Dominating Sets in Web Graphs*

Colin Cooper[1] Ralf Klasing[2†] Michele Zito[3]

[1]Department of Computer Science, King's College
London WC2R 2LS (UK). e-mail: ccooper@dcs.kcl.ac.uk

[2]MASCOTTE project, I3S-CNRS/INRIA, Université de Nice-Sophia Antipolis
2004 Route des Lucioles, BP 93, F-06902 Sophia Antipolis Cedex (France)
e-mail: Ralf.Klasing@sophia.inria.fr

[3]Department of Computer Science, University of Liverpool
Peach Street, Liverpool L69 7ZF (UK). e-mail: M.Zito@csc.liv.ac.uk

December 10, 2004

Abstract

In this paper we study the size of dominating sets, and their generalizations, in two graph processes which are widely used to model aspects of the world-wide web. In these processes each new vertex connects to the existing graph by a constant number, m, of edges. The terminal vertices of these edges are chosen uniformly at random or by preferential attachment depending on the process. We show that almost all such graph processes have minimal dominating sets linear in the size of the graph and give bounds for this size as a function of m. We obtain the upper bounds from simple on-line algorithms for dominating sets. The lower bounds are obtained by proving that the lexicographically first set of a given size is the most likely to dominate.

1 Introduction

In recent years the world wide web has grown dramatically. Its current size is measured in billions of pages [20], and pages are added to it every day. As this graph (nodes correspond to web pages and edges to links between pages) continues to grow it becomes increasingly important to study mathematical models which capture its structural properties [6, 19]. Such models can be used to design efficient algorithms for web applications and may even uncover unforeseen properties of this huge evolving structure. Several mathematical models for analysing the web have been proposed (for instance [5, 9, 19]). The (evolution of the) web graph is usually modelled by a (random) process in

*The first two authors were partially supported by a Royal Society Grant under the European Science Exchange Programme.

†Partially supported by the European projects RTN ARACNE (contract no. HPRN-CT-1999-00112) and IST FET CRESCCO (contract no. IST-2001-33135). Partial support by the French CNRS AS Dynamo.

which new vertices appear from time to time. Such vertices may be linked randomly to the existing structure through some form of *preferential attachment*: existing vertices with many neighbours are somewhat more likely to be linked to the newcomers.

The main focus of research so far has been on capturing empirically observed [3, 6] features of the web. No attempt has been made to characterise graph-theoretic sub-structures of such graphs. We initiate such investigation by looking at sets of vertices that, in a sense, cover all other vertices. More formally, a vertex in a graph *dominates* all vertices that are adjacent to it (parallel edges give multiple domination). In the spirit of Harary and Haynes [14], an *h-dominating set* for a graph $G = (V, E)$ is a set $\mathcal{S} \subseteq V$ such that each vertex in $V \setminus \mathcal{S}$ is dominated at least h times by vertices in $\mathcal{S}$. Let $\gamma_h = \gamma_h(G)$ denote the size of the smallest h-dominating sets in G. The *minimum h-dominating set* problem (MhDS) asks for an h-dominating set of size γ_h.

Dominating sets play an important role in many practical applications, e.g. in the context of distributed computing or mobile ad-hoc networks [2, 10, 22]. The reader is referred to [15, 16] for an in-depth view of the subject. The typical fundamental task in such applications is to select a subset of nodes in the network that will 'provide' a certain service to all other vertices. For this to be time-efficient, all other vertices must be directly connected to the selected nodes, and in order for it to be cost-effective, the number of selected nodes must be minimal. In relation to web graphs, a dominating set may be used to devise efficient web searches. For $h > 1$ an h-dominating set can be considered as a more fault-tolerant structure. If up to $h-1$ vertices or edges fail, the domination property is still maintained (i.e. it is still possible to provide the service).

The MhDS problem is NP-hard [13, 18] and, moreover, it is not likely that it may be approximated effectively [12, 18]. Polynomial time algorithms exist on special classes of graphs (e.g. [21]). The M1DS problem has been studied also in random graphs. In the *binomial model* $G(n, p)$ [24] the value of γ_1 can be pin-pointed quite precisely, provided p is not too small compared to n. In random regular graphs of degree r (see for example results in the *configuration model* [26] and references therein) upper and lower bounds are known on γ_1.

In this paper we look at simple and efficient algorithms for building small h-dominating sets in graphs. The performance guarantees of these algorithms are analysed under the assumption that the input is a random web graph. We also analyse the tightness of the performances of such algorithms, by proving combinatorial lower bounds on γ_h, for any fixed $h \geq 1$. Such bounds show that most of the time the sets returned by the various algorithms are only a constant factor away from optimal. Finally, we compare the quality of the solutions returned by (some of) our algorithms with empirical average values of γ_1.

The main outcome of this paper can be stated informally by saying that web graphs have fairly large dominating sets. Hence a crawler who wants to use a dominating set to explore the web will need to store a large proportion of the whole graph. Interestingly, the results in this paper also uncover a(nother) difference between models of the web based on preferential attachment and more traditional random graph models. The tendency to choose neighbours of high degree affects the size of the smallest dominating sets.

Most of our algorithms are *on-line* in the sense that the decision to add a particular vertex to the dominating set is taken without total information about the web graph under consideration, and *greedy* in the sense that decisions, once taken, are never changed. The algorithms are also quite efficient: only a constant amount of time is used per update of the dominating set. Such algorithms are of particular interest in the context of web graphs. As the web graph is evolving, one wants to decide whether a new vertex is to be added to the already existing dominating set without recomputing the existing dominating set and with minimal computational effort. On-line strategies for the dominating set problem have been considered in the past [11, 17] for general graphs. However the authors are not aware of any result on on-line algorithms for this problem in random graphs.

Our results hold *asymptotically almost surely* (a.a.s.), i.e. with probability approaching one as the size of the web graph grows to infinity. The algorithmic results are based on the analysis of a number of (Markovian) random processes. In each case the properties of the process under consideration lead to the definition of a (deterministic) continuous function that is very close (in probability) to the values of the process, as the size of the graph grows. It should be pointed out at this stage that the proposed analysis methodology is quite general. We apply it to analyse heuristics for the MhDS problem only, but it would allow to prove results about other graphs parameters such as the independence or the chromatic number. The method is closely related to the so called *differential equation method* [25]. In fact a version of the main analytical tool proposed by Wormald can be adapted to work for the processes considered in this paper. However the machinery employed in [25] is not needed to analyse the processes considered in this paper. Our results are obtained by proving concentration of the various processes of interest around their mean and by devising a method for getting close estimates on the relevant expectations. In Section 2 we review the definitions of the models of web graphs that we will use. We also state our main result in the context of these models, and present more detailed comments on our general analysis method. In the following section we consider a very simple algorithm and apply the proposed method to obtain non-trivial upper bounds on γ_1. Refined algorithms are introduced and analysed in Section 4 and 5. In Section 6 we discuss generalisations to $h > 1$. Then we turn to lower bounds. In Section 7 and 8 we present our argument for the lower bounds stated in Section 2. Finally we briefly comment on some empirical work carried out on a sub-class of the graphs considered in this paper.

2 Models and Results

The models used in this paper are based on the work of Albert and Barabasi [3]. A *web graph* (see also [9]) can be defined as an ever growing structure in which, at each step, new vertices, new edges or a combination of these can be added. Decisions on what to add to the existing graph are made at random based on the values of a number of defining parameters. The existence of these parameters makes the model very general. For the purposes of this paper, to avoid cluttering the description of our results, we prefer to make a drastic simplification. We will consider graphs generated according to two rather extreme procedures derived from the general model. In each case the generation process is governed by a single integer parameter m. The second of these mimics the prefential attachment phenomenon. The first one, related to more traditional random graph models, is considered mainly for comparison.

uar graph process. The initial graph $G_0^{R,m}$ is a single vertex v_0 with m loops attached to it. For $t \geq 1$, let $G_{t-1}^{R,m}$ be the graph generated in the first $t-1$ steps of this process, to define $G_t^{R,m}$ a new vertex v_t is generated, and it is connected to $G_{t-1}^{R,m}$ through m (undirected) edges. The neighbours of v_t are chosen uniformly at random (uar) with replacement from $\{v_0, \ldots, v_{t-1}\}$.

