The analysis of a prioritised probabilistic algorithm to find large induced forests in regular graphs with large girth

Carlos Hoppen

The analysis of a prioritised probabilistic algorithm to find large induced forests in regular graphs with large girth

Carlos Hoppen[†]

Instituto de Matemática, Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves, 9500, Porto Alegre-RS, Brazil, 91509-900

The analysis of probabilistic algorithms has proved to be very successful for finding asymptotic bounds on parameters of random regular graphs. In this paper, we show that similar ideas may be used to obtain deterministic bounds for one such parameter in the case of regular graphs with large girth. More precisely, we address the problem of finding a large induced forest in a graph G, by which we mean an acyclic induced subgraph of G with a lot of vertices. For a fixed integer $r \geq 3$, we obtain new lower bounds on the size of a maximum induced forest in graphs with maximum degree r and large girth. These bounds are derived from the solution of a system of differential equations that arises naturally in the analysis of an iterative probabilistic procedure to generate an induced forest in a graph. Numerical approximations suggest that these bounds improve substantially the best previous bounds. Moreover, they improve previous asymptotic lower bounds on the size of a maximum induced forest in a random regular graph.

Keywords: Probabilistic algorithms, graphs, large girth

1 Introduction and main results

An *induced forest* in a graph G is an acyclic induced subgraph of G. The problem of finding a large induced forest in a graph G has been of great interest in graph theory, especially in its form known as the *decycling set problem*, also called the *feedback vertex set problem*. A decycling set of a graph is a subset of its vertices whose deletion yields an acyclic graph. Hence, induced forests and decycling sets are dual concepts, as a set $S \subseteq V$ induces a forest in $G = (V, E)$ if and only if $V \setminus S$ is a decycling set of G. The problem of obtaining an acyclic subgraph of a graph G by removing vertices was already considered by Kirchhoff (1847) in an early work on spanning trees. Erdős et al. (1986) addressed this problem stated in terms of maximum induced trees.

Finding a minimum decycling set in a graph is listed among the NP-complete problems in the seminal paper of Karp (1972). As a matter of fact, there is no efficient algorithm for this problem, unless $P = NP$, even in special families of graphs such as bipartite graphs, planar graphs or perfect graphs. Nevertheless,

[†]The author was partly funded by IM-UFRGS and by FAPESP (Proc. FAPESP 2007/56496-3).

there exist polynomial-time algorithms to solve instances of this problem in cubic graphs, permutation graphs and interval graphs. Also, tighter bounds or even the exact value of the decycling number have been determined for bipartite graphs and for graphs such as grids and cubes.

The hardness of determining an exact solution to this problem for general graphs has motivated a rich literature on bounds on the maximum number of vertices in an induced forest. For instance, Alon et al. (1987) showed that, if the average degree of an n-vertex graph G is at most $d \geq 2$, then the maximum cardinality $\tau(G)$ of a subset of vertices inducing an acyclic subgraph of G satisfies $\tau(G) \geq 2n/(d+1)$. Whenever $d+1$ divides n, this bound is tight, as shown by the graph containing $n/(d+1)$ disjoint copies of the graph K_{d+1}. In the case of triangle-free graphs with maximum degree Δ, Alon et al. (2001) established that $\frac{\tau(G)}{n} = \Omega\left(\frac{\log \Delta}{\Delta}\right)$ as Δ tends to infinity, which is tight for some classes of triangle-free graphs. Another result in their paper gives a lower bound on $\tau(G)$ as a function of its order n, its maximum degree Δ and its independence number $\alpha(G)$, namely $\tau(G) \geq \alpha(G) + \frac{n-\alpha(G)}{(\Delta-1)^2}$, provided that $\Delta \geq 3$.

Results concerning asymptotic bounds on the size of a maximum induced forest in a random regular graph have also been obtained. This means that, if an r-regular graph on n vertices is chosen uniformly at random over all such graphs, then the probability that the size of a maximum induced forest satisfies such bounds tends to one as n tends to infinity. The best upper and lower bounds prior to the present work have been given by Bau et al. (2002). As is the case with several bounds for random r-regular graphs, their lower bounds are obtained through the analysis of a randomised algorithm over the probability space of random regular graphs. It makes use of a standard tool in this area, the so-called Differential Equation Method Wormald (1995). Hoppen and Wormald (2008) used a probabilistic algorithm for induced forests in regular graphs to conclude that the numerical values in Bau et al. (2002) are also deterministic lower bounds in the case of graphs with large girth, that is, *every* r-regular graph with sufficiently large girth has an induced forest whose size is at least the lower bound given in Bau et al. (2002).

In this paper, we extend the procedure in Hoppen and Wormald (2008) to a class of iterative probabilistic algorithms in regular graphs, which we call *neighbourly algorithms*. The main feature of this class is that it allows for prioritised algorithms, that is, algorithms in which vertices that benefit the algorithm the most are chosen with higher probability.

For a particular choice of parameters, the bounds obtained through the analysis of neighbourly algorithms surpass the results of Hoppen and Wormald (2008). More precisely, we prove the following result. Given $r \geq 3$, consider the constant $\xi(r)$ defined in (10) in terms of the solutions to the system of differential equations (9) with initial conditions given with $p_0 = 0$.

Theorem 1.1 *Let $\delta > 0$ and $r \geq 3$. Then there exists $g > 0$ such that every r-regular graph G on n vertices with girth greater than or equal to g satisfies $\tau(G) \geq (\xi(r) - \delta)n$.*

Numerical approximations of the numbers $\xi(r)$ are given in Table 1, and have been obtained by numerically solving this system of differential equations. The numbers $\xi_1(r)$ and $\Xi(r)$ in the table are the lower bounds obtained in Hoppen and Wormald (2008) and the upper bounds found in Bau et al. (2002), respectively. The numbers $\xi(r)$ improve the bounds previously known for all values of $r \geq 4$ for which they were calculated. If $r = 3$, the best lower bound is $\tau(G) \geq (0.75 - \delta)n$, for every $\delta > 0$, implied by the work of Edwards and Farr (2001). As we shall see, the bounds in this paper are also new best asymptotic bounds for random regular graphs, improving, for all $r \geq 4$, the bounds first given in Bau et al. (2002).

