_0, G_1, \ldots$, which depends on n. We write $Y_i(t)$ for $Y_i(G_t)$, and for any domain $\hat{\mathcal{D}} \subseteq \mathbb{R}^{a+1}$ define the *stopping time* $T_{\hat{\mathcal{D}}}(Y_1, \ldots, Y_a)$ to be the minimum t such that $(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \notin \hat{\mathcal{D}}$. This is written as $T_{\hat{\mathcal{D}}}$ for short. In the following theorem, $\mathbf{P}$ and $\mathbf{E}$ denote probability and expectation for the random process. Note that even G_0 may be randomly distributed.

Theorem 3 *For $1 \le i \le a$, where a is fixed, let Y_i be a real-valued function of the components of a discrete time Markov process $\{G_t\}_{t\ge 0}$. Assume that $\hat{\mathcal{D}} \subseteq \mathbb{R}^{a+1}$ is closed and bounded and contains the set*

$$\{(0, y_1, \ldots, y_a) : \mathbf{P}(Y_i(0) = y_i n, 1 \le i \le a) \ne 0 \text{ for some } n\}$$

and

(i) for some constant β

$$\max_{1\le i\le a} |Y_i(t+1) - Y_i(t)| \le \beta$$

always for $t < T_{\hat{\mathcal{D}}}$,

(ii) for some functions $f_i : \mathbb{R}^{a+1} \to \mathbb{R}$ which are Lipschitz on some open set $\hat{\mathcal{D}}_0$ containing $\hat{\mathcal{D}}$ for all $i \le a$, and $\lambda = \lambda(n) = o(1)$,

$$|\, \mathbf{E}(Y_i(t+1) - Y_i(t) \mid G_0, \ldots, G_t) - f_i(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \,| \le \lambda$$

for $t < T_{\hat{\mathcal{D}}}$ and $1 \le i \le a$.

Then the following are true.

(a) For $(0, \hat{y}_1, \ldots, \hat{y}_a) \in \hat{\mathcal{D}}$ the system of differential equations

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = f_i(x, y_1, \ldots, y_a), \qquad i = 1, \ldots, a$$

has a unique solution in $\hat{\mathcal{D}}$ for $y_i : \mathbb{R} \to \mathbb{R}$ passing through

$$y_i(0) = \hat{y}_i,$$

$1 \le i \le a$, and which extends for positive x past some point, at which $x = \sigma$ say, at the boundary of $\hat{\mathcal{D}}$;

(b) Asymptotically almost surely

$$Y_i(t) = ny_i(t/n) + o(n) \tag{4.1}$$

uniformly for $0 \le t \le \min\{\sigma n, T_{\hat{\mathcal{D}}}\}$ *and for each* i*, where* $y_i(x)$ *and* σ *are as in (a) with* $\hat{y}_i = \frac{1}{n}Y_i(0)$. ∎

In part (b) of this theorem, "uniformly" refers to the convergence implicit in the $o()$ term. (We omit the hypothesis of [11, Theorem 5.1] that Y_i is bounded above by a constant times n, since this property follows anyway from the different hypothesis (i) we have here, together with the assumption that $\hat{\mathcal{D}}$ is bounded.)

We will also need the following property of solutions of first order differential equations.

Lemma 1 *Suppose that* $\mathbf{y}$ *satisfies the equations*

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = g_i(x, \mathbf{y})$$

for $(x, \mathbf{y}(x))$ *in a bounded open set* $\mathcal{D}$*, with initial conditions* $\mathbf{y}(0) = \hat{\mathbf{y}} = \hat{\mathbf{y}}(n)$*. Let* $\mathbf{z}$ *denote another solution, with initial conditions* $\mathbf{z}(0) = \hat{\mathbf{z}} = \hat{\mathbf{z}}(n)$*. Suppose that the functions* g_i *are Lipschitz on* $\mathcal{D}$ *and* $|\hat{\mathbf{y}}(n) - \hat{\mathbf{z}}(n)| \to 0$ *as* $n \to \infty$*. Let* $x_1 = \inf\{x : (x, \mathbf{y}(x)) \notin \mathcal{D}$ *or* $(x, \mathbf{z}(x)) \notin \mathcal{D}\}$*. Then* $|\mathbf{y}(x) - \mathbf{z}(x)| \to 0$ *uniformly for* $x \in [0, x_1)$*.*

Proof: This is standard, by the method of successive approximations (see [7, Theorem 2, Chapter 2] or [8, Section 3.22]). ∎

The differential equation (2.5) corresponds by (3.5) to the deprioritised **p**-algorithm with

$$p_i = \begin{cases} \frac{\tau_k}{\tau_k+\alpha_k} & i = d-k, \\ \frac{\alpha_k}{\tau_k+\alpha_k} & i = d-k-1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

The proof of Theorem 2 relies on the fact that $\tilde{\mathbf{y}}$ approximates the variables Y_i/n as the vector $(y_1, \ldots, y_a)$ in Theorem 3(b) a.a.s. The only great difficulty is that, due to the priority constraints, it turns out that (4.2) does not give an algorithm which is always compatible with the permissibility condition (3.6).

Proof of Theorem 2 The overall structure of the proof is to define a **p**-algorithm, depending on some arbitrarily small $\epsilon_1 > 0$, whose scaled variables Y_i/n agree with the functions $\tilde{y}_i$ to error $O(\epsilon_1)$. For the first $\epsilon_1 n$ steps, **p** is chosen to force Op_d to be used, which makes all variables strictly positive in the differential equations. From then onwards, the variables remain strictly positive. The advantage of having the variables positive is that every operation is then permissible in the algorithm. Then the theorem follows upon letting ϵ_1 go to 0. (One way of expressing the last step is to say that $\epsilon_1 \to 0$ sufficiently slowly, while $n \to \infty$ quickly.) We treat phase 1, where $k = 1$, in detail since it has some special features but also contains almost all of the ideas required for the general case. We assume for the present that $m > 1$. (The places in the argument which use this assumption will be signposted. The modifications required when $m = 1$ are covered in the discussion of the case $k = m$, which will be described at the end. They affect only the argument pertaining to the part of the process where $x = t/n \approx x_m$.)