Pure copy graph process. The initial graph $G_0^{C,m}$ is a single vertex v_0 with m loops attached to it. For $t \geq 1$, graph $G_t^{C,m}$ is defined from $G_{t-1}^{C,m}$ by generating a new vertex v_t, and connecting it to $G_{t-1}^{C,m}$ through m (undirected) edges. A vertex $u \in \{v_0, \ldots, v_{t-1}\}$ is connected to v_t with probability $\frac{|\Gamma(u)|}{2mt}$ (where $\Gamma(u) = \{w : \{u,w\} \in E(G_{t-1}^{C,m})\}$).

We will often refer to the uar graph process as a random graph process. We use the word copy as interchangeable with preferential attachment. The edges have an intrinsic direction which we ignore.

Both models are dynamic, with new vertices and edges continuously increasing the size of the graph. However they represent two extreme cases. In the uar graph process the terminal vertex of an edge

Table 1: Numerical bounds for the minimum dominating set problem.

m	α_{lo}^R	α_{up}^R	α_{lo}^C	α_{up}^C
1	0.3678	0.5	0.2939	0.3333
2	0.1868	0.3714	0.1009	0.2342
3	0.1488	0.3054	0.0709	0.1777
4	0.1259	0.2634	0.0544	0.1422
5	0.1102	0.2335	0.0439	0.1178
6	0.0987	0.2110	0.0366	0.1000
7	0.0898	0.1932	0.0313	0.0865

is chosen randomly from the set of available vertices, whereas in the copy model the terminal vertex of an edge is chosen proportional to the current degree of the various vertices.

Recall that $\gamma_h(G_t)$ is the size of the minimal h-dominating set in G_t. We find positive constants α_{lo}, α_{up} such that $\alpha_{\mathrm{lo}} \cdot t \leq \gamma_h(G_t) \leq \alpha_{\mathrm{up}} \cdot t$ almost surely. The upper bounds come from the size of the dominating set returned by simple on-line algorithms, whereas the lower bounds come from probabilistic arguments.

More precisely: Let $M \in \{R, C\}$, and m be a positive integer. For each $h \geq 1$ there exist positive real constants α_{lo}^M and α_{up}^M (dependent on M, m and h but independent of t) with $\alpha_{\mathrm{lo}}^M < \alpha_{\mathrm{up}}^M < 1$ such that $\alpha_{\mathrm{lo}}^M \cdot t \leq \gamma_h(G_t^{M,m}) \leq \alpha_{\mathrm{up}}^M \cdot t$ a.a.s. For $h = 1$, the bounds are given in Table 1. Bounds for $h > 1$ are discussed in Section 6.

The proof of the algorithmic results are based on the fact that natural edge-exposure martingales can be defined on the graph processes under consideration [9]. More precisely, if $f(G)$ is any graph theoretic function (e.g. the size of the dominating set returned by a particular algorithm), the random process defined by setting $Z_0 = \mathbf{E}\ (f(G_t^{M,m}))$, and Z_i (for $i \in \{1, \ldots, mt\}$) to be the expectation of $f(G_t^{M,m})$ conditioned on the "exposure" of the first i edges in the graph process is a martingale. Notice that the space of all graphs which can be generated according to the given model $G_t^{M,m}$ is partitioned into classes (or *i-blocks*) containing all those graphs which coincide w.r.t. the first i edge exposures.

In the forthcoming sections we will repeatedly use the following concentration result (for a proof see, for instance, [1]).

Theorem 2.1 *Let $c = Z_0, \ldots, Z_n$ be a martingale with $|Z_{i+1} - Z_i| \leq 1$ for all $i \in \{0, \ldots, n-1\}$. Then $\mathbf{Pr}[|Z_n - c| > \lambda\sqrt{n}] \leq 2\mathrm{e}^{-\lambda^2/2}$.*

In all our applications $c = \mathbf{E}\ (f(G_t^{M,m}))$, $n = mt$ and $\lambda = O(\log t)$. In order to apply Theorem 2.1 one needs to prove that $|Z_{i+1} - Z_i| \leq 1$. Such inequality follows from the smoothness of f (i.e. $|f(G) - f(H)| \leq 1$ if G and H differ w.r.t. the presence of a single edge) and the ability to demonstrate the existence of a measure preserving bijection between $i+1$-blocks in a same i-block. This is obvious in the random graph process as edges are inserted independently. In the case of the copy model it is convenient to identify each graph with the projection of a particular type of *configuration*, i.e. an ordered collection of mt labelled pairs of points (see [4]). Let C_1 and C_2 be two such configurations that are identical up to the ith pair. Suppose the $i+1$-st pair is $\{a, b\}$ in C_1 and $\{a, c\}$ in C_2. If C_1 never uses point b again then the image of C_1 under the measure preserving bijection will be a configuration C' identical to C_1 except that pair $\{a, b\}$ is replaced by pair $\{a, c\}$. If b is used in a following pair (say $\{d, b\}$) of C_1 then C' will have $\{a, c\}$ instead of $\{a, b\}$ and $\{d, c\}$ instead of $\{d, b\}$, and so on. Similar construction is presented in [9].

Finally we will need the following

Lemma 1 *If Z_n is a martingale and there exist constants $c_1, c_2 > 0$ such that $\mu_n = \mathbf{E}\,(Z_n) \in [c_1 n, c_2 n]$, then for each fixed integer $j > 0$, there exists positive constant K and ϵ such that $|(Z_n)^j - (\mu_n)^j| \leq K n^{j-\epsilon}$ a.a.s..*

Proof. If Z_n is a martingale, then by Theorem 2.1, $\mu_n - \lambda\sqrt{n} \leq Z_n \leq \mu_n + \lambda\sqrt{n}$ where w.l.o.g. we assume $\lambda = o(\sqrt{n})$. From this we also have, for each fixed integer $j > 0$,

$$(\mu_n)^j (1 - \tfrac{\lambda\sqrt{n}}{\mu_n})^j \leq (Z_n)^j \leq (\mu_n)^j (1 + \tfrac{\lambda\sqrt{n}}{\mu_n})^j.$$

The result now follows (provided K is chosen big enough) since, the assumptions on λ and μ_n entail that $(1 + \frac{\lambda\sqrt{n}}{\mu_n})^j$ is at most $1 + \frac{j^2\lambda\sqrt{n}}{\mu_n}$, whereas $(1 - \frac{\lambda\sqrt{n}}{\mu_n})^j$ is at least $1 - \frac{2j\lambda\sqrt{n}}{\mu_n}$ □

3 Simplest algorithm

The algorithm presented in this section is a very simple "first attempt" solution for the problem at hand. Although in many cases it does not lead to a very small dominating set, it represents a natural benchmark for any more refined heuristic.

Algorithm 1. Before the first step of the algorithm the graph consists of a single vertex v_0 and $\mathcal{S} = \{v_0\}$. At step t if the newly generated vertex v_t does not have any neighbours in $\mathcal{S}$ (i.e. $\Gamma(v_t) \cap \mathcal{S} = \emptyset$) then v_t is added to $\mathcal{S}$.

In the forthcoming discussion m is a fixed positive integer. Let X_t denote the size of the dominating set $\mathcal{S}$ computed by Algorithm 1 before v_t is added to the current graph and let $\mu_t = \mathbf{E}\,(X_t)$. For graphs generated according to the $G_t^{R,m}$ model, the probability that v_t misses the dominating set is $(1 - \frac{X_t}{t})^m$. Hence we can write

$$\mu_{t+1} = \mu_t + \mathbf{E}\,[(1 - \tfrac{X_t}{t})^m].$$

Let $x = x(m)$ be the unique solution of the equation $x = (1-x)^m$ in $(0, 1)$. Table 2 gives the values of x for the first few values of m.