Tab. 1: Lower and upper bounds on $\tau(G)/n$

r	$\xi(r)$	$\xi_1(r)$	$\Xi(r)$	r	$\xi(r)$	$\xi_1(r)$	$\Xi(r)$
3	0.7368	0.7268	0.7500	7	0.4746	0.4283	0.5403
4	0.6351	0.6045	0.6667	8	0.4415	0.3940	0.5086
5	0.5662	0.5269	0.6216	9	0.4137	0.3658	0.4811
6	0.5149	0.4711	0.5776	10	0.3898	0.3419	0.4570

It is not hard to see that, given a graph G with maximum degree r, we may construct a graph G' by taking copies of G and joining vertices in different copies so as to make G' r-regular. This can be done without decreasing the girth if sufficiently many copies of G are used. Moreover, we have the inequality $\tau(G)/n \geq \tau(G')/|V(G')|$, because the copy of G containing the most vertices in a largest induced forest in G' satisfies this property. Thus the bounds in Theorem 1.1 may be translated into bounds for graphs with large girth and maximum degree r.

Corollary 1.2 *Let $\delta > 0$ and $r \geq 3$. Then there exists $g > 0$ such that every graph G on n vertices with maximum degree r and girth greater than or equal to g satisfies $\tau(G) \geq (\xi(r) - \delta)n$.*

Moreover, it is a well known fact (see for instance Bollobás (1980)) that, for fixed integers r and g, a random r-regular graph G may asymptotically almost surely be turned into a graph G' with maximum degree r and girth at least g with the deletion of at most $\log n$ vertices. Corollary 1.2 then leads to the following result.

Corollary 1.3 *Let $\delta > 0$ and $r \geq 3$. Then a random r-regular graph G on n vertices asymptotically almost surely satisfies $\tau(G) \geq (\xi(r) - \delta)n$.*

We shall see that, although asymptotic lower bounds on random regular graphs do not directly imply the bounds on regular graphs with large girth, our analysis resembles the analysis of an algorithm in a random graph. In other words, the random regular case is a source of inspiration in the design of the algorithm, and part of the analysis consists of proving that the distribution of vertices of each type in an instance of the algorithm is "random-graph-like" when the girth of the input graph is sufficiently large.

Furthermore, for graphs with large girth, the cardinality of a maximum induced forest is related to the concept of fragmentability of a graph. Given a constant $\lambda > 0$ and an integer m, a graph is (λ, m)-*fragmentable* if there is a set $X \subseteq V(G)$ such that $|X| \leq \lambda|V(G)|$ and $Y = V(G) \setminus X$ is m-*fragmented*, that is, every component of $G[Y]$ has at most m vertices. For an integer r, define $\lambda(r)$ as the infimum of all constants λ such that there is an m for which every graph with maximum degree at most r is (λ, m)-fragmentable. A remark in Haxell et al. (2008) relates, for a random r-regular graph G, the problem of finding the cardinality of a minimum decycling set in G and the problem of finding the infimum of all λ such that G is (λ, m)-fragmentable for some m. This relation can be easily restated in the case when G is a graph with large girth and maximum degree r: for any r, g and $\epsilon > 0$, there is an n_0 such that, for any r-regular graph G of order $n \geq n_0$ and girth larger than g, the size $\phi(G)$ of a minimum decycling set in G satisfies $|\phi(G) - n\lambda(r)| < \epsilon\, n$. Thus, the problem of finding bounds on $\lambda(r)$ and on the size of $\phi(G)$ of a minimum decycling set (and hence on the cardinality $\tau(G)$ of a largest induced forest) are equivalent. The best upper bound on the fragmentability of r-regular graphs with girth sufficiently large is $\lambda(r) \leq \frac{r-2}{r+1}$, as established by Edwards and Farr (2001). This bound gives the best possible result for

$r = 3$. Note that, in the case of induced forests, this implies the bound $\tau(G)/n \geq 0.75 - \delta$, where $\delta > 0$ is arbitrary and G is a graph on n vertices. However, for all values of $r \geq 4$ for which an approximation of the bound given in this paper is calculated, the bounds provided here are superior, and hence they lead to new best upper bounds for the fragmentability of regular graphs with large girth.

2 The algorithm

This section is devoted to the class of neighbourly algorithms. We define this class and prove some of its main properties. We then present a general framework under which the behaviour of such algorithms can be described in terms of the solution of a system of differential equations.

Algorithm 1 Neighbourly algorithm

Require: An r-regular graph G, a positive integer N, an initial probability p_0, and vectors of probabilities $\mathbf{p}_i = (p_{i,j,k} : 0 \leq j \leq 1, 0 \leq k \leq r - j)$, $i = 1, \ldots, N$.

Ensure: The set $\overline{P}$ of light purple vertices.

1: Start with all the vertices of the graph coloured white. As an initialization step, colour each vertex purple with probability p_0, at random, independently of all others. Purple vertices are light purple if no neighbours have also turned purple. Then colour non-purple vertices yellow if they have at least two purple neighbours.

2: **for** $i = 1, \ldots, N$ **do**

3: Choose a set S of white vertices, where a white vertex with j purple neighbours and k yellow neighbours is added to S randomly, independently of all others, with probability $p_{i,j,k}$. Colour all vertices in S purple, with a vertex being light purple if none of its neighbours lies in S. Then colour non-purple vertices yellow if they have at least two purple neighbours.

4: **end for**

It is clear that the output $\overline{P}$ of this algorithm induces a forest in the input graph, as a cycle can only be created in the set P of purple vertices if two adjacent vertices are chosen in the same step, in which case they are not light purple. We should keep in mind that, although deleting all dark purple vertices from the final forest is wasteful, this does not affect the asymptotic size of the forest produced. This could be easily optimised in an actual implementation of this algorithm.

Let $G = (V, E)$ be an r-regular graph with girth at least g. Let N be a fixed positive integer and consider a set of probabilities $p_{i,j,k}$, $i \geq 0$, $(j, k) \in \mathcal{I}$, where $\mathcal{I} = \{(j, k) \in \mathbb{Z}^2 : 0 \leq j \leq 1, 0 \leq k \leq r - j\}$. The random sets of white, yellow and purple vertices after i steps of the algorithm are denoted by W_i, Y_i and P_i, respectively, while $W_i^{j,k}$ is the set of white vertices with j purple neighbours and k yellow neighbours after i steps. We shall analyse the performance of Algorithm 1 by calculating the probability of several events related to this randomised algorithm, such as the probability of a fixed vertex having some given colour at a fixed time. To achieve this, some independence results will be proved.