The proof is broken into six parts, referring to the behaviour of various functions on given intervals.

Part 1: $\tilde{\mathbf{y}}$ on $[0, x_1]$

It is convenient to first make some observations about $\tilde{\mathbf{y}}$ and verify that $x_1 > 0 = x_0$. We will apply Theorem 3 (or, more conveniently, a standard result in the theory of ordinary differential equations which implies Theorem 3(a)) to the solution of the differential equations (2.5) determining $\tilde{\mathbf{y}}$. Note that from the assumptions in the theorem, the initial conditions are

$$\tilde{y}_d(0) = Y_d(0)/n > c_0 > 0, \quad \tilde{y}_i(0) = 0 \text{ for } i < d. \tag{4.3}$$

Define

$$\hat{\mathcal{D}} = \mathcal{D}_{\epsilon,M} \cap Q \tag{4.4}$$

where Q denotes the set of all points at which $\tau_1 + \alpha_1 \geq \epsilon$. Since $\mathcal{D}_{\epsilon,M}$ is a convex set, it is easy to extend or amend the definitions of the functions $f_{i,r}$ to a bounded open set $\hat{\mathcal{D}}_0$ containing $\mathcal{D}_{\epsilon,M}$, so that the Lipschitz property in condition (A2) applies on the entirety of $\hat{\mathcal{D}}_0$ and satisfying $\tau_1 + \alpha_1 > \epsilon/2$. For example, $f_{i,r}$ can be defined to take at $\mathbf{x}$ the value it has on the closest point to $\mathbf{x}$ in $\hat{\mathcal{D}}$. These properties ensure that Theorem 3(a) holds (as in [7, Chapter 2, Theorem 11]). Thus the solution $\tilde{\mathbf{y}}$ is defined uniquely for $x > 0$ past the boundary of $\hat{\mathcal{D}}$. To show that $x_1 > 0 = x_0$, it needs to be verified that this solution satisfies

$$\tau_1 > 0,\ \tau_1 + \alpha_1 > \epsilon,\ \tilde{y}_{d-1} > 0, |\tilde{y}_i| \leq M \text{ (all } i),\ \tilde{y}_i \geq 0 \text{ (all } i \leq d-1),\ \tilde{y}_d \geq \epsilon \tag{4.5}$$

for all x in some nonempty open interval $(0, c')$. This will show that the solution does not exit $\hat{\mathcal{D}}$ at $x = 0$, and hence exits at $x = x_1 > 0$, or at some larger value of x (if for instance $\tau_1 = 0$ at x_1).

By the Lipschitz property of the $f_{i,r}$ and the condition $\tau_1 + \alpha_1 > \epsilon/2$, for any initial condition $\mathbf{y}(0)$ there is an upper bound on the absolute value of the derivatives of the y_i, as given in (2.5), on $\hat{\mathcal{D}}_0$. This statement (and minor variations in which the derivatives of the y_i are similar functions involving the $f_{i,r}$) will be used several times, so we call it the *boundedness principle*. It follows that at any point $\mathbf{x} = (x, \tilde{\mathbf{y}}(x))$ in $\hat{\mathcal{D}}_0$, for $c' > 0$ sufficiently small, and for some C',

$$||\tilde{\mathbf{y}}(x') - \tilde{\mathbf{y}}(x)|| < C'|x' - x| \tag{4.6}$$

for $|x' - x| < c'$. Hence, for $c' > 0$ sufficiently small, for $x \in (0, c')$, $(x, \tilde{\mathbf{y}}(x))$ stays inside $\hat{\mathcal{D}}_0$ and, using (4.3), the upper bounds on $|\tilde{y}_i|$ and the lower bound on $\tilde{y}_d$ in (4.5) all hold. Similarly, we may assume the inequalities $\tau_1 > 0$ and $\tau_1 + \alpha_1 > \epsilon$ for x in the same interval, by (2.10) with $k = 1$. Additionally, note from (2.5) that

$$\frac{\mathrm{d}\tilde{y}_{d-1}}{\mathrm{d}x} = \frac{\tau_1}{\tau_1 + \alpha_1} f_{d-1,d-1}\left(x, \tilde{\mathbf{y}}\right) + \frac{\alpha_1}{\tau_1 + \alpha_1} f_{d-1,d-2}\left(x, \tilde{\mathbf{y}}\right).$$

By the initial conditions (4.3) and the upper and lower bounds in condition (B), $\alpha_1 = f_{d-2,d-1} = 0$ at $(0, \tilde{\mathbf{y}}(0))$. So

$$F(0, \tilde{\mathbf{y}}(0), d-1, 1) = f_{d-1,d-1}(0, \tilde{\mathbf{y}}(0)) > 0 \tag{4.7}$$

by the first inequality in (2.11). Thus the derivative (with respect to x) of $\tilde{y}_{d-1}$ is strictly positive at 0, and by the Lipschitz property of the functions in (2.6), it is therefore bounded below by a positive constant when $0 < x \le c'$ for $c' > 0$ sufficiently small. Hence

$$\tilde{y}_{d-1} > 0 \text{ for } x \in (0, c']. \tag{4.8}$$

The only part of (4.5) remaining to be shown is $\tilde{y}_i \ge 0$ (for $1 \le i \le d-2$). The differential equation (2.5) with $i = d-2$ shows that $\mathrm{d}\tilde{y}_{d-2}/\mathrm{d}x = 0$, and hence $\tilde{y}_{d-2} = 0$ for $x \in (0, c')$ (since $\tilde{y}_{d-2}(0) = 0$ by (4.3)). For $1 \le i < d-2$, the initial values are all 0, and condition (B) ensures, by downward induction on i beginning with the case $i = d-2$ already established, that $f_{i,r} = 0$ for $r = d-1$ and $d-2$, so that $\mathrm{d}\tilde{y}_i/\mathrm{d}x = 0$. We now conclude that $x_1 > x_0 = 0$ and that

