Vertex-pursuit in hierarchical social networks

A. Bonato[1], D. Mitsche[1], and P. Prałat[1]

Department of Mathematics
Ryerson University
Toronto, Canada
abonato@ryerson.ca, dmitsche@ryerson.ca, pralat@ryerson.ca *

Abstract. Hierarchical social networks appear in a variety of contexts, such as the on-line social network Twitter, the social organization of companies, and terrorist networks. We examine a dynamic model for the disruption of information flow in hierarchical social networks by considering the vertex-pursuit game Seepage played in directed acyclic graphs (DAGs). In Seepage, agents attempt to block the movement of an intruder who moves downward from the source node to a sink. We propose a generalized stochastic model for DAGs with given expected total degree sequence. Seepage is analyzed rigorously in stochastic DAGs in both the cases of a regular and power law degree sequence.

1 Introduction

The on-line social network Twitter is a well known example of a complex real-world network with over 300 million users. The topology of Twitter network is highly directed, with each user following another (with no requirement of reciprocity). By focusing on a popular user as a source (such as Lady Gaga or Justin Bieber, each of whom have over 11 million followers [14]), we may view the followers of the user as a certain large-scale *hierarchical social network*. In such networks, users are organized on ranked levels below the source, with links (and as such, information) flowing from the source downwards to sinks. We may view hierarchical social networks as *directed acyclic graphs*, or *DAGs* for short. Hierarchical social networks appear in a wide range of contexts in real-world networks, ranging from terrorist cells to the social organization in companies; see, for example [1, 8, 10, 12, 13].

In hierarchical social networks, information flows downwards from the source to sinks. Disrupting the flow of information may correspond to halting the spread of news or gossip in OSN, or intercepting a message sent in a terrorist network. How do we disrupt this flow of information while minimizing the resources used? We consider a simple model in the form of a vertex-pursuit game called Seepage introduced in [6]. Seepage is motivated by the 1973 eruption of the Eldfell volcano in Iceland. In order to protect the harbour, the inhabitants poured water on the lava in order to solidify it and thus, halt its progress. The game has two players, the *sludge* and a set of *greens*, a DAG with one source (corresponding to the top of the volcano) and many sinks (representing the lake). The players take turns, with the sludge going first by contaminating the top node (source). On subsequent moves the sludge contaminates a non-protected node that is adjacent (that is, downhill) to a contaminated node. The greens, on their turn, choose some non-protected, non-contaminated node to protect. Once protected or contaminated, a node

* The authors gratefully acknowledge support from Mprime, NSERC, and Ryerson.

stays in that state to the end of the game. The sludge wins if some sink is contaminated; the greens win if they erect a cutset of nodes which separates the contaminated nodes from the sinks. The name "seepage" is used because the rate of contamination is slow. The game is related to vertex-pursuit games such as Cops and Robbers (see [3]), although the greens in our case need not move to neighbouring nodes. For an example, see the DAG in Figure 1. (We omit orientations of directed edges in the figure, and assume all edges point from higher nodes to lower ones.)

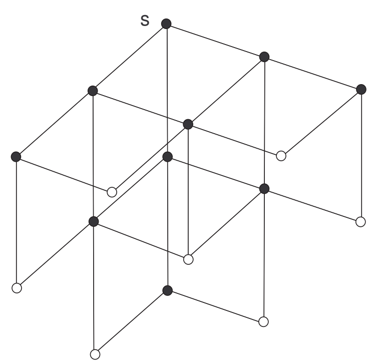

Fig. 1. A DAG where 2 greens are needed to win. The white nodes are the sinks.

Seepage displays some interesting similarities to an approach used in mathematical counterterrorism, where cut sets in partially ordered sets (which are just a special kind of DAG) are used to model the disruption of terrorist cells. As described in Farley [9, 8], the maximal elements of the poset are viewed as the leaders of the terrorist organization, who submit plans down via the edges to the nodes at the bottom (the foot soldiers or minimal nodes). Only one messenger needs to receive the message for the plan to be executed. Farley considered finding minimum-order sets of elements in the poset, which when deleted, disconnect the minimal elements from the maximal one (that is, find a *minimum cut*). We were struck by the similarities in the underlying approaches in [6] and [9, 8]; for example, in Seepage the greens are trying to prevent the sludge from moving to the sinks by blocking nodes. The main difference is that Seepage is "dynamic" (that is, the greens can move, or choose new sets of nodes each time-step), while the min-cut-set approach is "static" (that is, find a cutset in one time-step). Seepage is perhaps a more realistic model of counterterrorism, as the agents do not necessarily act all at once but over time. However, in both approaches deterministic graphs are used.

We note that a stochastic model was presented for so-called *network interdiction* in [11], where the task of the interdictor is to find a set of edges in a weighted network such that the removal of those edges would maximally increase the cost to an evader of traveling on a path through the network. A stochastic model for complex DAGs was given in [4]. For more on models of OSNs and other complex networks, see [2].