The first, which we call *independence of vertex labelling*, shows that the colouring produced by the algorithm in a small connected subgraph H of G depends only on the isomorphism type of H, and not on the particular vertices in the subgraph, where "small" is measured with respect to the girth of the graph. More precisely, let $u \in V$ and let $u_1, \ldots, u_r$ be its neighbours. For each $s \in \{1, \ldots, r\}$ and positive integer $m < g/2$, the component $T_{u,s,m}$ of $G[\{v : d(u, v) \leq m\} \setminus u]$ containing u_s is a tree, which we consider as a rooted tree with root u_s. We shall refer to these trees as *branches around* u.

Lemma 2.1 ***(Independence of vertex labelling)*** *Let $u \in V$ and fix nonnegative integers i and t. Consider t distinct neighbours $u_{s_1}, \ldots, u_{s_t}$ of u and let $d'_1, \cdots, d'_t$ be positive integers such that $2(2i+1+\max\{d'_j\}) < g$. Then the probability that u has colour c and each T_{u,s_j,d'_j} has colouring χ_{j,d'_j} at time i is independent of u and of the set of neighbours $\{u_{s_1}, \ldots, u_{s_t}\}$.*

In particular, the values of $w_{i,j,k} = \mathbf{P}(u \in W_i^{j,k})$, where u is a vertex of G, are independent of the particular vertex u, as long as i is small in terms of the girth of g. The objective now is to calculate these probabilities as functions of $r \geq 3$, and of the probabilities p_0 and $(p_{i,j,k})_{0 \leq i \leq N,(j,k) \in \mathcal{I}}$ when the input graph G is r-regular and has girth g greater than $4N+4$. In the case when $i = 0$, these numbers depend only on r and p_0. Straightforward calculations lead to the formula

$$w_{0,j,k} = \binom{r}{j}\binom{r-j}{k} p_0^j (1-p_0)^{r-j+1} \left(1 - h(r,p_0)\right)^k h(r,p_0)^{r-j-k}, \tag{1}$$

where $h(r,p_0) = \sum_{s=0}^{1} \binom{r-1}{s} p_0^s (1-p_0)^{r-s}$. When $i > 0$, we aim to express the vector $\mathbf{w}_i = (w_{i,j,k} : (j,k) \in \mathcal{I})$ as a function of $\mathbf{p}_i = (p_{i,j,k} : (j,k) \in \mathcal{I})$ and $\mathbf{w}_{i-1}$, which would allow us to inductively calculate all the values in $\mathbf{w}_i$ through a recurrence relation. To this end, we need a second independence result, which we call *conditional independence of branches*. It specifies conditions under which the colourings of a set of branches rooted at different neighbours of u are mutually independent. As usual, we say that a collection of events $H_1, ..., H_m$ is *mutually independent* if, for any subset of the collection, the joint probability of all events is equal to the product of the probabilities of the individual events.

Lemma 2.2 ***(Conditional independence of branches)*** *Let $u \in V$ and $A, B \subseteq N(u)$, $A \cap B = \emptyset$. Fix nonnegative integers i and d' such that $2(2i+1+d')+1 < g$. Consider colourings χ_t of $T_{u,t,d'}$, $u_t \in N(u) \setminus (A \cup B)$ and let C_t be the event that $T_{u,t,d'}$ has colouring χ_t at time i. Then, conditional upon the event $\{u \in W_i \wedge (N(u) \cap P_i = A) \wedge (N(u) \cap Y_i = B)\}$, the events C_t are mutually independent.*

The importance of these independence results in the calculation of the probability of events in the algorithm is illustrated by the following corollary.

Corollary 2.3 *Let $u \in V$, $A, B \subseteq N(u)$, $A \cap B = \emptyset$, and $u_t \in N(u) \setminus (A \cup B)$. Fix nonnegative integers i and d' such that $2(2i+1+d')+2 < g$. Let χ_t be a colouring of the branch $T_{u,t,d'}$ and consider the event C_t that $T_{u,t,d'}$ has colouring χ_t at time i. Then*

$$\mathbf{P}(C_t \mid u \in W_i \wedge (N(u) \cap P_i = A) \wedge (N(u) \cap Y_i = B)) = \mathbf{P}(C_t \mid u \in W_i \wedge u_t \in W_i).$$

An important special case of this result is that the probability $\mathbf{P}\left(v \in W_i^{j,k} \mid u \in W_i^{j',k'} \wedge v \in W_i\right)$ is independent of j' and k' and can be denoted by $q_{i,j,k}$. We may obtain a formula for $q_{i,j,k}$ in terms of the probabilities $w_{i,j',k'}$, namely $q_{i,j,k} = \frac{(r-j-k)w_{i,j,k}}{\sum_{j',k'}(r-j'-k')w_{i,j',k'}}$. This implies that, for any white vertex, the distribution of white neighbours of each type, that is, with given numbers of purple and yellow neighbours, is as if the algorithm was applied to a random regular graph.

3 A framework for analysis

Using the independence results from the previous section, we are able to obtain a system of recurrence equations of the form

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mathbf{F}(\mathbf{p}_i, \mathbf{w}_{i-1}) + \mathbf{E}(\mathbf{p}_i, \mathbf{w}_{i-1}),\ i = 1, \ldots, N, \tag{2}$$

where $\mathbf{F} = (F_{j,k})_{(j,k)\in\mathcal{I}}$ is given by the formula below. Details about the derivation of this formula are in the full paper.

$$\begin{aligned} F_{j,k}(\mathbf{p}_i, \mathbf{w}_{i-1}) = &- p_{i,j,k} w_{i-1,j,k} - \frac{\chi(\mathbf{p}_i, \mathbf{w}_{i-1})}{s(\mathbf{w}_{i-1})}(r-j-k)\, w_{i-1,j,k} \\ &+ \delta_{j=1} \frac{\chi(\mathbf{p}_i, \mathbf{w}_{i-1})}{s(\mathbf{w}_{i-1})}(r-j-k+1)\, w_{i-1,0,k} \\ &+ \frac{\lambda(\mathbf{w}_{i-1})\chi(\mathbf{p}_i, \mathbf{w}_{i-1})}{s(\mathbf{w}_{i-1})^2}\left(\delta_{k\geq 1}(r-j-k+1)w_{i-1,j,k-1} - (r-j-k)w_{i-1,j,k}\right). \end{aligned} \tag{3}$$