$$\tilde{y}_i(x) \equiv 0 \quad (0 \le x \le x_1,\ 1 \le i \le d-2), \tag{4.9}$$

which is the case $k = 1$ of (3.8). For future reference, also note the following. Since (by assumption) $m > 1$, the definition of x_1 in (2.8) implies that

$$\tau_1(x_1, \tilde{\mathbf{y}}(x_1)) = 0. \tag{4.10}$$

Part 2: $\mathbf{p}$ and $\tilde{\mathbf{y}}^{(1)}$ on $[0, \epsilon_1]$

To define the initial part of the **p**-algorithm, set $\epsilon_1 > 0$ with $\epsilon_1 < x_1$, and satisfying upper bounds specified by the condition (4.11) below and others imposed later. Put $p_d = 1$ and $p_i = 0$ $(i \ne d)$. With $\mathbf{p} = (p_1, \ldots, p_d)$ for $0 \le t \le t_1 = \lfloor \epsilon_1 n \rfloor$, the first $\epsilon_1 n$ operations of the **p**-algorithm are Op_d. (Technically, if Y_d dropped to 0 during these operations, this algorithm would have to be terminated, but it will be ensured that this cannot happen.) We call this part of phase 1 the *preprocessing* subphase.

Choose ϵ and ϵ_1 so small that

$$C(\epsilon + \epsilon_1) < c_0 \tag{4.11}$$

where C is as in (A1) and c_0 is from (3.7). Apply Theorem 3 to the variables $Y_1, \ldots, Y_j$, as determined by the **p**-algorithm, with

$$\hat{\mathcal{D}} = \mathcal{D}_{\epsilon,M} \cap \{(x, \mathbf{y}) : x \le \epsilon_1\},$$

and $\mathcal{D}_{\epsilon,M}$ as in (3.3). Note that part (i) of Theorem 3 holds by (A1), and (ii) holds by (2.2) and (A2). We also use here the fact that when $(t/n, \mathbf{Y}(t)) \in \hat{\mathcal{D}}$ we have $Y_d > 0$, so that Op_d is always permissible, (3.6) holds, and the algorithm does not become stuck. The conclusion from Theorem 3(b) is that a.a.s. $Y_i(t) = n y_i(t/n) + o(n)$ for each $1 \le i \le j$, where the y_i satisfy the system

$$\frac{dy_i}{dx} = f_{i,d}(x, \mathbf{y}), \quad y_i(0) = Y_i(0)/n \qquad (i = 1, \ldots, j) \tag{4.12}$$

for all t until either $(t/n, \mathbf{Y}(t)/n) \notin \hat{\mathcal{D}}$ or $(t/n, \mathbf{y}(t/n)) \notin \hat{\mathcal{D}}$ (at say $t = t_1$). Here of course $\mathbf{y}$ denotes $(y_1, \ldots, y_j)$. Note that the only boundary constraints of $\hat{\mathcal{D}}$ active at the initial conditions are $x = 0$ and possibly $y_i = 0$ $(1 \le i \le d)$. Furthermore, neither vector

can leave $\hat{\mathcal{D}}$ at this point. This is because, firstly, the appropriate Y_i are nonnegative by (3.2), and, secondly, since $y_d(0) > c_0 > 0$, condition (B) with $r = d$ ensures that for $i < d$, either y_i is at least some positive constant, or the derivative of y_i is non-negative. Thus, by downward induction on i, in place of (4.9) we have

$$y_i > 0 \text{ on } (0, \epsilon_1], \quad (i = d-1, d-2, \ldots, 1). \tag{4.13}$$

By (A1) and (4.11), and (3.2) once more, it is true deterministically that the constraint on $\hat{\mathcal{D}}$ which the vector $(t/n, \mathbf{Y}(t)/n)$ first violates is $x = \epsilon_1$, and that it cannot come arbitrarily close to the other boundaries of $\mathcal{D}_{\epsilon,M}$ given by upper bounds on the Y_i (and on x since $x_m < M$), or the lower bound on Y_d. In view of the a.a.s. approximation of this vector by $(x, \mathbf{y})$, the solution of the differential equation must also exit the domain $\hat{\mathcal{D}}$ at this boundary. Let us denote the solution of this equation by $\tilde{\mathbf{y}}^{(1)}$ $(0 \le x \le \epsilon_1)$, with components $\tilde{y}_i^{(1)}$.

Then, in particular, from the above argument there exists $c = c(\epsilon_1) > 0$ for which

$$\tilde{y}_i^{(1)}(\epsilon_1) > c \quad (i = 1, \ldots, d). \tag{4.14}$$

Part 3: $\tilde{\mathbf{y}}^{(1)}$ on $(\epsilon_1, x_1^{(1)}]$

Recall that by choice, $\epsilon_1 < x_1$. To continue the definition of the deprioritised algorithm, first define $\tilde{\mathbf{y}}^{(1)}(x)$ for $x > \epsilon_1$ to satisfy the basic differential equation (2.5) with $k = 1$ and with α_1 and τ_1 defined as the functions in (2.3). This is analogous with the definition of $\tilde{\mathbf{y}}$ in (2.8) for $x \le x_1$, but with initial conditions given by the value $\tilde{\mathbf{y}}^{(1)}(\epsilon_1)$, thus determining another version of x_1, which we denote by $x_1^{(1)}$. Note that the definition of $x_1^{(1)}$ depends on ϵ_1. We shall also be imposing further upper bounds on ϵ_1. Since what has been discussed holds for all ϵ_1 sufficiently small, this is permissible.