Lemma 2 *For any $\frac{1}{2} < \rho < 1$ constant, there exists an absolute positive constant C such that for all $t > 0$, $|\mu_t - xt| \leq C t^\rho$ a.a.s.*

Table 2: Numerical values defined in Lemma 2.

m	x
1	0.5
2	0.382
3	0.3177
4	0.2755
5	0.2451
6	0.2219
7	0.2035

Proof. We claim that the difference $|\mu_t - xt|$ satisfies a recurrence of the form

$$|\mu_{t+1} - x(t+1)| \le |\mu_t - xt| + O\left(\sqrt{\frac{\log t}{t}}\right).$$

This can be proved by induction on t. By definition $X_1 = 1$, hence $|\mu_1 - x| = 1 - x$. We also have $|\mu_{t+1} - x(t+1)| = |\mu_t - xt + \mathbf{E}\,[(1 - \frac{X_t}{t})^m] - (1-x)^m|$. The difference $\mathbf{E}\,[(1 - \frac{X_t}{t})^m] - (1-x)^m$ can be rewritten as $-\frac{m}{t}(\mu_t - xt) + \{\mathbf{E}\,[(1 - \frac{X_t}{t})^m] - 1 + m\frac{\mu_t}{t} - (1-x)^m + 1 - mx\}$. Hence

$$|\mu_{t+1} - x(t+1)| = |(1 - \tfrac{m}{t})(\mu_t - xt) + \{\mathbf{E}\,[(1 - \tfrac{X_t}{t})^m] - 1 + m\tfrac{\mu_t}{t} - (1-x)^m + 1 - mx\}|.$$

To complete the proof notice that, by Lemma 1,

$$\mathbf{E}\,[(1 - \tfrac{X_t}{t})^m] - 1 + m\tfrac{\mu_t}{t} = (1 - \tfrac{\mu_t}{t})^m - 1 + m\tfrac{\mu_t}{t} + O(\sqrt{\tfrac{\log t}{t}}).$$

and the function $f(z) = (1-z)^m - 1 + mz$ satisfies $|f(z_1) - f(z_2)| \le m|z_1 - z_2|$, for $z_1, z_2 \in [0,1]$.
□

The following Theorem is a direct consequence of Lemma 2 and the concentration result mentioned in the previous section.

Theorem 3.1 $X_t \sim xt$ *a.a.s.*

4 Improved approximations in the random graph process

Although Algorithm 1 is quite simple, it seems difficult to beat, as a glance at $\alpha^R_{\rm up}$ in Table 1 and 2 shows. This is especially true for $m = 1$ where no improvement could be obtained. For larger values of m, a better way of finding small dominating sets is obtained by occasionally allowing vertices to be dropped from $\mathcal{S}$. It is convenient to classify the vertices in the dominating set as *permanent* (set $\mathcal{P}$) and *replaceable* (set $\mathcal{R}$). Thus $\mathcal{S} = \mathcal{P} \cup \mathcal{R}$. Let P_t and R_t denote the sizes of such sets at time t (set $P_1 = 0$ and $R_1 = 1$).

Algorithm 2. Before the first step of the algorithm the graph consists of a single vertex v_0 and $\mathcal{R} = \{v_0\}$. After v_t is created and connected to m neighbours, if $\Gamma(v_t) \cap \mathcal{P} \neq \emptyset$ then v_t is moved to $V \setminus \mathcal{S}$. Otherwise v_t is added to $\mathcal{R}$ if $\Gamma(v_t) \cap \mathcal{R} = \emptyset$, otherwise v_t is added to $\mathcal{P}$ and all vertices in $\Gamma(v_t) \cap \mathcal{R}$ are moved to $V \setminus \mathcal{S}$.

The expectations $\pi_t = \mathbf{E}\,(P_t)$ and $\rho_t = \mathbf{E}\,(R_t)$ satisfy:

$$\begin{aligned}\pi_{t+1} &= \pi_t + \mathbf{E}\,[(1 - \tfrac{P_t}{t})^m] - \mathbf{E}\,[(1 - \tfrac{P_t}{t} - \tfrac{R_t}{t})^m],\\ \rho_{t+1} &= \rho_t + \mathbf{E}\,[(1 - \tfrac{R_t}{t} - \tfrac{P_t}{t})^m] - m\mathbf{E}\,[\tfrac{R_t}{t}(1 - \tfrac{P_t}{t})^{m-1}].\end{aligned}$$

Define $\alpha^R_{\rm up}$ as $p + r$, where $p = p(m)$ and $r = r(m)$ satisfy

$$\begin{aligned} r &= \tfrac{(1-p)^m - p}{1 + m(1-p)^{m-1}},\\ p &= (1-p)^m - (1-p-r)^m.\end{aligned}$$

Lemma 3 *For any $\frac{1}{2} < \rho < 1$ constant, there exist absolute positive constants C_1 and C_2 such that for all $t > 0$, $|\pi_t - pt| \le C_1 t^\rho$ and $|\rho_t - rt| \le C_2 t^\rho$ a.a.s.*

Proof. The proof is, essentially, a generalisation of that of Lemma 2. We present the argument in some details for π_t. At the inductive step:

$$|\pi_{t+1} - p(t+1)| = |\pi_t - pt + \mathbf{E}\,[(1 - \tfrac{P_t}{t})^m] - (1-p)^m - \{\mathbf{E}\,[(1 - \tfrac{P_t}{t} + \tfrac{R_t}{t})^m] - (1-p-r)^m\}|.$$

The proof is completed by decomposing the differences $\mathbf{E}\,[(1 - \frac{P_t}{t})^m] - (1-p)^m$ and $\mathbf{E}\,[(1 - \frac{P_t}{t} - \frac{R_t}{t})^m] - (1-p-r)^m$ in parts that are proportional to either $P_t - \pi_t$ or $\pi_t - pt$.

The result about ρ_t is proved similarly after noticing that r satisfies:

$$r = (1-p-r)^m - mr(1-p)^{m-1}.$$

□

Let X_t^2 denote the size of the dominating set returned by Algorithm 2 when run on a random graph process until time t.

Theorem 4.1 $X_t^2 \sim (p+r)t$, *a.a.s.*

Proof. Theorem 2.1 implies that the sum $p+r$ is a.a.s. very close to $\frac{|\mathcal{S}|}{t} = \frac{P_t}{t} + \frac{R_t}{t}$. The result follows. □

The values of $p+r$ for $m \le 7$ are reported in the column labelled α_{up}^R of Table 1.

5 Improved approximations in the pure copy process

Algorithm 1, described in Section 3, can be analysed in the copy model as well. The expected change in the variable X_t can be computed by keeping track of the total degree of the dominating set, D_t. In particular the following relationships hold

$$\begin{aligned}\mathbf{E}\,(X_{t+1}) &= \mathbf{E}\,(X_t) + \mathbf{E}\,[(1 - \tfrac{D_t}{2mt})^m],\\ \mathbf{E}\,(D_{t+1}) &= (1 + \tfrac{1}{2t})\mathbf{E}\,(D_t) + m\mathbf{E}\,[(1 - \tfrac{D_t}{2mt})^m].\end{aligned}$$

Not surprisingly an analysis similar to the one described in the previous sections implies that such algorithm returns dominating sets of size xt in $G_t^{C,m}$. However, in the copy model we can improve on this by pushing high degree vertices in the dominating set. The natural way to accomplish this would be, for any newly generated uncovered vertex, to select a neighbour of maximum degree and add it to $\mathcal{S}$. Unfortunately such algorithm is not easy to analyse because in the graph processes that we consider there may be few vertices of rather large degree (this is a consequence of the power law distribution of vertex degrees [3, 8]). However a draconian version of this heuristic can be analysed. The following algorithm takes as input an additional integer parameter $k > 0$.

Algorithm 3. Before the first step of the algorithm the graph consists of a single isolated vertex v_0 and $\mathcal{S} = \{v_0\}$. After v_t is created and connected to m neighbours, let Z be the set of all neighbours of v_t in $V \setminus \mathcal{S}$ of degree $km+1$. If $Z \neq \emptyset$ then all vertices in Z are added to $\mathcal{S}$. Otherwise if v_t

is not dominated by some element of $\mathcal{S}$ then a vertex of maximum degree in $\Gamma(v_t)$ is added to the dominating set.

Notice that after each v_t is generated and connected to $G_{t-1}^{C,m}$, all vertices whose degree has become larger than km are moved inside $\mathcal{S}$. The analysis of the evolution of $|\mathcal{S}|$ is based again on the definition of a random process that describes the algorithm dynamics and on the proof that such process behaves in a predictable way for large t.

Let $n = (k-1)m+1$. For each $i \in \{0, \dots, n-1\}$ and $t > 0$, define $Y_t^i = |V_{m+i} \setminus \mathcal{S}|$ in $G_{t-1}^{C,m}$ (V_i is the set of vertices of degree i) before v_t is added to the graph. Let Y_t^n denote the total degree inside $\mathcal{S}$ (i.e. $Y_t^n = \sum_{v \in \mathcal{S}} |\Gamma(v)|$) and X_t the size of the dominating set before v_t is added to the graph. The state of the system, at each step t, is modelled by the (random) vector $(Y_t^0, \dots, Y_t^n, X_t)$. Notice that, for each $t > 0$, the variation in each of the variables is at most m. Also, $Y_t^n + \sum_{i=0}^{(k-1)m} (m+i) Y_t^i = 2mt$, and, at each step $t \geq 1$, when v_t is created the probability that it hits a vertex of degree $m+i$, for $i \in \{0, \dots, (k-1)m\}$ (resp. the dominating set) in any one of the m trials available to it is approximately (omitting $o(1)$ factors, for t large) equal to $P_i = \frac{((m+i-1)(1-\delta_{i,n})+1)Y_t^i}{2mt}$ (where $\delta_{i,n} = 1$ if $i = n$ and zero otherwise).