Here, δ_E denotes the characteristic function for the event E, and we have

$$\begin{aligned} s(\mathbf{w}_i) &= \sum_{j'=0}^{1} \sum_{k'=0}^{r-1-j} (r-j'-k') w_{i,j',k'}, \\ \chi(\mathbf{p}_{i+1}, \mathbf{w}_i) &= \sum_{j'=0}^{1} \sum_{k'=0}^{r-1-j'} p_{i+1,j',k'}(r-j'-k') w_{i,j',k'} \end{aligned}$$

and $\lambda(\mathbf{w}_i) = \sum_{k'=0}^{r-3}(r-k'-1)(r-k'-2)w_{i-1,1,k'}$. The vector function $\mathbf{E} = (E_{j,k})_{(j,k)\in\mathcal{I}}$ is such that every $E_{j,k} = E_{j,k}(\mathbf{p}_i, \mathbf{w}_{i-1})$ is a polynomial in the variables corresponding to the vector $\mathbf{p}_i$ whose coefficients are rational functions in the variables corresponding to $\mathbf{w}_{i-1}$; moreover, as a polynomial in the first $|\mathcal{I}|$ variables, $E_{j,k}$ contains only monomials of degree larger than one. This function $\mathbf{E}$ is called the *error function* associated with Algorithm 1, as its influence is negligible when the probabilities $p_{i,j,k}$ are small.

We now present a general strategy to analyse Algorithm 1. For fixed r and a given initial probability p_0, the vector of initial conditions $\mathbf{w}_0(\mathbf{p}_0) = (w_{0,j,k}(p_0))_{(j,k)\in\mathcal{I}}$ is defined as in (1). To apply Algorithm to an r-regular graph, we still need to define the number of steps N and the probabilities $p_{i,j,k}$, $1 \leq i \leq N$, $(j,k) \in \mathcal{I}$. For the latter, fix a positive integer m and a sequence of real numbers $x_{-1} = 0 < x_0 < x_1 < \cdots < x_m$. We specify bounded nonnegative functions $\hat{p}_{j,k} : [0, x_m) \to \mathbb{R}$ for every $(j,k) \in \mathcal{I}$, which are required to be piecewise continuous with discontinuities restricted to the set $\{x_0, \ldots, x_{m-1}\}$. Now, given a sufficiently small constant $\epsilon > 0$, we may define the probabilities $p_{i,j,k}$ as $p_{i,j,k} = p_{i,j,k}(\epsilon) = \epsilon \hat{p}_{j,k}(i\epsilon)$. Having the probabilities defined in this way is convenient, since we may consider arbitrarily small probabilities by letting ϵ go to zero. The real numbers in the set $\{x_0, \ldots, x_{m-1}\}$ are related with the points of *phase transition* in the algorithm, while x_{-1} and x_m give starting and termination points.

For any fixed $\epsilon > 0$, we consider an application of Algorithm 1 with the previous r, p_0 and $p_{i,j,k} = \epsilon \hat{p}_{j,k}(i\epsilon)$, for $i \in \{1, \ldots, N\}$ and $(j,k) \in \mathcal{I}$. The number of steps N is given by $N = \lfloor x_m/\epsilon \rfloor$. We also require the input graph G to have girth larger than $4N+4$ to ensure that independence of vertex labelling and independence of branches hold throughout the application of the algorithm. In particular, we look at Algorithm 1 as a randomised function of the parameters r, p_0, ϵ and $\hat{\mathbf{p}} = (\hat{p}_{j,k})_{(j,k)\in\mathcal{I}}$. However, as r and $\hat{\mathbf{p}}$ will be fixed throughout the discussion, we shall treat the algorithm as a function of ϵ and p_0 and call it $\mathcal{A}(\epsilon, p_0)$.

Let $w_{i,j,k}^{\epsilon,p_0}$ denote the probability $\mathbf{P}(u \in W_i^{j,k})$ for fixed values of ϵ and p_0. By our previous discussion,

the quantities $w_{i,j,k}^{\epsilon,p_0}$ satisfy a system of recurrence equations of the form

$$\begin{aligned} &w_{i,j,k}^{\epsilon,p_0} = w_{i-1,j,k}^{\epsilon,p_0} + F_{j,k}(\epsilon\hat{\mathbf{p}}(i\epsilon), \mathbf{w}_{i-1}^{\epsilon,p_0}) + E_{j,k}(\epsilon\hat{\mathbf{p}}(i\epsilon), \mathbf{w}_{i-1}^{\epsilon,p_0}),\ i = 1, \ldots, N,\ (j,k) \in \mathcal{I}, \\ &w_{0,j,k}^{\epsilon,p_0} = w_{0,j,k}(p_0),\ (j,k) \in \mathcal{I}. \end{aligned} \tag{4}$$

When the error function is negligible, the behaviour of this system is determined by the functions $f_{j,k}(x, \mathbf{y}) = \frac{1}{\epsilon}F_{j,k}(\epsilon\hat{\mathbf{p}}(x), \mathbf{y}) = F_{j,k}(\hat{\mathbf{p}}(x), \mathbf{y})$, which, for $(j,k) \in \mathcal{I}$, describe the expected rate of change of the variable indexed by it in an application of $\mathcal{A}(\epsilon, p_0)$. To emphasise the fact that each phase is analysed separately, we use the notation $f_{j,k}^{(t)}(x, \mathbf{y}) = f_{j,k}(x, \mathbf{y})$ for every $(j,k) \in \mathcal{I}$, $t \in \{0, \ldots, m\}$ and $x \in [x_{t-1}, x_t)$. With foresight, it is important to assume that these functions satisfy additional technical conditions. First, for any fixed constants $M, \gamma > 0$, we define the region

$$\Omega_{\gamma,M} = \{(x, \mathbf{y}) \in \mathbb{R}^{|\mathcal{I}|+1} : 0 \leq x \leq M,\ y_{0,0} \geq \gamma \text{ and } 0 \leq y_{j,k} \leq M, \text{ for every } (j,k) \in \mathcal{I}\}, \tag{5}$$

and we consider $\gamma > 0$ and $M > x_m$ in such a way that the vector of initial conditions $(0, (w_{0,j,k}(p_0)))$ lies in $\Omega_{\gamma,M}$. Note that γ and M may be chosen as functions of p_0 only. We assume that the following conditions hold.