To make the situation clear, the next immediate aim is to show that $x_1^{(1)} > \epsilon_1$. In view of the boundedness principle described above, by taking ϵ_1 sufficiently small, we may assume that $||\tilde{\mathbf{y}}^{(1)}(\epsilon_1) - \tilde{\mathbf{y}}^{(1)}(0)|| = ||\tilde{\mathbf{y}}^{(1)}(\epsilon_1) - \tilde{\mathbf{y}}(0)||$ is arbitrarily small. It follows that we may assume that the inequalities in (2.10) involving τ_1 and α_1 holding at $(0, \tilde{\mathbf{y}}(0))$ also hold at $(\epsilon_1, \tilde{\mathbf{y}}^{(1)}(\epsilon_1))$. Thus $(\epsilon_1, \tilde{\mathbf{y}}^{(1)}(\epsilon_1))$ is interior to $\hat{\mathcal{D}}$ as defined in (4.4), so as before the required solution of the differential equation exists, and moreover $x_1^{(1)} > \epsilon_1$.

It also clarifies issues to establish that $\tilde{\mathbf{y}}^{(1)}$ and $\tilde{\mathbf{y}}$ are (for small enough ϵ_1) arbitrarily close to each other, as are $x_1^{(1)}$ and x_1. By taking ϵ_1 arbitrarily small we may assume, by the boundedness principle, that $||\tilde{\mathbf{y}}^{(1)}(\epsilon_1) - \tilde{\mathbf{y}}(\epsilon_1)||$ is arbitrarily small. Hence, since $\tilde{\mathbf{y}}^{(1)}$ and $\tilde{\mathbf{y}}$ satisfy the same differential equation, by Lemma 1 there is a fixed function g with $\lim_{x \to 0} g(x) = 0$, such that

$$||\tilde{\mathbf{y}}^{(1)}(x) - \tilde{\mathbf{y}}(x)|| < g(\epsilon_1) \quad \text{for all } x \in [\epsilon_1, x_1^{(1)}]. \tag{4.15}$$

Next consider $|x_1^{(1)} - x_1|$. Note that $\tilde{\mathbf{y}}^{(1)}$ satisfies, at $x_1^{(1)}$, some condition which either determines a boundary of $\mathcal{D}_{\epsilon,M}$ or is one of the other conditions given in (2.8) defining x_1 there. We will show that it is indeed the condition $\tau_1 \le 0$. Consider first the condition $\tilde{y}_{d-1}^{(1)} \le 0$ (recalling $k = 1$ here). For convenience, we first revisit x in the interval $[0, \epsilon_1]$, where the derivative of $\tilde{y}_{d-1}^{(1)}$ is $f_{d-1,d}$. The latter function is positive at

0 by condition B. As in the argument leading to (4.8), we observe that it must remain positive for $x \in [0, c']$ ($c' > 0$ sufficiently small, independent of ϵ_1), and conclude (since $\tilde{y}_{d-1}^{(1)}(0) = 0$) that

$$\tilde{y}_{d-1}^{(1)}(x) > c_0 x \text{ for } 0 < x \leq \epsilon_1,$$

where c_0 is some positive constant, for ϵ_1 sufficiently small. This inequality extends to the interval $0 < x \leq c'$ by repeating the argument again, beginning with noting that the appropriate derivative here is F, given by (2.5), and is positive at 0 again by (4.7). Thus we may assume

$$\tilde{y}_{d-1}^{(1)} > c_0 x \text{ for } x \in (0, c']. \tag{4.16}$$

Here c' can be taken to be the same as c' in (4.8), by setting each equal to the minimum of the two.

To treat $\tilde{y}_{d-1}^{(1)}$ for $x > c'$, first consider $\tilde{y}_{d-1}$. Note that by the definition of x_1, $\tilde{y}_{d-1} > 0$ for $x \in (0, x_1)$, and by the termination condition $\tilde{y}_{d-k} \leq 0$ in (2.8) and the assumption $m > 1$,

$$\tilde{y}_{d-1}(x_1) > 0. \tag{4.17}$$

(If $k = m$ this is of course not valid; in the treatment of that case in Part 6, a modification to this argument is given which, roughly speaking, will redefine $x_m^{(1)}$ to be a little smaller than x_m.) Hence, by continuity of $\tilde{y}_{d-1}$, it is bounded below by some positive constant on $[c', x_1]$. Thus, by (4.15), for ϵ_1 sufficiently small,

$$\tilde{y}_{d-1}^{(1)} > c_0' \text{ for } x \in [c', x_1] \quad (c_o', c' > 0) \tag{4.18}$$

where c_0' and c' are independent of ϵ_1. From this and (4.16), and the boundedness principle, the boundary $\tilde{y}_{d-1}^{(1)} \leq 0$ cannot be reached for $x \leq \min\{x_1^{(1)}, x_1 + c''\}$ for some $c'' > 0$ (and for ϵ_1 sufficiently small). Here c'' does not depend on ϵ_1.

The other boundaries and conditions, except for the condition $\tau_1 \leq 0$, can be dealt with in a similar fashion, to show that they cannot come into effect for $x \leq \min\{x_1^{(1)}, x_1 + c''\}$ (redefining c'', of course). These arguments for the most part are easier, since for example τ_1 begins life positive, so there is no need to consider its derivative in order to show that $\tau_1 > 0$ near $x = 0$. The only case different enough to require special attention is the boundary of $\mathcal{D}_{\epsilon,M}$ given by $\tilde{y}_i^{(1)} = 0$ for $i < d - 1$. The argument leading to (4.9) can be combined with the initial conditions at $x = \epsilon_1$ in (4.14). This shows firstly that the derivative of $\tilde{y}_{d-2}$ is 0, so that $\tilde{y}_{d-2}(x) > 0$ on $[\epsilon_1, x_1^{(1)}]$, and secondly, using inequalities rather than equalities as for (4.13), that

$$\tilde{y}_i^{(1)} > 0 \quad (1 \leq i \leq d - 2) \tag{4.19}$$

on $[\epsilon_1, x_1^{(1)}]$. We turn to examining τ_1. By (4.10) and (4.15),

$$\tau_1(x_1, \tilde{\mathbf{y}}^{(1)}(x_1)) \to 0 \text{ as } \epsilon_1 \to 0. \tag{4.20}$$