For $d \in \{0, \dots, n-1\}$ (resp. $d = n$), let E_d denote the event "v_t missed $\mathcal{S}$ and the maximum degree in $\Gamma(v_t)$ is $m+d$" (resp. "v_t did not miss $\mathcal{S}$"). The expected change to Y_t^i, conditioned to the process history up to time t can be computed by further conditioning on the family of events $(E_d)_{d \in \{0, \dots, n\}}$. We can write such quantity as

$$\sum_{d=0}^{n} \mathbf{E}\ (Y_{t+1}^i - Y_t^i \mid E_d) \mathbf{Pr}[E_d].$$

For t large, the probability in the expression above is approximately $\chi_d = (S_0^d)^m - (S_0^{d-1})^m$ (notation S_a^b stands for $P_a + \ldots + P_b$, with $S_a^b \equiv 0$ if $a > b$). Furthermore, we can approximate $\mathbf{E}\ (Y_{t+1}^i - Y_t^i \mid E_d)$ by the expression:

$$\sum C(h_0, \dots, h_d, 0, \dots, 0) \frac{m!}{h_0!\ h_1!\ \dots\ h_d!} \prod_{i=0}^{d} P_i^{h_i} \frac{1}{\chi_d}$$

where C has $n+1$ arguments, the righmost $n-d$ of which are zero, the sum is over all possible ordered tuples of values $h_0, \dots, h_d$ such that $\sum_{i=0}^{d} h_i = m$ and $h_d > 0$, and $C(h_0, \dots, h_d, 0, \dots, 0)$ contains:

- a term for the addition of v_t to $G_{t-1}^{C,m}$;
- a term $\phi_{i,d}$ for the change to Y_t^i due to the handling of the chosen vertex of maximum degree $m+d$ in $\Gamma(v_t)$ (for $d = n$ this is just one of the vertices hitting $\mathcal{S}$), and
- a term $\psi_{i,s}$ for the change to Y_t^i due to the handling of a vertex accounted for by Y_t^s in $\Gamma(v_t)$, for $s \leq d$.

The first of these is simply $\delta_{i,0}$. We also have

$$\phi_{i,d} = \delta_{d,n} \times \delta_{i,d} + (1 - \delta_{d,n}) \times \{(m+d+1)\delta_{i,n} - \delta_{i,d}\}$$

(if $d = n$ (i.e. if v_t hits the dominating set) then one is added to Y_t^n, otherwise Y_t^d is decreased and $m+d+1$ units are added to Y_t^n), and

$$\psi_{i,s} = \delta_{s,n} \times \delta_{i,s} + (1 - \delta_{s,n}) \times \{((m+s)\delta_{s,n-1} + 1)\delta_{i,s+1} - \delta_{i,s}\}$$

(if $s = n$ then Y_t^n is increased by one, if $s = n-1$ the newly created vertex of degree n must be added to $\mathcal{S}$, and finally, in any other case the vertex that has been hit is moved from V_{m+s} to V_{m+s+1}). Therefore

$$C(h_0, \ldots, h_d, 0, \ldots, 0) = \delta_{i,0} + \phi_{i,d} + (h_d - 1)\psi_{i,d} + \sum_{s=0}^{d-1} h_s \psi_{i,s}.$$

Doing all the sums, for each $i \in \{0, \ldots, n\}$, the expected change to Y_t^i is approximately equal to

$$\delta_{i,0} + \sum_{d=0}^{n} \left\{ \chi_d \phi_{i,d} + \psi_{i,d} \left[m P_d (S_0^d)^{m-1} - \chi_d \right] + \frac{m[\chi_d - P_d (S_0^d)^{m-1}]}{S_0^{d-1}} \sum_{s=0}^{d-1} \psi_{i,s} P_s \right\}.$$

Also $\mathbf{E}\ (X_{t+1} - X_t)$ is approximately equal to

$$\frac{km\mathbf{E}\ (Y_t^{n-1})}{2t} + \mathbf{E}\ \left[\left(1 - \frac{Y_t^n}{2mt} - \frac{kY_t^{n-1}}{2t} \right)^m \right].$$

For each positive integer m, let x be the value$kmy^{n-1} + (1 - \frac{y^n}{2m} - \frac{ky^{n-1}}{2})^m$ where y^i, for $i \in \{0, \ldots, n\}$ satisfy $y^i = \left[\mathbf{E}\ (Y_{t+1}^i) - \mathbf{E}\ (Y_t^i) \right] |_{Y^i = y^i t, i \in \{0, \ldots, n\}}$ and the parameter k can be chosen arbitrarily (larger values of k give slightly smaller values of x). The main result about Algorithm 3 is the following theorem whose proof is a consequence of the argument above and Theorem 2.1.

Theorem 5.1 *$X_t \sim xt$, a.a.s.*

For each $m \leq 7$ values of $k \leq 20$ give the values of x reported in Table 1 in the column labelled $\alpha_{\rm up}^C$.

6 Generalisations

The algorithms presented in the previous sections generalise naturally to larger values of h. We present here the generalisation of Algorithm 2 for finding an h-dominating set in the random graph process and of Algorithm 3 for the pure copy process. We then briefly sketch the analysis of the first heuristic, and comment on the numerical results presented in Table 3.

Algorithm 4. A vertex in the h-dominating set can be either *permanent* (set $\mathcal{P}$) as it will never be removed from the set or *i-removable* (set $\mathcal{R}_i$) meaning that it is dominated $i \in \{0, \ldots, h-1\}$ times by other permanent vertices.

Before the first step of the algorithm the graph consists of a single vertex v_0 and $\mathcal{R}_0 = \{v_0\}$. After v_t is created and connected to m neighbours, v_t is added to $\mathcal{R}_0$ if $\Gamma(v_t) \cap (\mathcal{P} \cup \bigcup_i \mathcal{R}_i) = \emptyset$, otherwise if $\Gamma(v_t) \cap \mathcal{P} = \emptyset$, v_t is added to $\mathcal{P}$, any vertex in $\Gamma(v_t) \cap \mathcal{R}_i$, for $i < h-1$ is moved to $\mathcal{R}_{i+1}$ and any vertex in $\Gamma(v_t) \cap \mathcal{R}_{h-1}$ is moved to $V \setminus \mathcal{S}$.

The process in Algorithm 4 can be modelled by h sequences of random variables: R_t^i for $i \in \{0, \ldots, h-1\}$ with $R_t^i = |\mathcal{R}_i|$ in the graph $G_{t-1}^{R,m}$ just before v_t is added to it, and $P_t = |\mathcal{P}|$. The expected changes in these variables satisfy

$$\begin{aligned}
\mathbf{E}\ (P_{t+1}) &= \mathbf{E}\ (P_t) + \mathbf{E}\ [(1 - \tfrac{P_t}{t})^m] - \mathbf{E}\ [(1 - \tfrac{P_t}{t} - \textstyle\sum_i \tfrac{R_t^i}{t})^m] \\
\mathbf{E}\ (R_{t+1}^i) &= \mathbf{E}\ (R_t^i) + \mathbf{E}\ [(1 - \tfrac{P_t}{t} - \textstyle\sum_i \tfrac{R_t^i}{t})^m] \delta_{i,0} + m\mathbf{E}\ [(\tfrac{R_t^{i-1}}{t}(1 - \delta_{i,0}) - \tfrac{R_t^i}{t})(1 - \tfrac{P_t}{t})^{m-1}]
\end{aligned}$$

Table 3: Upper bounds on γ_h/t, for $h > 1$.

m	$h=2$	$h=3$	$h=4$	$h=5$
2	0.4484			
3	0.368	0.3836		
4	0.3162	0.3307	0.332	
5	0.2795	0.2929	0.2944	0.2931
6	0.2517	0.2641	0.2658	0.2647
7	0.2298	0.2413	0.2431	0.2422

Algorithm 4.

m	$h=2$	$h=3$	$h=4$	$h=5$
2	0.5			
3	0.4075	0.6		
4	0.3359	0.4852	0.6663	
5	0.282	0.4073	0.5423	0.7036
6	0.2428	0.3523	0.4613	0.5862
7	0.2132	0.3066	0.4035	0.5056

Algorithm 5.

therefore, we have $P_t \sim pt$ where p satisfies:

$$p = (1-p)^m - \{1 - [(1-p)^m - p][1 - (\tfrac{m(1-p)^{m-1}}{1+m(1-p)^{m-1}})^h] - p\}^m$$

and $R_t^i \sim r^i t$ where $r^0 = \frac{(1-p)^m - p}{1+m(1-p)^{m-1}}$, and $r^i = \frac{m(1-p)^{m-1}}{1+m(1-p)^{m-1}} r^{i-1}$. The values reported in Table 3 below are given by $p + \sum_{i=0}^{h-1} r^i$.