Our goal in the present extended abstract is to analyze Seepage and the green number when played on a random DAG as a model of disrupting a given hierarchical social network. We focus on mathematical results, and give a precise formulation of our random DAG model in Section 2. Our model includes as a parameter the total degree distribution of nodes in the DAG. This has some similarities to the $G(\mathbf{w})$ model of random graphs with expected degree sequences (see [5]) or the pairing model (see [16]). We study two cases: regular DAGs (where we would expect each level of the DAG to have nodes with about the same out-degree), and power law DAGs (where the degree distribution is heavy tailed, with many more low degree nodes but a few which have a high degree). Rigorous results are presented for regular DAGs in Theorem 1, and in power law DAGs in Theorem 2. Proofs are largely omitted and will appear in the full version of the paper.

Throughout, G will represent a finite DAG. For background on graph theory, the reader is directed to [7, 15]. Additional background on seepage and other vertex-pursuit games may be found in [3]. We denote the natural numbers (including 0) by $\mathbb{N}$, and the positive integers and real numbers by $\mathbb{N}^+$ and $\mathbb{R}^+$, respectively.

2 Definitions

We now give a formal definition of our vertex-pursuit game. Fix $v \in V(G)$ a node of G. We will call v the *source*. For $i \in \mathbb{N}$ let

$$L_i = L_i(G, v) = \{u \in V(G) : \mathrm{dist}(u, v) = i\},$$

where $\mathrm{dist}(u, v)$ is the distance between u and v in G. In particular, $L_0 = \{v\}$. For a given $j \in \mathbb{N}^+$ and $c \in \mathbb{R}^+$, let $\mathcal{G}(G, v, j, c)$ be the game played on graph G with the source v and the *sinks* L_j. The game proceeds over a sequence of discrete time-steps. Exactly

$$c_t = \lfloor ct \rfloor - \lfloor c(t-1) \rfloor$$

new nodes are protected at time-step t. (In particular, at most ct nodes are protected by time t.) Note that if c is an integer, then exactly c nodes are protected at each time-step, so this is a natural generalization of Seepage. To avoid trivialities, we assume that $L_j \neq \emptyset$.

The *sludge* starts the game on the node $v_1 = v$. The second player, the *greens*, can protect $c_1 = \lfloor c \rfloor$ nodes of $G \setminus \{v\}$. Once nodes are protected they will stay protected to the end of the game. At time $t \geq 2$, the sludge makes the first move by sliding along a directed edge from v_{t-1} to v_t, which is an out-neighbour of v_{t-1}. After that the greens have a chance to protect another c_t nodes. Since the graph is finite and acyclic, the sludge will be forced to stop moving, and so the game will eventually terminate. If he reaches any node of L_j, then the sludge wins; otherwise, the greens win.

If $c = \Delta(G)$ (the maximum out-degree of G), then the game $\mathcal{G}(G, v, j, c)$ can be easily won by the greens by protecting of all neighbours of the source. Therefore, the following graph parameter, *the green number*, is well defined:

$$g_j(G, v) = \inf\{c \in \mathbb{R}^+ : \mathcal{G}(G, v, j, c) \text{ is won by the greens}\}.$$

It is clear that for any $j \in \mathbb{N}^+$ we have $g_{j+1}(G, v) \leq g_j(G, v)$.

2.1 Random DAG model

There are two parameters of the model: $n \in \mathbb{N}^+$ and an infinite sequence

$$\mathbf{w} = (w_1, w_2, \ldots)$$

of non-negative integers. Note that the w_i may be functions of n. The first layer (that is, the source) consists of one node: $L_0 = \{v\}$. The next layers are recursively defined. Suppose that all layers up to and including the layer j are created, and let us label all nodes of those layers. In particular,

$$L_j = \{v_{d_{j-1}+1}, v_{d_{j-1}+2}, \ldots, v_{d_j}\},$$

where $d_j = \sum_{i=0}^{j} |L_i|$. We would like the nodes of L_j to have a total degree with the following distribution $(w_{d_{j-1}+1}, w_{d_{j-1}+2}, \ldots, w_{d_j})$. However, it can happen that some node $v_i \in L_j$ has an in-degree $\deg^-(v_i)$ already larger than w_i, and so there is no hope for the total degree of w_i. If this is not the case, then the requirement can be easily fulfilled. As a result, the desired degree distribution will serve as a lower bound for the distribution we obtain during the process.