However, by the third inequality in (2.11) with $k = 2$, the derivative of τ_1 is negative at $(x_1, \tilde{\mathbf{y}}(x_1))$. By (3.4), this extends to give a negative upper bound on the derivative of τ_1 in a neighbourhood of $(x_1, \tilde{\mathbf{y}}(x_1))$, implying by (4.20) that $\tau_1(x_1, \tilde{\mathbf{y}}^{(1)}(x_1))$ is forced to pass through 0 on such a neighbourhood for sufficiently small ϵ_1. This yields $x_1^{(1)} <$

$x_1 + c''$, so the other boundary conditions cannot come into play. It thus determines the location of $x_1^{(1)}$ and yields

$$|x_1^{(1)} - x_1| \to 0 \text{ as } \epsilon_1 \to 0. \tag{4.21}$$

Part 4: p-algorithm on $(\epsilon_1, x_1^{(1)}]$

For $x \in [\epsilon_1, x_1^{(1)}]$, define $\mathbf{p} = (p_1, \ldots, p_d)$ as given in (4.2) with $k = 1$ and with α_1 and τ_1 defined in (2.3) with $\mathbf{y} = \tilde{\mathbf{y}}^{(1)}$. For the deprioritised **p**-algorithm to be feasible, it is necessary that $\tau_1(x, \tilde{\mathbf{y}}^{(1)}(x))$ and $\alpha_1(x, \tilde{\mathbf{y}}^{(1)}(x))$ are nonnegative for $x \in [\epsilon_1, x_1^{(1)}]$. For τ_1, this is guaranteed by definition of $x_1^{(1)}$, but for α_1 it is not immediate and requires a more careful examination of $\tilde{y}_{d-2}^{(1)}$ (and also requires further restrictions on the size of ϵ_1). We basically argue that $\tilde{y}_{d-1}^{(1)}$ is at least of the order of ϵ_1, whereas $\tilde{y}_{d-2}^{(1)} = O(\epsilon_1^2)$. The lower bound in condition (B) with $i = d-2$, $r = d-1$ then shows that $\alpha_1 > 0$.

From the initial condition $\tilde{y}_{d-1}^{(1)}(0) = 0$ and the boundedness principle, we have $\tilde{y}_{d-1}^{(1)}(x) = O(\epsilon_1)$ for $x \in [0, \epsilon_1]$. So by the upper bound in condition (B), $f_{d-2,d} = O(\epsilon_1)$ on the same interval, and hence by (4.12) and the fact that $\tilde{y}_{d-2}^{(1)}(0) = 0$, it follows that $|\tilde{y}_{d-2}^{(1)}(\epsilon_1)| = O(\epsilon_1^2)$. Now the derivative of $\tilde{y}_{d-2}^{(1)}$ on $[\epsilon_1, x_1^{(1)}]$ given by (2.5) with $k = 1$ is identically 0, so

$$|\tilde{y}_{d-2}^{(1)}| = O(\epsilon_1^2) \text{ for } x \in [\epsilon_1, x_1^{(1)}]. \tag{4.22}$$

From this, (4.16) and the lower bound in condition (B) with $r = d-1$, we obtain $\alpha_1 > 0$ for $0 < x < c'$. For $c' \le x \le x_1^{(1)}$ we obtain $\alpha_1 > 0$ similarly using (4.18), (4.22) and condition (B), noting that the (possible) interval $(x_1, x_1^{(1)})$ does not cause a problem, by (4.21) and the boundedness principle.

We have now shown that the values of p_i in (4.2) are indeed probabilities for $x \in [\epsilon_1, x_1^{(1)}]$ with $\mathbf{p} = (p_1, \ldots, p_d)$ defined as above, where $\tau_1 = \tau_1(x, \tilde{\mathbf{y}}^{(1)}(x))$ and $\alpha_1 = \alpha_1(x, \tilde{\mathbf{y}}^{(1)}(x))$. Apply Theorem 3 to the deprioritised **p**-algorithm with

$$\hat{\mathcal{D}} = \mathcal{D}_{\epsilon,M} \cap Q \cap \{(x, \mathbf{y}) : \epsilon_1 \le x \le x_1^{(1)},\ y_i > 0 \ (i = d-2 \text{ and } d-1)\}$$

where Q is defined as for (4.4), trivially translating the theorem statement so that the initial point is $t = t_1 \approx \epsilon_1 n$ rather than $t = 0$. The hypotheses of Theorem 3 are satisfied as in the application in Part 2 for the **p**-algorithm with $t \le t_1$, the only difference being the check that the algorithm cannot become stuck for $t < T_{\hat{\mathcal{D}}}$. In this case, the condition in $\hat{\mathcal{D}}$ on positivity of y_{d-2} and y_{d-1} ensures that (3.6) holds for such t. We conclude that a.a.s.

$$Y_i(t) = n\tilde{y}_i^{(1)}(t/n) + o(n) \tag{4.23}$$

for each $1 \le i \le j$, uniformly for all $t \ge t_1$ until either $(t/n, \mathbf{Y}(t)/n) \notin \hat{\mathcal{D}}$ or $(t/n, \tilde{\mathbf{y}}^{(1)}(t/n)) \notin \hat{\mathcal{D}}$.

In view of (4.18), the observation following it, and (4.19), $(x, \tilde{\mathbf{y}}^{(1)}(x))$ cannot approach arbitrarily close to the boundaries $\tilde{y}_i^{(1)} = 0$ $(1 \le i \le d-1)$ of $\hat{\mathcal{D}}$ for $x \in [\epsilon_1, x_1^{(1)}]$ (for fixed ϵ_1). Since $m > 1$, by continuity the solution cannot approach arbitrarily close to any other boundaries of $\hat{\mathcal{D}}$ either, apart from $\tau_1 = 0$. Thus, in view of (4.23), a.a.s. $(t/n, \mathbf{Y}(t)/n)$ exits $\hat{\mathcal{D}}$ when $t = \lfloor x_1^{(1)} \rfloor + 1$. Incorporating the analysis in Part 2 for $t < t_1$, we have (4.23) a.a.s. for $0 \le t \le \lfloor x_1^{(1)} \rfloor$.