Algorithm 5. Before the first step of the algorithm the graph consists of a single vertex v_0 and $\mathcal{S} = \{v_0\}$. After v_t is created and connected to m neighbours, the set Z of all newly generated vertices of degree more than km are added to $\mathcal{S}$. If v_t is dominated $h-x$ times by elements of $\mathcal{S}$ then the x vertices of highest degree in $\Gamma(v_t) \setminus Z$ are added to the dominating set.

7 Tightness of the algorithmic results

The simplest way of finding lower bounds for the minimal size $\gamma_h(G_t)$ of an h-dominating set in G_t is based on the following result.

Lemma 4 *Let V_i be the set of vertices of degree i. Let i_0 is the largest index i for which $\sum_{j>i} j|V_j| \geq d$. Let $\mathcal{S}$ be an h-dominating set in a graph $G = (V, E)$. If the total degree of the vertices in $\mathcal{S}$ is at least d, then $|\mathcal{S}| \geq \sum_{i>i_0} |V_i|$.*

Proof. If $\Delta = \max_{v \in V} |\Gamma(v)|$, then the set $V_{i_0} \cup \ldots \cup V_\Delta$ is the smallest set of vertices in G with total degree at least d (any other vertex in G would have smaller degree and therefore it would contribute less to the total degree). $\square$

The total degree of an h-dominating set in $G_t^{M,m}$ must be at least $h(t - |\mathcal{S}|) \geq ht(1 - \alpha_{\text{up}}^M)$. Hence a lower bound on the size of any dominating set in a web-graph is obtained by using information on the proportional degree sequence.

For scale-free graphs Bollobas *et al.* [5] (see Cooper [8] for the equivalent preferential attachment model) proved that $|V_i| = tn_i + O(\sqrt{t \log t})$ a.a.s. for any $i \geq m$, where

$$n_i = \frac{2m(m+1)}{i(i+1)(i+2)}.$$

The same papers (eg. [5] p. 288) gives results for $G_t^{R,m}$. In the uar graph process, for any $i \geq m$,

$$n_i = \frac{1}{m+1}\left(\frac{m}{m+1}\right)^{i-m}.$$

Again it is possible to prove that $|V_i|$ is concentrated around $n_i t$. Bounds for $h > 1$ are given in the table below.

Uar graph process

m	$h = 2$	$h = 3$	$h = 4$	$h = 5$
2	0.1317			
3	0.1001	0.178		
4	0.0687	0.1342	0.1678	
5	0.0649	0.0935	0.1346	0.1938
6	0.0535	0.0849	0.1155	0.1573
7	0.0406	0.0692	0.1033	0.1349

Preferential attachment graph process

m	$h = 2$	$h = 3$	$h = 4$	$h = 5$
2	0.0545			
3	0.0316	0.0351		
4	0.023	0.0333	0.0246	
5	0.0183	0.0284	0.0302	0.0183
6	0.0141	0.0233	0.0299	0.0269
7	0.0113	0.0203	0.0271	0.0283

In the case of dominating sets ($h = 1$) we can find stronger lower bounds for the minimal size γ_1 of a dominating set. The argument is based on the first moment method by proving that of all sets of a given size s, the set of vertices $\{1, ..., s\}$ is the most likely to dominate. For $m \geq 2$ the lower bounds in Table 1 are those established in this way. The proofs are given in Sections 7.1.1, 7.1.2 below for the uar and preferential attachment processes respectively. For $m = 1$ the lower bounds in Table 1 come from a precise analysis of the tree structure of the graph process; further detail is given in Section 8.

7.1 Existential lower bounds for the size of dominating sets

7.1.1 Lower bounds for the uar process

Lemma 5 *There exists a function $f(x, y) : \mathbb{R} \to \mathbb{R}$, such that, for each positive integer m, there exists a positive real number $d = d(m)$, defined by $f(m, d) = 1$, such that a.a.s. there is no dominating set in $G_t^{R,m}$ of size at most dt.*

The values of d for the first few values of m are given in the following table:

m	*1*	*2*	*3*	*4*	*5*	*6*	*7*
$d(m)$	*0.26894*	*0.18683*	*0.14881*	*0.12591*	*0.11027*	*0.09876*	*0.08986*

Proof. In the proof we assume that the t vertices in $G_t^{R,m}$ are labelled $1, 2, \ldots, t$. The argument is as follows:
(i) Let $\mathcal{S}^* = [s] = \{1, ..., s\}$, let $|\mathcal{S}| = s$, then

$$\mathbf{Pr}(\mathcal{S} \text{ dom. } \overline{\mathcal{S}}) \leq \mathbf{Pr}(\mathcal{S}^* \text{ dom. } \overline{\mathcal{S}}^*).$$

(ii) For $s \leq dt$, there is a constant $c < 1$ such that the expected number of dominating sets of size s is at most

$$\binom{t}{s}\mathbf{Pr}(\mathcal{S}^* \text{ dom. } \overline{\mathcal{S}}^*) = O(c^t).$$

The first moment method implies that a.a.s. there is no dominating set $\mathcal{S}$ of size at most dt.

Proof of (i). We regard the m out-edges $(e_1, ..., e_m)$ of any vertex as labelled and ordered so that all t vertex graphs G have the same probability in the uar graph process ie. $\mathbf{Pr}(G) = 1/((t-1)!)^m$. (Vertex v_0 has m loops).

Given a partition $(\mathcal{S}, \overline{\mathcal{S}})$ of $[t]$ with $|\mathcal{S}| = s$, let u be the smallest element $u = v_i \in \overline{\mathcal{S}}$ such that the next vertex $v = v_{i+1} \in \mathcal{S}$ in the natural ordering $1, 2, \ldots, t$. If u does not exist then $\mathcal{S} = \mathcal{S}^*$.

We swap u, v to give the partition $(\mathcal{S}', \overline{\mathcal{S}'})$ where $\mathcal{S}' = \mathcal{S} - v + u, \overline{\mathcal{S}'} = \overline{\mathcal{S}} - u + v$. We prove that

$$\mathbf{Pr}(\mathcal{S} \text{ dom. } \overline{\mathcal{S}}) \leq \mathbf{Pr}(\mathcal{S}' \text{ dom. } \overline{\mathcal{S}'}).$$

We do this by showing that for every graph $G \in \{\mathcal{S} \text{ dom. } \overline{\mathcal{S}}\}$ for which the swap destroys the dominance property, ie. $G \notin \{\mathcal{S}' \text{ dom. } \overline{\mathcal{S}}'\}$ (G is good but goes bad on swapping) there is a unique bad graph $H = H(G)$ ($H \notin \{\mathcal{S} \text{ dom. } \overline{\mathcal{S}}\}$) which becomes good on swapping ($H \in \{\mathcal{S}' \text{ dom. } \overline{\mathcal{S}}'\}$). Thus the set of good graphs is of non-decreasing size (and hence probability) on swapping.