Let S be a new set of nodes of cardinality n. All directed edges that are created at this time-step will be from the layer L_j to a random subset of S that will form a new layer L_{j+1}. Each node $v_i \in L_j$ generates $\max\{w_i - \deg^-(v_i), 0\}$ random directed edges from v_i to S. Therefore, we generate

$$e_j = \sum_{v_i \in L_j} \max\{w_i - \deg^-(v_i), 0\}$$

random edges at this time-step. The destination of each edge is chosen uniformly at random from S. All edges are generated independently, and so we perform e_j independent experiments. The set of nodes of S that were chosen at least once forms a new layer L_{j+1}. Note that it can happen that two parallel edges are created during this process. However, for sparse random graphs we are going to investigate in this paper, this is rare and excluding them, by slightly modifying the process, would not affect any of the results.

3 Main results

In this paper, we focus on two specific sequences: regular and power law. We will describe them both and state main results in the next two subsections. We consider asymptotic properties of the model as $n \to \infty$. We say that an event in a probability space holds *asymptotically almost surely* (*a.a.s.*) if its probability tends to one as n goes to infinity.

3.1 Random regular DAGs

We consider a constant sequence; that is, for $i \in \mathbb{N}^+$ we set $w_i = d$, where $d \geq 3$ is a constant. In this case, we refer to the stochastic model as *random d-regular DAGs*.

Since $w_i = d$, observe that $|L_j| \le d(d-1)^{j-1}$ (deterministically) for any j, since at most $d(d-1)^{j-1}$ random edges are generated when L_j is created. We will write g_j for $g_j(G, v)$ since the graph G is understood to be a d-regular random graph, and $L_0 = \{v\} = \{v_1\}$.

Theorem 1. *Let $\omega = \omega(n)$ be any function that grows (arbitrarily slowly) as n tends to infinity. For the random d-regular DAGs, we have the following.*

(i) *A.a.s. $g_1 = d$.*

(ii) *If $2 \le j = O(1)$, then a.a.s.*

$$g_j = d - 2 + \frac{1}{j}.$$

(iii) *If $\omega \le j \le \log_{d-1} n - \omega \log\log n$, then a.a.s.*

$$g_j = d - 2.$$

(iv) *If $\log_{d-1} n - \omega \log\log n \le j \le \log_{d-1} n - \frac{5}{2} s \log_2 \log n + \log_{d-1} \log n - O(1)$ for some $s \in \mathbb{N}^+$, then a.a.s.*

$$d - 2 - \frac{1}{s} \le g_j \le d - 2.$$

(v) *Let $s \in \mathbb{N}^+$, $s \ge 4$. There exists a constant $C_s > 0$ such that if $j \ge \log_{d-1} n + C_s$, then a.a.s.*

$$g_j \le d - 2 - \frac{1}{s}.$$

Theorem 1 tells us that the green number is slightly bigger than $d-2$ if the sinks are located near the source, and then it is $d-2$ for a large interval of j. Later, it might decrease slightly since an increasing number of vertices have already in-degree 2 or more, but only for large j (part (v)) we can prove better upper bounds than $d-2$. One interpretation of this fact is that the resources needed to disrupt the flow of information is in a typical regular DAG is (almost) independent of j, and relatively low (as a function of j).

3.2 Random power law DAGs

We have three parameters in this model: $\beta > 2$, $d > 0$, and $0 < \alpha < 1$. For a given set of parameters, let

$$M = M(n) = n^{\alpha}, \quad i_0 = i_0(n) = n \left(\frac{d}{M} \frac{\beta - 2}{\beta - 1} \right)^{\beta - 1},$$

and

$$c = \left(\frac{\beta - 2}{\beta - 1} \right) d n^{\frac{1}{\beta - 1}}.$$

Finally, for $i \ge 1$ let

$$w_i = c(i_0 + i - 1)^{-\frac{1}{\beta - 1}}.$$

In this case, we refer to the model as *random power law DAGs.*

We note that the sequence $\mathbf{w}$ is decreasing and so the number of coordinates that are at least k is equal to

$$n\left(\frac{\beta-2}{\beta-1}\frac{d}{k}\right)^{\beta-1}-i_0=(1+o(1))n\left(\frac{\beta-2}{\beta-1}\frac{d}{k}\right)^{\beta-1},$$

and hence the sequence follows a power-law with exponent β. From the same observation it follows that the maximum value is

$$w_1=ci_0^{-\frac{1}{\beta-1}}=M.$$

Finally, the average of the first n values is

$$\frac{c}{n}\sum_{i=i_0}^{i_0+n-1}i^{-\frac{1}{\beta-1}}=(1+o(1))\frac{c}{n}\left(\frac{\beta-1}{\beta-2}\right)n^{1-\frac{1}{\beta-1}}=(1+o(1))d,$$

since $M=o(n)$.

Our main result on the green number $g_j=g_j(G,v)$ in the case of power law sequences is the following.