Part 5: Phase k, $2 \le k \le m-1$

Now consider arbitrary $k > 1$ (but $k < m$, so in particular $k < d-1$). We give some details to show that the above argument for t/n in the interval $[0, x_1^{(1)}]$, which should be regarded as $[x_0^{(1)}, x_1^{(1)}]$ may be repeated inductively for the interval $[x_{k-1}^{(1)}, x_k^{(1)}]$, where

$$|x_k^{(1)} - x_k| \to 0 \text{ as } \epsilon_k \to 0. \tag{4.24}$$

The conclusions derived for the first phase become, in their general form, inductive hypotheses for the general argument. The argument in each interval includes a preprocessing subphase in which all operations are Op_d. This introduces a positive quantity ϵ_k which will be assumed to be sufficiently small for our purposes, and in particular imposes a new upper bound on ϵ_{k-1} (and hence on ϵ_i for $i < k-1$) depending on ϵ_k.

The argument gives

$$|\tilde{\mathbf{y}}^{(1)}(x) - \tilde{\mathbf{y}}(x)| < g(\epsilon_k) \quad \text{for all } x \in [x_{k-1}^{(1)}, x_k^{(1)}] \tag{4.25}$$

for $1 \le k < m$ where $\lim_{x \to 0} g(x) = 0$. It also gives

$$Y_i(t) = n\tilde{y}_i^{(1)}(t/n) + o(n) \text{ for each } 1 \le i \le j \text{ uniformly on } [x_{k-1}^{(1)}, x_k^{(1)}] \text{ a.a.s.} \tag{4.26}$$

There are only a few real differences in the argument, encountered in the discussion below. Firstly, in phase $k \ge 2$ the random variables $Y_1, \ldots, Y_{d+1-k}$ play the role of $Y_1, \ldots, Y_d$ in phase 1, and $Y_{d+2-k}, \ldots, Y_d$ can be treated almost like $Y_{d+1}, \ldots, Y_j$. In particular, the inductive analogue of (4.17) is

$$\tilde{y}_i(x_k) > 0 \quad (i \ge d-k). \tag{4.27}$$

Secondly, the derivative of $\tilde{y}_{d-k}$ at the start of a phase is 0, not positive as it was for $k = 1$, so we argue with the second derivative. Thirdly, $\tilde{y}_i^{(1)}$ is positive, not 0, at $x_{k-1}^{(1)}$ for $i \le d-k$ and $k \ge 2$. (This affects the argument in a couple of places.) Fourthly, as shown above, $f_{d-k,d-k} = 0$ at $(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))$ for $k \ge 2$ by the fact that $\tau_{k-1} = 0$ here. Lastly, there is a new effect occurring, that the upper bound on ϵ_{k-1} involves a positive function of ϵ_k (so, working backwards, the argument in later phases affects the upper bound on ϵ_1).

The proof in Part 1 that $x_1 > x_0$ is easily adapted to show that $x_k > x_{k-1}$ for general $k \ge 2$. For this, we assume as part of the inductive hypotheses the analogues of (4.9), (4.10) and (4.17), and thus,

$$\tilde{y}_{d-k+1}(x_{k-1}) > 0, \quad \tau_{k-1}(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1})) = 0, \quad \tilde{y}_i(x_{k-1}) = 0 \text{ for } i \le d-k. \tag{4.28}$$

The main modification required for the proof in Part 1 is the derivation of the generalisation of (4.8), that $\tilde{y}_{d-k} > 0$ for $x \in (x_{k-1}, x_{k-1} + c']$. In the case of $k \ge 2$, the right-hand derivative of $\tilde{y}_{d-k}$ at x_{k-1} is not positive. For, by (2.5), this derivative is

$$\frac{\tau_k}{\tau_k + \alpha_k} f_{d-k,d-k}\left(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1})\right) + \frac{\alpha_k}{\tau_k + \alpha_k} f_{d-k,d-k-1}\left(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1})\right).$$

This is equal to 0 since by the middle equation in (4.28), $f_{d-k,d-k}\left(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1})\right) = 0$, and by condition (B) and the right equation in (4.28), $\alpha_k = 0$ at $(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))$.

Differentiating again (and noting again the quantities above which are zero) gives the second derivative of $\tilde{y}_{d-k}$ at x_{k-1}, with respect to x, to be $f'_{d-k,d-k} + f'_{d-k-1,d-k}\frac{f_{d-k,d-k-1}}{\tau_k}$ at $(x_{k-1}, \tilde{\mathbf{y}}(x_{k-1}))$ (where prime denotes differentiation with respect to x). Note that $\tau_k > 0$ by (2.10), so this second derivative is strictly positive by the second inequality in (2.11). It follows that

$$c_1\theta^2 < \tilde{y}_{d-k}(x_{k-1}+\theta) < c_2\theta^2 \tag{4.29}$$

for θ sufficiently small but positive. Thus we may assume that $\tilde{y}_{d-k} > 0$ for $x \in (x_{k-1}, x_{k-1}+c']$ (for c' sufficiently small), as required. Note that the argument for (4.9) now shows that (3.8) holds in general (with the only restriction $k < m$ at present).

The other modification for Part 1 is, as required for (4.27), that $\tilde{y}_i$, $d-k+1 \le i \le d-1$, which begins phase k at a positive value by (4.27), remains positive. This follows easily in the same way as (4.19).

We conclude inductively that $x_k > x_{k-1}$ and hence that the first and third equations in (4.28) hold with k replaced by $k+1$. The second follows similarly, arguing as for the justification for (4.17).