More formally, let $\boldsymbol{G}(t) = G_t^{R,m}$ denote the space of uar graph processes after the t-th vertex has been added. Let $\mathcal{S}$ be the distinguished set. Let GG denote the subset of $\boldsymbol{G}$ which is good for both $\mathcal{S}$ and $\mathcal{S}'$ (ie. $\mathsf{GG} = \{G : \mathcal{S} \text{ dom. } \overline{\mathcal{S}}\} \cap \{G : \mathcal{S}' \text{ dom. } \overline{\mathcal{S}}'\}$) and similarly $\mathsf{GB}, \mathsf{BG}, \mathsf{GB}$, so that GB is the set of graphs which are good for S but not $\mathcal{S}'$ etc. Thus $\mathsf{GG}, \mathsf{GB}, \mathsf{BG}, \mathsf{GB}$ is a partition of $\boldsymbol{G}$. We define (below) a one-one map $\phi : \boldsymbol{G} \to \boldsymbol{G}$ such that $\phi(\mathsf{GB}) \subseteq \mathsf{BG}$. It follows that

$$\begin{aligned}
|\{G : \mathcal{S} \text{ dom. } \overline{\mathcal{S}}\}| &= |\mathsf{GG}| + |\mathsf{GB}| \\
&= |\mathsf{GG}| + |\phi(\mathsf{GB})| \\
&\leq |\mathsf{GG}| + |\mathsf{BG}| = |\{G : \mathcal{S}' \text{ dom. } \overline{\mathcal{S}}'\}|.
\end{aligned}$$

With some abuse of notation, we associate $\text{out}(x)$ the sequence of out-edges of vertex x with the multi-set of terminal vertices, and similarly for $\text{in}(x)$ (the sequence of in-edges of x ordered naturally) with their initial vertices.

Construction of ϕ. Let $[t] = (1, 2, ..., u-1, u, v, v+1, ..., t)$ where u, v are the swap pair.Consider the graph G.
Let $i(x) = \text{in}(x) \cap [v+1, ..., n]$ and $o(x) = \text{out}(x) = \text{out}(x) \cap [1, ..., v]$.
Let $i'(u) = i(v),\ i'(v) = i(u)$.

Case that there is no edge (v, u). Let $o'(u) = o(v),\ o'(v) = o(u)$.
Construct $\phi(G) = H(G)$ from G by replacing $i(x), o(x)$ by $i'(x), o'(x),\ x = u, v$. The graph H is uniquely associated with G, as we can reconstruct G from H.

Case that there are edges (v, u). Let the (v, u) edges be $\{e_k, k \in K\}$ where K indexes their occurrences in $(e_1, ..., e_m)$. Let $\{f_k\}$ be the out-edges of u with the same index set.
Let $o'(u) = o(v) - \{e_k\} + \{f_k\}$, let $o'(v) = o(u) + \{e_k\} - \{f_k\}$.
Thus v keeps its out edges to u, and u keeps its out edges with the same labels.

If there is no (v, u) edge and $G \in \mathsf{GB}$ then H was bad but becomes good (ie. $H \in \mathsf{BG}$. For, in H, u (now in the dom. set) has the v edges from G, and v (no longer in the dom. set) got the u edges from G.

If there is a (v,u) edge and $G \in \mathsf{GB}$, then u dominates v after the swap, just as v dominated u before. The in edges have been swapped over to preserve any dominance which v made with them (now made by u). The edge set $\{f_k\}$ was not necessary for G to be good, as v dominated u and u did not dominate anybody. If v dominated anybody else with $o(v) - \{e_k\}$ then u still does it with $o'(u) = o(v) - \{e_k\} + \{f_k\}$. Thus H is good.

Proof of (ii). Let $S^* = [s]$. We have that

$$\mathbf{Pr}(\exists \text{ dominating set } S, |S| = s) \leq \binom{t}{s} \mathbf{Pr}(\mathcal{S}^* \text{ dom. } \overline{\mathcal{S}}^*).$$

Let $s = dt$ then

$$\tbinom{t}{s} \leq \left(\tfrac{1}{d^d(1-d)^{(1-d)}}\right)^t,$$

and

$$\begin{array}{rcl} \mathbf{Pr}(\mathcal{S}^* \text{ dom. } \overline{\mathcal{S}}^*) & = & \prod_{v=s}^{t-1} \left(1 - \left(1 - \frac{s}{v}\right)^m\right) \\ & = & \exp \sum_s^{t-1} \log\left(1 - \left(1 - \frac{s}{v}\right)^m\right) \\ & = & \exp(1 + o(1)) \int_{s+1}^t \log\left(1 - \left(1 - \frac{s}{v}\right)^m\right) dv \\ & = & \exp(1 + o(1))\ dt \int_d^1 x^{-2} \log\left(1 - (1-x)^m\right) dx. \end{array}$$

To obtain (m, d) as given we compute the largest d such that

$$\left(\tfrac{1}{d^d(1-d)^{(1-d)}}\right) \exp d \int_d^1 x^{-2} \log\left(1 - (1-x)^m\right) dx < 1.$$

$\square$

7.1.2 Lower bounds for the preferential attachment process

Lemma 6 *There exists a function $f(x,y) : \mathbb{R} \to \mathbb{R}$, such that, for each positive integer m, there exists a positive real number $d = d(m)$, defined by $f(m,d) = 1$, such that a.a.s. there is no dominating set in $G_t^{C,m}$ of size at most dt.*

The values of d for the first few values of m are given in the following table:

m	1	2	3	4	5	6	7
$d(m)$	0.17379	0.10098	0.07099	0.05443	0.04391	0.03663	0.03130

Proof. The argument is the same as the one used for the uar graph process ie. we prove that for any set S of size s then $\mathbf{Pr}(S \text{ dom. } \overline{S}) \leq \mathbf{Pr}(S^* \text{ dom. } \overline{S}^*)$ where $S^* = [s] = \{1, ..., s\}$. As before, we do this by moving the vertices of the dominating set S downward. Now however, in the case where there is a (v,u) edge, swapping edges between the vertices u, v can alter the probability of the graphs, ie. $\mathbf{Pr}(G) \neq \mathbf{Pr}(\phi(G))$, and some care is needed to overcome this problem. In order to get a precise value for $\mathbf{Pr}(G)$ we work in the scale-free model of Bollobas and Riordan. A full description of the model can be found in [5]. The scale-free model differs slightly from our preferential attachment (pure copy) model in that loops are allowed. We consider the case $m = 1$, as the general case is easily derived from this. We have

$$\begin{aligned} \mathbf{Pr}(v_t \text{ chooses } u) &= \frac{\deg(u, t-1)}{2t-1} \\ \mathbf{Pr}(v_t \text{ chooses } v_t) &= \frac{1}{2t-1}, \end{aligned}$$

where $\deg(v, t)$ is the degree of vertex v at the end of step t. For fixed $t = n$ the scale-free model is formulated as a configuration model on the edge endpoints, labelled $1, ..., 2n$. A pairing $\mathcal{P}$ of the endpoints describes the set of edges, and an edge is completed at step k in the linear order $1, ..., 2n$ if the larger point in the pairing is k. Given a point k, the left partial pairing $\mathcal{L}$ is the partial pairing on $1, ..., k$, the right partial pairing $\mathcal{R}$ is the partial pairing on $k+1, ..., 2n$ both of which may have some completed edges. A further matching of unpaired points between the partial pairings completes $\mathcal{P}$. See [5] for more detail.

As usual let $i'(x), x = u, v$ be the (modified) in-degree (or associated set of in-edges) of vertex x. Thus $i'(x)$ is the in-degree except that $i'(u)$ does not include the edge (v, u) if it exists. Let $i'(u) = a_1, i'(v) = a_2$. Thus the in-degree of u is $a_1 + 1$ if (v, u) exists and a_1 if not. Let $G(a_1, a_2)$ be a graph with these in-degrees at u, v respectively. To save on notation we let a_i be the multi-set of distinguished in-edges as well as the set size. Writing $G(a_2, a_1)$ means the in-edges of u, v other than (v, u) have been swapped, and $\phi(G(a_1, a_2))$ means the out-edges other than (v, u) and its corresponding labelled edge(s) at u have also been swapped. We use $a_1 + a_2$ for set union, preserving multiplicity. The meaning will always be clear from the context.

Let $u = x_k$ be the k-th vertex, and so $v = x_{k+1}$. Suppose that x_k the k-th right endpoint has label $2k + s + a_1 + 1$ and the $(k+1)$-th right endpoint has label $2(k+1) + s + a_1 + a_2$ in the linear order. If there is no edge (v, u) let the $(k-1)$-th right endpoint have label $2(k-1) + s + 2$ and if there is a (v, u) edge let it have label $2(k-1) + s + 1$. Thus vertex u has in-degree a_1 in the first case and $a_1 + 1$ in the second case. In either case vertex v has in-degree a_2 and there are s un-paired labels in the left partial pairing of vertex x_{k-1}.