- (P_1) The coefficients of the polynomials $E_{j,k} = E_{j,k}(\mathbf{p}, \mathbf{w})$, which are rational functions in the variables $\mathbf{w}$, do not have poles in the region $\Omega_{\gamma,M}$.
- (P_2) Each function $f_{j,k}$ is defined over the region $\Omega_{\gamma,M}$, and the functions $f_{j,k}^{(t)}$ are Lipschitz continuous in the region $\Omega_{\gamma,M} \cap \left([x_{t-1}, x_t) \times \mathbb{R}^{|\mathcal{I}|}\right)$, for each $t \in \{0, \ldots, m\}$ and $(j,k) \in \mathcal{I}$.
- (P_3) There exist functions $\hat{\mathbf{w}}^{p_0}(x) = (\hat{w}_{j,k}^{p_0}(x))_{(j,k)\in\mathcal{I}}$ defined for x in the interval $[0, x_m)$ such that $(x, \hat{\mathbf{w}}^{p_0}(x)) \in \Omega_{\gamma,M}$ for every x and that

$$\begin{aligned} &\frac{d\hat{w}_{j,k}^{p_0}}{dx} = f_{j,k}^{(t)}(x, \hat{\mathbf{w}}^{p_0}) \text{ in the interval } [x_{t-1}, x_t),\ t = 0, \ldots, m \\ &\hat{w}_{j,k}^{p_0}(x_{t-1}) = \beta_{t-1,j,k}, \end{aligned} \tag{6}$$

  where $\beta_{t-1,j,k}$ is equal to the initial condition $w_{0,j,k}(p_0)$, if $t = 0$, and is inductively defined as $\lim_{x\to x_{t-1}^-} \hat{w}_{j,k}^{p_0}(x)$ if $t > 0$.

Intuitively, the first condition ensures that the influence of the error term $E_{j,k}(\epsilon\hat{p}(i\epsilon), \mathbf{w}_{i-1}^{\epsilon,p_0})$ in equation (4) is negligible in comparison with the influence of $F_{j,k}(\epsilon\hat{p}(i\epsilon), \mathbf{w}_{i-1}^{\epsilon,p_0})$. Combined with the fact that $F_{j,k}(\epsilon\hat{p}(i\epsilon), \mathbf{w}_{i-1}^{\epsilon,p_0}) = \epsilon f_{j,k}(i\epsilon, \mathbf{w}_{i-1}^{\epsilon,p_0})$ for $i = 1, \ldots, N$, this suggests that the system of differential equations (6) is a natural approximation of the system of recurrence equations (4) as ϵ tends to zero.

Condition (P_3) establishes that the system of differential equations (6) can be solved, while condition (P_2) guarantees that the functions describing the behaviour of the algorithm are well-behaved within each phase. Using essentially the proof of convergence of Euler's method for the solution of differential equations, we may establish that, as ϵ tends to zero, the solutions to (4) converge uniformly to the solutions to (6) within the region $\Omega_{\gamma,M}$. As a matter of fact, to ensure that the comparison between these systems occurs within the region $\Omega_{\gamma,M}$, we define the *final step* $N_f = N_f(p_0, \epsilon)$ as the last step for which $(i\epsilon, \mathbf{w}_i^{\epsilon,p_0})$ is inside $\Omega_{\gamma,M}$. Observe that, because the solutions of (4) are probabilities in a well defined probability space, the only reason for $(i\epsilon, \mathbf{w}_i^{\epsilon})$ to leave $\Omega_{\gamma,M}$ is that $w_{i,0,0}^{\epsilon,p_0}$ becomes smaller than γ or that some of $w_{i,j,k}^{\epsilon,p_0}$ becomes larger than M. Note that the latter never occurs if we set M to be larger than 1, which will be the case in our applications.

Lemma 3.1 *For any $\xi > 0$ and $p_0 \in (0,1)$, there exists $\epsilon' > 0$ such that, if $0 < \epsilon < \epsilon'$, then $\left| w_{i,j,k}^{\epsilon,p_0} - \hat{w}_{j,k}^{p_0}(i\epsilon) \right| < \xi,\ i = 0, 1, \ldots, N_f = N_f(\epsilon, p_0)$.*

One of the consequences of this result is that $\epsilon N_f \to x_m$ as $\epsilon \to 0$. To see why this is true, observe that, in the interval $[0, x_m)$, the derivative of $\hat{w}_{0,0}^{p_0}$ is negative, hence $\hat{w}_{0,0}^{p_0}(x)$ is a decreasing function lying within $\Omega_{\gamma,M}$. Moreover, Lemma 3.1 implies that $w_{i,0,0}^{\epsilon,p_0}$ can be made arbitrarily close to $\hat{w}_{0,0}^{p_0}(i\epsilon)$ for ϵ sufficiently small. This prevents the condition $w_{i,0,0}^{\epsilon,p_0} \geq \gamma$ from being violated until i is very close to x_m/ϵ, which establishes our claim.

In the following, for each $0 < \epsilon < 1/C$, where C is an upper bound on the values assumed by the bounded functions $\hat{p}_{j,k}$, for every $(j,k) \in \mathcal{I}$, let $\bar{P}(\epsilon, p_0)$ be the random set of light purple vertices in an instance of $\mathcal{A}(\epsilon, p_0)$ running for $N_f = N_f(\epsilon, p_0)$ steps. Moreover, $\hat{p}_{j,k}$, $\hat{w}_{j,k}$ and x_m are defined as in the previous discussion, and the conditions (P_1), (P_2) and (P_3) are satisfied.

Lemma 3.2 *Let $r \geq 3$ be an integer. Given $\delta > 0$ and $p_0 \in (0,1)$, there exists $\epsilon' > 0$ such that, if $0 < \epsilon < \epsilon'$ and G is an r-regular graph with n vertices and girth larger than $4N_f(\epsilon', p_0) + 4$, we have*

$$\left| \mathbf{E}|\bar{P}(\epsilon, p_0)| - n \left(p_0(1-p_0)^r + \int_0^{\epsilon N_f} \sum_{(j,k)\in\mathcal{I}} \hat{p}_{j,k}(x)\hat{w}_{j,k}(x)\, dx \right) \right| \leq \delta n.$$

Because the expected behaviour of this algorithm is the same for every r-regular graph with girth sufficiently large, a bound on $\tau(G)$ may be obtained directly from the above theorem.