We next repeat the argument in Part 2, on an interval $[x^{(1)}_{k-1}, x^{(1)}_{k-1}+\epsilon_k]$, and Parts 3 and 4, on $(x^{(1)}_{k-1}+\epsilon_k, x^{(1)}_k]$. Let us examine Part 2. For $[x^{(1)}_{k-1}, x^{(1)}_{k-1}+\epsilon_k]$, put $p_{d-k+1}=1$ and all other p_i equal to 0, put $\mathbf{p} = (p_1, \dots, p_d)$, and let $\tilde{\mathbf{y}}^{(1)}$ be the solution of the equations

$$\frac{dy_i}{dx} = f_{i,d-k+1}(x, \mathbf{y})$$

with initial value $\tilde{\mathbf{y}}^{(1)}(x^{(1)}_1)$ at $x^{(1)}_1$. The argument goes through as for $k=1$, and in particular we obtain the generalisation of (4.13), that $\tilde{y}^{(1)}_i > 0$ on $[x^{(1)}_{k-1}, x^{(1)}_{k-1}+\epsilon_k]$ for $i \le d-k$. (The fact that $\tilde{y}^{(1)}_i > 0$ at $x^{(1)}_{k-1}$ follows from the generalisation of (4.19), which is part of the inductive hypothesis.) A similar argument shows that the same conclusion holds for $d-k+2 \le i \le d-1$. The case of $i=d$ is exactly as before. Finally, for $i = d-k+1$ we rely on the inductive analogue of (4.18). This says that $\tilde{y}^{(1)}_{d-k} > c'_0$ for $x \in [c', x_k]$ and implies that $\tilde{y}^{(1)}_{d-k+1} > c'_0$ at x_{k-1} which translates by the usual argument to $x^{(1)}_{k-1}$.

For Part 3, define $\tilde{\mathbf{y}}^{(1)}(x)$ for $x > x^{(1)}_{k-1}+\epsilon_k$ to satisfy (2.5) with α_k and τ_k defined as in (2.3), and $\mathbf{p} = (p_1, \dots, p_d)$ as given in (4.2). As with $k=1$, define $x^{(1)}_k$ as the analogue of x_k for $\tilde{\mathbf{y}}^{(1)}$. For this and Part 4, the structure of the argument should be clear, so we mainly point out how it is modified to take account of the differences listed above, as they are encountered.

Early in Part 3, we argue that $|\tilde{\mathbf{y}}^{(1)}(\epsilon_1) - \tilde{\mathbf{y}}(0)| \to 0$ as $\epsilon_1 \to 0$. The required generalisation is

$$|\tilde{\mathbf{y}}^{(1)}(x^{(1)}_{k-1}+\epsilon_k) - \tilde{\mathbf{y}}(x_{k-1})| \to 0 \text{ as } \epsilon_k \to 0. \tag{4.30}$$

This follows using (by the inductive version of (4.15)) $||\tilde{\mathbf{y}}^{(1)}(x_{k-1}) - \tilde{\mathbf{y}}(x_{k-1})|| \to 0$ as $\epsilon_{k-1} \to 0$, the inductive version of (4.21), and the boundedness principle.

The analogue of (4.16) is not valid, so requires modification. The derivative of $\tilde{y}^{(1)}_{d-k}$ on $[x^{(1)}_{k-1}, x^{(1)}_{k-1}+\epsilon_k]$ is bounded below by a positive constant (using condition B) and so $\tilde{y}^{(1)}_{d-k}$ is at least $c_0\epsilon_k$ at $x^{(1)}_{k-1}+\epsilon_k$, for some $c_0 > 0$. On the other hand, the argument leading to (4.29) shows that the first derivative of $\tilde{y}_{d-k}$ is 0 at x_{k-1}, and the second

derivative is positive. By taking ϵ_k sufficiently small, and using (3.4), (4.24) and (4.30) and the boundedness principle, the second derivative of $\tilde{y}^{(1)}_{d-k}$ is shown to be greater than some positive constant (independent of ϵ_k) on the interval $(x^{(1)}_{k-1}+\epsilon_k, x^{(1)}_{k-1}+c')$ for some $c'>0$, whilst the first derivative is $O(\epsilon_k)$ at $x^{(1)}_{k-1}+\epsilon_k$. Here we assume (as at other places) that ϵ_{k-1} is sufficiently small, in particular smaller than ϵ_k. Together with the statement above, this implies that

$$\tilde{y}^{(1)}_{d-k} > c_0\epsilon_k \text{ on } [x^{(1)}_{k-1}, x^{(1)}_{k-1}+c'] \tag{4.31}$$

for some $c_0>0$ for ϵ_k sufficiently small. The rest of the argument in Part 3 goes through as before, and gives in particular the inductive version of (4.21).

For Part 4, let $x \in [x^{(1)}_{k-1}+\epsilon_k, x^{(1)}_k]$ and define $\mathbf{p}=(p_1,\ldots,p_d)$ as given in (4.2). The verification that $\alpha_1>0$ is a little different from the case $k=1$. From (4.25), by taking ϵ_{k-1} sufficiently smaller than ϵ_k we can ensure that $\tilde{y}^{(1)}_{d-k-1}(x^{(1)}_{k-1}) = o(\epsilon_k)$. Then the argument as before using condition B and $\tilde{y}^{(1)}_{d-k}=O(\epsilon_k)$ on $[x^{(1)}_{k-1}, x^{(1)}_{k-1}+\epsilon_k]$ shows that $\tilde{y}^{(1)}_{d-k-1}=o(\epsilon_k)$ on this interval. Since its derivative is zero on $[x^{(1)}_{k-1}+\epsilon_k, x^{(1)}_k]$, we have $\alpha_k>0$ in view of (4.31) and condition B.