For t even, let $\Phi(t)$ denote the number of pairings of t points, and thus $\Phi(t) = t!/((t/2)!2^{t/2})$. We can write down the probability of the graph $G(a_1, a_2)$ in the two cases as follows: $\mathbf{Pr}(G(a_1, a_2)) = F_L F_{R,L}/\Phi(2n)$ where F_L is the number of left partial pairings of $2(k+1) + s + a_1 + a_2$ labels and $F_{R,L}$ is the is the number of completions with the right partial pairing on $2n - (2(k+1) + s + a_1 + a_2)$ labels. In either case

$$F_{R,L} = \binom{2n - (2(k+1) + s + a_1 + a_2)}{s + a_1 + a_2} (s + a_1 + a_2)!\Phi(2(n - ((k+1) + s + a_1 + a_2))). \tag{1}$$

If there is no (v, u) edge then

$$F_L = (2k + s - 1)(2k + s - 2)(2k + s - 3)\binom{2k + s - 4}{s}\Phi(2(k-2)), \tag{2}$$

and if there is a (v, u) edge then

$$F_L = (2k + s - 2)(2k + s - 3)(a_1 + 1)\binom{2k + s - 4}{s}\Phi(2(k-2)). \tag{3}$$

The first two terms on the right are for $x_{k-1}, x_k = u$ completing a pairing and the third is for $x_{k+1} = v$. We then choose s points for the left-right pairing and pair the remaining points on the left.

From the above expressions we see that if there is no edge (v, u) in G then expressions (1), (2) are unchanged on swapping and $\mathbf{Pr}(G) = \mathbf{Pr}(\phi(G))$. If there is an edge (v, u) then

$$\begin{aligned}
\mathbf{Pr}(G(a_1, a_2)) &= c(a_1 + 1) &(4)\\
\mathbf{Pr}(\phi(G(a_1, a_2))) &= \mathbf{Pr}(G(a_2, a_1)) = c(a_2 + 1) &(5)
\end{aligned}$$

where c is given above. We see that if $a_2 < a_1$ we cannot use the uar proof without modification.

Recall that GB is the set of graphs which are good for S (S dominates $\overline{S}$) but bad for $S' = S + u - v$ (S' does not dominate $\overline{S}'$). Our approach is to partition GB into sets C which have the property $\mathbf{Pr}(C) \le \mathbf{Pr}(\phi(C))$. In preparation for this we note:

Let $\sigma = \lceil n/\log n \rceil$, let

$$\begin{aligned}
B_I &= \{G : \exists i \ge \sigma \text{ such that } G \text{ has at least 2 edges } (x_{i+1}, x_i)\}\\
B_{II} &= \{G : \exists j > i \ge \sigma \text{ such that } G \text{ has an edge } (x_{i+1}, x_i) \text{ and at least 2 edges } (x_j, x_{i+1})\}.
\end{aligned}$$

Based on the almost sure maximum degree of vertex x at step n of $(n/x)^{1/2} c \log^3 n$ (see eg. [5], [8]) for all $x \ge \sigma$ we see that $\mathbf{Pr}(B_I) = O(\log n/n)$ and that $\mathbf{Pr}(B_{II}) = O(\log^6 n/n)$. To take advantage of this, we note that

$$\mathbf{Pr}(S \text{ dom. } \overline{S}) \le \mathbf{Pr}(S \cup [\sigma] \text{ dom. } \overline{S} \setminus [\sigma]).$$

Thus we need only consider swaps on $[\sigma + 1, ..., n]$ as $[\sigma]$ is always in the dominating set.

Let

$$A_u = \{G \in \mathsf{GB} : \exists (v, u) \text{ edge in } G\} \cap \overline{(B_I \cup B_{II})},$$

let

$$B_u = \mathsf{GB} \cap (B_I \cup B_{II}),$$

and let

$$C_u = \{G \in \mathsf{GB} : \text{ no edge } (v, u)\}.$$

The proof for C follows directly from the uar case as $\mathbf{Pr}(G) = \mathbf{Pr}(\phi(G))$. Also $\mathbf{Pr}(\cup_u B_u) = O(\log^6 n/n)$. The proof for A is simplified by the event $\overline{B_I}$ as there is exactly one (v, u) edge to consider.

For $G(a_1, a_2) \in A$ we define the minimal graph $G(\alpha_1, \alpha_2)$ of G as follows: Let $a_2 = \alpha_2 + \beta_2$ where every edge of α_2 is necessary for S to dominate $\overline{S}$ in G. Thus if $e \in \alpha_2$ then $G(a_1, \alpha_2 - e)$ does not satisfy S dominates $\overline{S}$, but $G(a_1, \alpha_2)$ does. It is quite possible that $\alpha_2 = \emptyset$ eg. only $\text{out}(v)$ dominates something or v dominates itself. Let $\alpha_1 = a_1 + \beta_2$. Every $G(a_1, a_2)$ has at least one minimal graph. We form the set

$$\boldsymbol{G}(a_1, a_2) = \{G(\alpha_1 - I, \alpha_2 + I) : I \subseteq \alpha_1\}.$$

Thus $\boldsymbol{G}(a_1, a_2) = \{G(\alpha_1, \alpha_2), ..., G(a_1, a_2), ..., G(\emptyset, \alpha_1 + \alpha_2)\}$. We claim that

(i) Given $G(a_1, a_2) \in A$ then $G(\alpha_1, \alpha_2)$ is unique.

(ii) $\boldsymbol{G}(a_1, a_2) \subseteq A$.

(iii) If $G(a, b), G(a', b')$ are distinct minimal graphs then

$$\boldsymbol{G}(a, b) \cap \boldsymbol{G}(a', b') = \emptyset.$$

(iv) The sets $\boldsymbol{G}(\alpha_1, \alpha_2)$ partition A.

Let $G(a_1, a_2)$ have two distinct minimal graphs $G(\alpha_1, \alpha_2), G(\alpha_1', \alpha_2')$ and thus $\alpha_2 \neq \emptyset$. As α_2 is minimal there is some edge $e = (x, v) \in \alpha_2$ such that v is the only vertex of S dominating x. Thus α_2' must also have an edge $e' = (x, v)$ for v to dominate x. The event B_{II} that (v, u) exists and that distinct parallel edges e, e' to v also exist has been excluded from A, so $\alpha_2' = \alpha_2$.

We next prove that if $G(a_1, a_2) \in A$ then $G(\alpha_1, \alpha_2) \in A$. As $G(a_1, a_2) \in \mathsf{GB}$ there exists a vertex x uniquely dominated by v. Suppose $x > v$ so that $\alpha_2 \neq \emptyset$ and that a_1 does not include an edge from x. Also $\beta_2 = a_2 - \alpha_2$ does not include an edge from x as there are no parallel edges. Thus there is no edge from x in $\alpha_1 = a_1 + \beta_2$, so $S' = S - v + u$ cannot dominate x and $G(\alpha_1, \alpha_2)$ is bad for S'.

The rest of the claims follow in a similar manner. Finally we claim that

$$\mathbf{Pr}(\boldsymbol{G}(\alpha_1, \alpha_2)) \leq \mathbf{Pr}(\phi(\boldsymbol{G}(\alpha_1, \alpha_2))). \tag{6}$$

Note that $\mathbf{Pr}(\phi(G(b_1, b_2))) = \mathbf{Pr}(G(b_2, b_1))$ as the out-edge swaps of u, v are measure preserving (see (1), (2)). It also follows from (1), (2) that if there is no (v, u) edge then $\mathbf{Pr}(G(b_1, b_2)) = \mathbf{Pr}(G(b_2, b_1))$. If there is a (v, u) edge then from (3), (4), (5)

$$\mathbf{Pr}(G(b_1, b_2)) = c(b_1 + 1), \qquad \mathbf{Pr}(G(b_2, b_1)) = c(b_2 + 1).$$

As $\boldsymbol{G}(\alpha_1, \alpha_2) = \{G(\alpha_1, \alpha_2), ..., G(\alpha_1 - I, \alpha_2 + I), ..., G(\emptyset, \alpha_1 + \alpha_2)\}$, we have

$$\begin{aligned} \mathbf{Pr}(\boldsymbol{G}(\alpha_1, \alpha_2)) &= c \sum_{j=0}^{\alpha_1} \binom{\alpha_1}{j} (j+1) \\ &= c \left(2^{\alpha_1} + \alpha_1 2^{\alpha_1 - 1}\right), \end{aligned}$$

whereas $\phi(\boldsymbol{G}(\alpha_1, \alpha_2))$ has the same probability as $\{G(\alpha_2, \alpha_1), ..., G(\alpha_2 + I, \alpha_1 - I), ..., G(\alpha_1 + \alpha_2, \emptyset)\}$, and thus

$$\begin{aligned} \mathbf{Pr}(\phi(\boldsymbol{G}(\alpha_1, \alpha_2))) &= c \sum_{j=0}^{\alpha_1} \binom{\alpha_1}{j} (\alpha_2 + \alpha_1 - j + 1) \\ &= c \left((\alpha_1 + \alpha_2 + 1) 2^{\alpha_1} - \alpha_1 2^{\alpha_1 - 1}\right). \end{aligned}$$

Thus

$$\mathbf{Pr}(\phi(\boldsymbol{G}(\alpha_1, \alpha_2))) - \mathbf{Pr}(\boldsymbol{G}(\alpha_1, \alpha_2)) = c\alpha_2 2^{\alpha_1} \geq 0$$

as required by (6).