Corollary 3.3 *Let $r \geq 3$ be an integer. Given $\delta > 0$ and $p_0 \in (0,1)$, there exists $g > 0$ such that every r-regular graph G on n vertices with girth greater than or equal to g satisfies*

$$\tau(G) \geq n \left(p_0(1-p_0)^r + \int_0^{x_m} \sum_{(j,k)\in\mathcal{I}} \hat{p}_{j,k}(x)\hat{w}_{j,k}(x)\, dx - \delta \right).$$

The lower bounds on the size of an induced forest obtained in Hoppen and Wormald (2008) can also be analysed in this framework, as they are derived from the analysis of a particular case of Algorithm 1, namely the case where $p_{i,1,k} = p$ and $p_{i,0,k} = 0$ for every $i > 0$, where p is any given positive constant.

4 Proof of Theorem 1.1

The aim of this section is to discuss the ideas behind the definition of the constants $\xi(r)$ in the statement of Theorem 1.1, as well as the validity of this theorem, which is an application of the framework of the previous section with an appropriate choice of parameters.

Before describing the choice of parameters, we address the intuition behind our approach. An *operation* consists of selecting a white vertex v, colouring it purple, and possibly colouring some of its neighbours yellow according to the rules of the algorithm. An operation is said to be of *type* (j,k) if v has j purple and k yellow neighbours prior to its selection. We rank operations of different types according to their benefit to the algorithm, and the probability functions $\hat{p}_{j,k}(x)$ are fixed so as to give priority to higher-rank operations. One criterion is quite natural: a white vertex with many yellow neighbours is high on the priority list, as any purple vertex obtained in this way is expected to cause fewer white vertices to turn yellow. As a second criterion, note that, because Algorithm 1 can only capture local information, an

unchosen vertex with multiple purple neighbours is automatically discarded (i.e., it turns yellow), even if its addition to the forest would not create cycles, but only merge some of its components. Thus creating extra components after the initial step of the algorithm is wasteful, and we avoid selecting white vertices with no purple neighbour after the initial step of the algorithm.

For convenience, white vertices devoid of white neighbours are also not selected. This is because, on the one hand, they cannot become yellow in a later step, and, on the other hand, they can all be added to the induced forest at the end of the algorithm without creating cycles.

In light of the above, we may restrict the above set of operations and say that the algorithm performs an *operation of type* t when it chooses a white vertex adjacent to one purple vertex and to t yellow vertices, where $t \in \{0, \dots, r-2\}$ and priority increases with t. To take maximum advantage of this ranking, one would ideally choose, at a given step i, only eligible vertices with the highest ranking. Algorithms of this type are often called *degree-greedy* and have been studied in several contexts when the input is a random regular graph; see for instance (Wormald (1995), Wormald (2003)).

We now illustrate the behaviour that we would like to mimic. Suppose that the graph to which this algorithm is applied is a large typical graph, or a random graph, and that the initialization step has already been taken. For simplicity, we discuss the behaviour of a degree-greedy algorithm for which a single vertex is processed in each iteration, with the vertex selected at step i being randomly chosen amongst all vertices with highest ranking. Early in the algorithm, if we look at the white vertices with one purple neighbour, most of them have no yellow neighbours. We say that the algorithm is in Phase 0, for which selecting a vertex with no yellow neighbours is the *basic* operation. Early in the process, it is unlikely that many white vertices with yellow neighbours will be created, and any such vertices created are chosen in the next few steps until none remain. As the algorithm evolves, however, the occasional white vertices with yellow neighbours become increasingly common, until we reach a point for which, as a white vertex with one yellow neighbour is chosen, more vertices of this type tend to be created, that is, they begin to regenerate themselves faster than they are consumed. Our algorithm now enters Phase 1, whose basic operation is to choose a white vertex with one purple neighbour and one yellow neighbour. In general, when the algorithm is in Phase k, the basic operation consists of selecting a vertex with one purple neighbour and k yellow neighbours. Again, white vertices with a purple neighbour and more yellow neighbours may be created during this phase, and they are all selected before another basic operation is performed. There will be a point in which we either run out of basic vertices, and the algorithm has to stop, or the vertices with one purple and $k+1$ yellow neighbours regenerate themselves faster than they are consumed, which marks the transition of the algorithm to Phase $k+1$.

Now, if we want to use these ideas to find parameters for Algorithm 1, we have to think of a "deprioritised" version of the above discussion, as we are not allowed to choose a single vertex at each step (there is a limited number of iterations until the independence provided by our independence lemmas breaks down, and so does our analysis). In other words, we will set the parameters so that the proportion of vertices of each type chosen by the algorithm forces the expected behaviour to follow the previous characterization, even though there is no guarantee that vertices with the highest ranking are chosen.

To this end, let α_ℓ denote the proportion of vertices of type ℓ among all the chosen vertices at some step of the algorithm. On the one hand, if the algorithm is in Phase t, we wish to have $\sum_{\ell=t}^{r-2} \alpha_\ell = 1$, as no vertices with fewer than t yellow neighbours should be chosen. On the other hand, using the independence obtained in the previous section, it is not hard to calculate the expected change on the proportion of vertices of each type if all the vertices processed have a single type, say type t. As a matter of fact, for $0 \le t \le r-2$ and $(j,k) \in \mathcal{I}$, the expected change in the proportion of vertices of type (j,k)

when all the vertices processed have type t is given by the formula

$$\begin{aligned}\phi_{j,k}{}^{(t)}(x,\mathbf{w}) &= -\delta_{k,t}\delta_{j,1} - (r-j-k)(r-t-1)w_{j,k}/s(\mathbf{w}) \\ &\quad + (r-t-1)\left((r-j-k+1)w_{j,k-1}\delta_{k\geq 1} - (r-j-k)w_{j,k}\right)\lambda(\mathbf{w})/s(\mathbf{w})^2 \\ &\quad + \delta_{j,1}(r-j-k+1)(r-t-1)w_{j-1,k}/s(\mathbf{w}),\end{aligned} \tag{7}$$

where $s(\mathbf{w}) = s(w_{0,0},\dots,w_{0,r},w_{1,0},\dots,w_{1,r-1}) = \sum_{j''=0}^{1}\sum_{k''=0}^{r-j}(r-j''-k'')w_{j'',k''}$ and $\lambda(\mathbf{w}) = \sum_{k''=0}^{r-3}(r-k''-1)(r-k''-2)w_{1,k''}$.