Part 6: Phase m

Finally, we turn attention to the case $k=m$. There are two points which differ for this: firstly, if $k=d-1$ then the definition of the derivative in (2.5) and (2.6) is different and the assumption in (2.11) is correspondingly different, and secondly, other conditions may now occur at the point $x_k=x_m$. We treat the second difference first. What may now occur at x_m, and has possible relevance to the argument, is any of the following: $\tau_k+\alpha_k=\epsilon$, $\tilde{y}_{d-k}=0$, $x=M$, $|y_i|=M$, $y_i<0$ for $1\le i<d$, or $y_d=\epsilon$. To avoid this, we make a special definition of $x^{(1)}_m$ as $x_m-\epsilon_2$ where $\epsilon_2>0$. Taking ϵ_2 sufficiently small, we have by continuity exactly the same situation as for $k<m$. Thus (4.25) holds also for $k=m$.

There is no change in the argument to show that (3.8) holds for $k=m<d-1$. On the other hand, it asserts nothing in the case $k=m=d-1$, so (3.8) is fully established.

With regard to $k=d-1$, the argument for this final phase of course requires altering the definition of $\mathbf{p}$ so that $p_1=1$ and $p_i=0$ for $i>1$. The rest of the argument goes through, though it only requires a much simpler version than for $k<d-1$, since in this case there are no requirements on α or τ. This is why the case $k=d-1$ is excluded from these conditions in (2.8) and in (2.10) and parts of (2.11). The altered assumption in (2.11) is all that is required to ensure that $\tilde{y}^{(1)}_i$ is positive for the initial segment of the last phase.

The main conclusion of this inductive argument is that (4.24), (4.25) and(4.26) hold for each k. Combining these, we may choose ϵ_m to be a function of n which tends to 0 sufficiently slowly, with each ϵ_k correspondingly smaller as requried by the inductive argument, and obtain a.a.s. $Y_i(t)=n\tilde{y}_i+o(n)$ uniformly for $0\le t\le x^{(1)}_m n$ and $1\le i\le j$, where $x_m-x^{(1)}_m=o(1)$. ∎

5 Extensions

Duckworth et al. [5] and Zito [12] studied the performance of greedy algorithms as heuristics for maximum induced matchings in random cubic graphs, and Duckworth [3] studied similar heuristics for star packing, maximum 2-independent sets, and minimum connected dominating sets in random cubic graphs. Hopefully, these studies can be extended to random d-regular graphs using the results in the present paper. At least, the methods may permit a simplified approach. For instance, extension of Example 2 in Section 2 to the d-regular version of the minimum (independent) dominating set problem is presently under way using Theorem 1.

We may also elaborate now on the comment in the Introduction that the approach of this paper may be useful even when the specific results proved here do not apply. There is a good chance that a particular deprioritised algorithm can be analysed more easily than the related prioritised algorithm, and it is clear that in many situations they can give the same result to any desired accuracy. One approach would be to use the same theoretical approach as in this paper, and show that one can approximate the original algorithm arbitrarily closely by a deprioritised version.

A more computational approach is potentially simpler and so may also be worth considering. Suppose that (2.2) holds, as well as conditions (A1) and (A2), but perhaps some of the other hypotheses of Theorem 2 do not. Then $\tilde{\mathbf{y}}$ may be computed numerically, and it may be suspected that it gives an accurate representation of the asymptotic behaviour of the algorithm. One way to establish this would be to compute $\tilde{\mathbf{y}}^{(1)}$ numerically, as defined in the proof of Theorem 2, for given values of ϵ_1, ϵ_2 and so on, such that the solution gives $(x_1^{(1)}, \tilde{\mathbf{y}}^{(1)}(x_1^{(1)}))$ sufficiently close to $(x_1, \tilde{\mathbf{y}}(x_1))$. Provided the trajectory of $\tilde{\mathbf{y}}^{(1)}$ does not touch the boundaries of $\mathcal{D}_{\epsilon,M}$ (apart from at x_0), the desired argument is likely to proceed successfully, with each phase only requiring a simple application of Theorem 3. The result should be an almost sure bound on the size of a set constructed by the algorithm, which is as close as desired to that of the result of the prioritised algorithm (in the sense of having difference less than $\epsilon' n$ for any desired $\epsilon' > 0$, by performing the numerical computation with ϵ_1 etc. sufficiently small). The initialisation subphases may not even be required after phase 1.

Acknowledgement I would like to thank the referees for numerous suggestions in improving the presentation of the first version of this paper.

References

[1] D. Achlioptas, Setting 2 variables at a time yields a new lower bound for random 3-SAT, In *32nd Annual ACM Symposium on Theory of Computing (STOC 32), Portland, Oregon* pp. 28–37 (2000).

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

[3] W. Duckworth, *Greedy algorithms and cubic graphs*, Doctoral thesis, University of Melbourne, 2001.

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

[5] W. Duckworth, N.C. Wormald and M. Zito, Maximum induced matchings of random cubic graphs, *Journal of Computational and Applied Mathematics* **142** (2002), 39–50.

[6] A.M. Frieze and S. Suen, On the independence number of random cubic graphs, *Random Structures and Algorithms* **5** (1994), 649–664.

[7] W. Hurewicz, *Lectures on Ordinary Differential Equations*, M.I.T. Press, Cambridge Massachusetts (1958).

[8] E.L. Ince, *Ordinary Differential Equations*, Longmans, London (1927).

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

[10] N.C. Wormald, Models of random regular graphs. In *Surveys in combinatorics, 1999 (Canterbury)*, J.D. Lamb and D.A. Preece (eds), pp. 239–298. Cambridge University Press, Cambridge, 1999.

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

[12] M. Zito, Induced Matchings in Regular Graphs and Trees. In *Proceedings of the 25th International Workshop on Graph Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science vol. 1665, 89–100. Springer-Verlag, 1999.

[13] M. Zito, Greedy algorithms for minimisation problems in random regular graphs, in *Algorithms - ESA 2001 (F. Meyer auf der Heide, Ed.), LNCS 2161*, pp. 524–536, Springer-Verlag, 2001.