Finally we explain the generalization of this proof to scale-free graphs of out-degree $m \geq 2$. The scale-free model treats vertex x as a sequence $(x(1), ..., x(m))$ of sub-vertices of degree 1 arising from m consecutive right endpoints of edges in the linear order $1, ..., 2mn$. Let the (v, u) edge in question be $e_i = (v(i), u(j))$. We swap the edges of $v(i)$ and $u(j)$ except e_i and the corresponding out-edge f_j of $u(j)$. The other sub-vertices are paired cyclically $v(i + k)$ with $u(j + k)$ for swapping.

Let $\boldsymbol{G}_S = \{G_t^m\}$ be the space of scale-free graphs of out-degree m. The space $\boldsymbol{G}_S$ is very close to the space of preferential attachment (pure copy) processes $\boldsymbol{G}_C = \{G_t^{C,m}\}$ described in this paper. In either model, on addition of the out-edges $e_j,\ j = 1, ..., m$ of vertex v_t, the terminal vertex for the edge e_j is chosen by selecting a random end-point of a random edge in a designated set. In the scale-free model all existing edges and half edges are included in the choice. In the preferential attachment (pure copy) model only out-edges of previously added vertices $v_1, ..., v_{t-1}$ are included. Thus in the scale-free model

$$\mathbf{Pr}(e_j \text{ chooses from } \mathrm{out}(v_1, ..., v_{t-1})) = \frac{2m(t-1)}{2m(t-1) + 2(j-1) + 1} \qquad j = 1, ..., m.$$

Thus in the scale-free model the subset C of preferential attachment graphs $\boldsymbol{G}_C$ has probability

$$\mathbf{Pr}(C) \geq \prod_{s=2}^{t} \left(1 - \frac{1}{s-1}\right)^m = \Omega(t^{-m}),$$

and conditional probability of graphs in C is the correct measure for $\boldsymbol{G}_C$. The existential lower bounds (derived below) for the smallest dominating set in the scale-free model have probability of the form

$$\mathbf{Pr}(\text{ there exists a dominating set } S \text{ of size } |S| \leq dt) = O(c^t),$$

for $c < 1$ constant. We conclude the results are valid for the preferential attachment model.

Let $S^* = [dt]$ then

$$\mathbf{Pr}(S^* \text{ dom. } \overline{S^*}) \leq \prod_{v=dn+1}^{n} \left(1 - \left(1 - \frac{\deg(S^*, v-1)}{2m(v-1)}\right)^m\right),$$

and $\deg(S^*, v) = 2mdt(v/dt)^{1/2}(1 + o(1))$ almost surely for $d > 0$ constant (see [5]). Thus

$$\mathbf{Pr}(S^* \text{ dom. } \overline{S^*}) = \exp \sum_{dt+1}^{t} \log(1 - (1 - (dt/v)^{1/2}(1 + o(1)))^m).$$

Let $x = (dt/v)^{1/2}$. Choosing d such that

$$\frac{1}{d^d(1-d)^{1-d}} \exp 2d \int_{\sqrt{d}}^{1} x^{-3} \log(1 - (1-x)^m)\, dx < 1$$

gives the required lower bound. Values of (m, d) are given in the table.

□

8 Trees

For $m = 1$ the graph processes under consideration generate a connected graph without cycles. Such structural property can be exploited to obtain improved lower bounds on γ_1. Without loss of generality, any vertex in such graphs that has at least one neighbour $u \in V_1$ must be part of a minimum size dominating set. The number of such vertices is precisely $t - |V_1| - |I|$ where $|I|$ is the number of vertices that have no neighbour in V_1. The cardinality of I can be estimated in both models via either a martingale argument similar to those used in previous sections or through the technique exploited in [8] to estimate $|V_i|$. The lower bounds in Table 1 for $m = 1$ come from this argument.

We end this Section reporting on some simple empirical results which help put the mathematical analysis performed so far into context. It is well known [7] that minimum size dominating sets can be found efficiently in trees. We implemented Cockayne et al's algorithm and tested its performance. For different values of t, we repeatedly ran the two graph processes up to time t, and then applied the optimal algorithm mentioned above. Table 4 reports the average values we obtained. The least square approximation lines over the full set of data we collected are (coefficients rounded to the sixth decimal place) $y = 0.374509x - 0.214185$ for the random graph case, and $y = 0.294294x + 0.32284$ for the pure copy case. These results indicate that our algorithms are able to get better results for graphs generated according to the pure copy process ($\alpha^C_{\text{up}} = 0.3333$) than for graphs generated by the other process ($\alpha^R_{\text{up}} = 0.5$). We leave the finding of improved algorithms especially in the random graph process or indeed better lower bounds in either models as an interesting open problem of this work.

Table 4: Average values obtained over 1000 experiments for each value of t.

t	$\gamma_1(G_{1,t}^R)/t$	$\gamma_1(G_{1,t}^C)/t$
10000	3745.053	2943.157
20000	7489.3	5887.301
30000	11233.68	8829.288
40000	14980.384	11772.175
50000	18725.448	14714.073
...		
100000	37451.20312	29424.216

References

[1] N. Alon, J. H. Spencer, and P. Erdős. *The Probabilistic Method.* John Wiley & Sons, 1992.

[2] K. Alzoubi, P. J. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proc. 3rd ACM Internat. Symp. on Mobile Ad-hoc Networking & Computing*, pp 157–164, 2002.

[3] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[4] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *JCT*, A 24:296–307, 1978.

[5] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *RSA*, 18:279–290, 2001.

[6] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proc. 9th WWW*, pp 309–320, 2000.

[7] E. Cockayne, S. Goodman, and S. Hedetniemi. A linear algorithm for the domination number of a tree. *IPL*, 4:41–44, 1975.

[8] C. Cooper. The age specific degree distribution of web-graphs. Submitted to Combinatorics Probability and Computing, 2002.

[9] C. Cooper and A. Frieze. A general model of web graphs. *RSA*, 22:311–335, 2003.

[10] W. Duckworth and M. Zito. Sparse hypercube 3-spanners. *DAM*, 103:289–295, 2000.

[11] S. Eidenbenz. Online dominating set and variations on restricted graph classes. Technical Report 380, Department of Computer Science, ETH Zürich, 2002.

[12] U. Feige. A threshold of $\ln n$ for approximating set cover. *JACM*, 45:634–652, 1998.

[13] M. R. Garey and D. S. Johnson. Strong NP-Completeness results: Motivation, examples, and implications. *Journal of the Association for Computing Machinery*, 25(3):499–508, 1978.

[14] F. Harary and T. W. Haynes. Double domination in graphs. *Ars Comb.*, 55:201–213, 2000.

[15] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, editors. *Domination in Graphs: Advanced Topics.* Marcel Dekker, 1998.

[16] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs.* Marcel Dekker, 1998.

[17] G.-H. King and W.-G. Tzeng. On-line algorithms for the dominating set problem. *IPL*, 61:11–14, 1997.

[18] R. Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *IPL*, 89:75–83, 2004.

[19] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. The web as a graph. In *Proc. ACM Symp. on PODS*, pp 1–10, 2000.

[20] M. Levene and R. Wheeldon. Web dynamics. *Software Focus*, 2:31–38, 2001.

[21] C.-S. Liao and G. J. Chang. k-tuple domination in graphs. *IPL*, 87:45–50, 2003.

[22] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Trans. Parallel and Dist. Systems*, 13:14–25, 2002.

[23] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(4):440–442, June 1998.

[24] B. Wieland and A. P. Godbole. On the domination number of a random graph. *Elec. J. Combinat.*, 8:# R37, 2001.

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

[26] M. Zito. Greedy algorithms for minimisation problems in random regular graphs. *Proc 9th ESA*, pp 524–536, LNCS 2161. Springer-Verlag, 2001.