By the previous discussion, while the algorithm is in Phase t, white vertices with one purple neighbour and $k > t$ yellow neighbours should not regenerate themselves faster than they are consumed, which leads to the equations $\sum_{\ell=t}^{r-2}\alpha_\ell\phi_{1,k}^{(\ell)} = 0$ if $t < k \leq r-2$. Hence the proportion $\alpha_k^{(t)}$ of operations of type k performed by the algorithm while in Phase t, for a fixed $t \in \{0,\dots,r-2\}$, is given by the solution to the linear system

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \phi_{t+1}^{(t)} & \phi_{t+1}^{(t+1)} & \cdots & \phi_{t+1}^{(r-2)} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \phi_{r-2}^{(t)} & \phi_{r-2}^{(t+1)} & \cdots & \phi_{r-2}^{(r-2)} \end{bmatrix} \begin{bmatrix} \alpha_t^{(t)} \\ \alpha_{t+1}^{(t)} \\ \cdot \\ \cdot \\ \alpha_{r-2}^{(t)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}. \tag{8}$$

Let $M^{(t)}$ denote the matrix in this equation. When its determinant is nonzero, the unique solution to this system is given by $\alpha_t^{(t)}(x,\mathbf{w}) = \frac{1-\sum_{\ell=t+1}^{r-2}(r-\ell-1)a_\ell}{1+\sum_{\ell=t+1}^{r-2}(\ell-t)a_\ell}$ and $\alpha_k^{(t)}(x,\mathbf{w}) = \frac{(r-t-1)a_k}{1+\sum_{\ell=t+1}^{r-2}(\ell-t)a_\ell}$, if $k > t$, where $a_k = a_k(x,\mathbf{w}) = -(r-k-1)w_{1,k}/s + (r-k)w_{0,k}/s + [(r-k)w_{1,k-1} - (r-k-1)w_{1,k}]\lambda/s^2$, with $s = s(\mathbf{w})$ and $\lambda = \lambda(\mathbf{w})$ defined as before.

Given $p_0 \in (0,1)$, and for constants $0 = x_{-1} < x_0 < \cdots < x_m$, this discussion suggests the system of differential equations

$$\frac{dw_{j,k}^{p_0}}{dx} = f_{j,k}^{(t)}(x,\mathbf{w}(x)) \text{ for } x \in [x_{t-1},x_t),\ w_{j,k}^{p_0}(x_t) = \mu_{t,j,k}^{p_0} \text{ for every } (j,k) \in \mathcal{I}, \tag{9}$$

where $f_{j,k}^{(t)}(x,\mathbf{w}) = \sum_{\ell=t}^{r-2}\alpha_\ell^{(t)}\phi_{j,k}^{(\ell)}$ and the numbers $\mu_{t,j,k}^{p_0}$ are defined as follows. If $t = -1$, it is equal to the initial condition $w_{0,j,k}(p_0)$ defined in equation (1). If $0 \leq t \leq m$, $\mu_{t,j,k}^{p_0} = \lim_{x\to x_t^-} w_{j,k}^{p_0}(x)$.

To prove Theorem 1.1, we proceed in a somewhat counterintuitive way, which mirrors the analysis of deprioritised algorithms undertaken in Wormald (2003) in the context of random regular graphs. We first study the differential equations (9) with $p_0 = 0$, and we show that there is a nonnegative integer b and real numbers $0 < x_0 < \ldots < x_b$ so that there is a solution $\mathbf{w}$ to (9) such that $0 \leq w_{j,k}(x) \leq 1$ for x in the interval $[0,x_b)$. Moreover, the functions $\alpha_k^{(t)}(x,\mathbf{w}(x))$ also lie in $[0,1]$ for $x \in [0,x_b)$, $t \in \{0,\dots,b\}$ and $k \in \{t,\dots,b\}$.

By analysing the sensitivity of the system to its initial conditions, we may then extend this to small values of p_0, that is, we may show that there is a positive constant p_0' for which we may define well-behaved phase-transition functions $x_0,\dots,x_b : [0,p_0'] \to \mathbb{R}_+$ such that, for $0 < p_0 < p_0'$, there is a solution $\mathbf{w}^{p_0} = (w_{j,k}^{p_0})$ of (9) satisfying $0 \leq w_{j,k}^{p_0}(x) \leq 1$ for every $(j,k) \in \mathcal{I}$ and every x in the

interval $[0, x_b(p_0))$, with phase transitions at the points $x_0(p_0) < \ldots < x_{b-1}(p_0)$. We also have $0 \leq \alpha_k^{(t)}(x, \mathbf{w}^{p_0}(x)) \leq 1$ for $x \in [0, x_b(p_0))$, for all $t \in \{0, \ldots, b\}$ and $k \in \{t, \ldots, r-2\}$. We are then able to appropriately set the functions $\hat{p}_{j,k}$ based on these solutions. In other words, we use solutions to the desired system of differential equation to define the parameters in such a way that the behaviour of Algorithm 1 is described by these differential equations.

The details of this process are quite technical and will apear in the full paper. In what follows, we informally describe the main ideas involved in the study of the differential equations (9) with $p_0 = 0$, which is done inductively as follows. We define a constant γ such that the initial conditions lie in the region $\Omega_{\gamma,1}$. The main tool for extending the initial conditions to solutions within this region is the following standard result in the theory of first order differential equations (see Hurewicz (1958)).

Lemma 4.1 *If a set of functions $f_i : \mathbb{R}^{s+1} \to \mathbb{R}$ is Lipschitz continuous in a bounded region Ω and the point $(x', y_1', \ldots, y_s')$ lies in Ω, then the solution of the system of differential equations*

$$\frac{dz_i}{dx} = f_i(x, z_1, \ldots, z_s), i = 1, \ldots, s$$

with initial conditions $z_i(x') = y_i',\ i = 1, \ldots, s$ may be uniquely extended arbitrarily close to the boundary of Ω.

There are two main concerns as we apply this result to Phase 0 of the system of differential equations (9) with $p_0 = 0$. First we wish this phase to be non-degenerate, that is, that the solutions may extended within $\Omega_{\gamma,1}$ for x in an interval $[0, x_0)$ with $x_0 > 0$. Moreover, we wish to ensure that the quantities $\alpha_k^{(0)}(x, \mathbf{w}(x))$ represent proportions, that is, they lie in the interval $[0, 1]$. To this end, at the start of the phase, we verify that: (i) the determinant of the matrix $M^{(0)}$ in (8) is bounded away from zero at the point $(0, \mathbf{w}(0))$; (ii) for all (j, k), either the value of $w_{j,k}(0)$ is bounded away from the corresponding boundary in $\Omega_{\gamma,1}$, or its first non-zero derivative is bounded away from zero in such a way that the solution points into the region; (iii) for all k, if $\alpha_k^{(0)}(0, \mathbf{w}(0))$ is not identically zero, it is either bounded away from the boundaries of the interval $[0, 1]$, or its first non-zero derivative is bounded away from zero in such a way that it points into the interval $(0, 1)$.

There are several natural conditions for Phase 0 to end. For instance, the solution $(x, \mathbf{w}(x))$ could approach the boundary of $\Omega_{\gamma,1}$, which would prevent the solution of (9) from being extended further based on Lemma 4.1. Moreover, the determinant $\det(M^{(0)}(x, \mathbf{w}(x))$ could approach zero, or the proportions $\alpha_k^{(0)}(x, \mathbf{y}(x))$ could become negative or larger than one for some value of k. Let x_0 be the infimum of all $x > 0$ for which at least one such conditions hold. The initial conditions tell us that $x_0 > 0$. If any termination condition other than $\alpha_0^{(0)}(x, \mathbf{w}(x)) = 0$ is active at x_0, or if $\alpha_0^{(0)}(x_0, \mathbf{w}(x_0)) = 0$ but its derivative with respect to x is nonnegative at this point, Phase 0 is said to be the *final phase*, and we fix $b = 0$ and $x_b = x_0$. The reason behind this condition is that $\alpha_0^{(0)}(x_0, \mathbf{w}(x_0)) = 0$ implies that the derivative of $\phi_{1,1}^{(1)}(x, \mathbf{w}(x))$ equals zero at the point x_0, that is, vertices of type $(1, 1)$ start to regenerate themselves faster than they are consumed when they are processed by the algorithm. Moreover, if $\alpha_0^{(0)}(x_0, \mathbf{w}(x_0)) = 0$ is the single termination condition that holds at x_0, we ensure that the starting conditions for the following phase, namely Phase 1, are satisfied, which implies that it can be extended to a point $x_1 > x_0$, the infimum of all $x > x_0$ for which one of the termination conditions hold, that is, the determinant $\det(M^{(1)}(x, \mathbf{w}(x))$ approaches zero, the proportions $\alpha_k^{(1)}(x, \mathbf{w}(x))$ become negative or larger than one for some value of k, or the solution $\mathbf{w}(x)$ approaches the boundary of $\Omega_{\gamma,1}$. Once again,

Phase 1 is said to be the final phase if any termination condition other than $\alpha_1^{(1)}(x, \mathbf{w}(x)) = 0$ is active at x_1, or if $\alpha_1^{(1)}(x_1, \mathbf{w}(x_1)) = 0$, but its derivative with respect to x is nonnegative, otherwise Phase 2 starts.

Let b denote the index of the final phase. We first discuss the dependency of b and $x_0, \ldots, x_b$ on the constant γ. On the one hand, it is clear that, by choosing a smaller value of γ, there is no change in the values of $x_0, \ldots, x_{b-1}$, since all the conditions and termination conditions would be verified in the same way. However, if the only active termination condition at x_b is $w_{1,b}(x_b) = \gamma$, it may be the case that, by decreasing γ, the solution of the differential equation could be extended beyond x_b. It is even conceivable that a different termination condition would then become active, which could potentially originate a new phase. However, if $b(\gamma)$ denotes the index of a final phase for a particular choice of γ, the limit $b = \lim_{\gamma \to 0^+} b(\gamma)$ is well defined, since $b(\gamma)$ is bounded above by $r - 2$ and is non-decreasing as γ decreases. Moreover, since $b(\gamma)$ is integer-valued, we know that this limit is achieved for γ sufficiently small. For this value of b, we may also define $x_b = \lim_{\gamma \to 0^+} x_b(\gamma)$, since $x_b(\gamma)$ is non-decreasing as γ decreases, and it is easy to see that 1 is an upper bound on its value. We may now define the constant

$$\xi(r) = x_b + w_{0,r}(x_b) + w_{1,r-1}(x_b). \tag{10}$$

The lower bounds on $\xi(r)$ given in Table 1 were obtained by "solving" the system of differential equations numerically, without using careful error bounds, but just apparent good convergence as the step size was made smaller. The points of phase transition were determined by the termination conditions. However, since the points of phase transition are not determined exactly, we also verified that, if we slightly perturb the points in which there is transition from one differential equation to the next, the overall change in the solutions is very small. Furthermore, we observed that, in a small interval in which a phase transition seems to occur, the values given by the numerical calculations suggest that the remaining conditions are "far" from being satisfied, hence we have numerically verified that the appropriate termination condition is active at the end of the phase, which ensures that the next phase does start.

To conclude the paper, we would like to emphasise that the main contribution of this work lies in the method presented rather than in these particular new bounds, as it allows us, in some sense, to directly analyse prioritised algorithms in regular graphs. A similar approach may be applied to a wide range of problems in regular graphs with large girth.

References

N. Alon, J. Kahn, and P. D. Seymour. Large induced degenerate subgraphs in graphs. *Graphs and Combinatorics*, 3:203–211, 1987.

N. Alon, D. Mubayi, and R. Thomas. Large induced forests in sparse graphs. *Journal of Graph Theory*, 38:113–123, 2001.

S. Bau, N. Wormald, and S. Zhou. Decycling number of random regular graphs. *Random Structures & Algorithms*, 21:397–413, 2002.

B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1:311–316, 1980.

K. Edwards and G. Farr. Fragmentability of graphs. *Journal of Combinatorial Theory, Series B*, 82:30–37, 2001.

P. Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

P. Erdős, M. Saks, and V. T. Sós. Maximum induced trees in graphs. *Journal of Combinatorial Theory, Series B*, 41:61–79, 1986.

P. Haxell, O. Pikhurko, and A. Thomason. Maximum acyclic and fragmented sets in regular graphs. *Journal of Graph Theory*, 57(2):149–156, 2008.

C. Hoppen and N. Wormald. Induced forests in regular graphs with large girth. *Combinatorics, Probability and Computing*, 17(3):389–410, 2008.

W. Hurewicz. *Lectures on Ordinary Differential Equations*. MIT Press, Cambridge, Mass., 1958.

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computation*, pages 85–103. Plenum, New York, 1972.

G. Kirchhoff. Über die auflösung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer ströme geführt wird. *Ann. Phys. Chem.*, 72:497–508, 1847.

N. Wormald. Differential equations for random processes and random graphs. *The Annals of Applied Probability*, 5(4):1217–1235, 1995.

N. Wormald. Analysis of greedy algorithms on graphs with bounded degrees. *Discrete Mathematics*, 273: 235–260, 2003.