Theorem 2. *Let*

$$\gamma=d^{\beta-1}\left(\frac{\beta-2}{\beta-1}\right)^{\beta-2}\left(\left(1+\left(d\frac{\beta-2}{\beta-1}\right)^{1-\beta}\right)^{\frac{\beta-2}{\beta-1}}-1\right)$$

if $\frac{1}{\alpha}-\beta+3\in\mathbb{N}^+\setminus\{1,2\}$, and $\gamma=1$ otherwise. Let j_1 be the largest integer satisfying $j_1\le\max\{\frac{1}{\alpha}-\beta+3,2\}$. Let $j_2=O(\log\log n)$ be the largest integer such that

$$d^{\beta-1}\left(\frac{\gamma}{d^{\beta-1}}n^{\alpha(j_1-1)-1}\right)^{\left(\frac{\beta-2}{\beta-1}\right)^{j_2-j_1}}\le(\omega\log\log n)^{-\max\{2,(\beta-1)^2\}}.$$

Finally, let

$$\xi=\left(\frac{\beta-2}{\beta-1}\right)d\left(\left(\frac{d(\beta-2)}{\beta-1}\right)^{\beta-1}+1\right)^{-\frac{1}{\beta-1}}.$$

Then for $1\le j\le j_2-1$ we have that a.a.s.

$$(1+o(1))\bar{w}_j\le g_j\le(1+o(1))\bar{w}_{j-1},$$

where $\bar{w}_0=\bar{w}_1=M$, for $2\le j<\frac{1}{\alpha}-\beta+3$,

$$\bar{w}_j=\begin{cases}n^\alpha & \text{if } 2\le j<\frac{1}{\alpha}-\beta+2\\ \xi n^\alpha & \text{if } 2\le j=\frac{1}{\alpha}-\beta+2\\ \left(\frac{\beta-2}{\beta-1}\right)dn^{\frac{1-\alpha(j-1)}{\beta-1}} & \text{if } \frac{1}{\alpha}-\beta+2<j<\frac{1}{\alpha}-\beta+3 \text{ and } j\ge 2,\end{cases}$$

and for $j_1\le j\le j_2-1$,

$$\bar{w}_j=\left(\frac{\beta-2}{\beta-1}\right)\left(\frac{\gamma}{d^{\beta-1}}n^{\alpha(j_1-1)-1}\right)^{-\left(\frac{\beta-2}{\beta-1}\right)^{j-j_1}/(\beta-1)}.$$

In the power law case, Theorem 2 tells us that the green number is smaller for large j. This reinforces the view that intercepting a message in a hierarchical social network following a power law is more difficult close to levels near the source.

4 Proof of Theorem 1 (v)

Owing to space limitations, we focus on the proof of Theorem 1 (v) only. All proofs will appear in the full version of this extended abstract.

Before we proceed with the proof we need an observation. It can be shown (with the proof appearing in the full version) that a.a.s. $|L_t| = (1 - o(1))d(d-1)^{t-1}$ for $t = \log_{d-1} n - \omega$ ($\omega = \omega(n)$ is any function tending to infinity with n, as usual). However, this is not the case when $t = \log_{d-1} n + O(1)$. This, of course, affects the number of edges from L_t to L_{t+1}. In fact, the number of edges between two consecutive layers converges to $c_0 n$ as shown in the next lemma.

Lemma 1. *Let c_0 be the constant satisfying*

$$\sum_{k=1}^{d-1}(d-k)\frac{c^k}{k!}e^{-c} = c.$$

For every $\varepsilon > 0$, there exists a constant C_ε such that a.a.s. for every $\log_{d-1} n + C_\varepsilon \le t \le 2\log_{d-1} n$,

$$(1 - e^{-c_0+\varepsilon})n \quad \le \quad |L_t| \quad \le \quad (1 - e^{-c_0-\varepsilon})n,$$

and the number of edges between L_t and L_{t+1} is at least $(c_0 - \varepsilon)n$ and at most $(c_0 + \varepsilon)n$.

The value of c_0 (and so $1 - e^{-c_0}$ as well) can be numerically approximated. It is straightforward to see that c_0 tends to $d/2$ (hence, $1 - e^{-c_0}$ tends to 1) when $d \to \infty$. Below we present a few approximate values.

d	3	4	5	10	20
c_0	0.895	1.62	2.26	4.98	≈ 10
$1 - e^{-c_0}$	0.591	0.802	0.895	0.993	≈ 1

Table 1. Approximate values of c_0 and $1 - e^{-c_0}$.

Finally, we are ready to prove the last part of Theorem 1.

Proof of Theorem 1(v). We assume that the game is played with parameter $c = d - 2 - \frac{1}{s}$ for some $s \in \mathbb{N}^+ \setminus \{1, 2, 3\}$. For every $i \in \mathbb{N}$, we have that $c_{si+1} = d - 3$, and $c_t = d - 2$, otherwise. To derive an upper bound of g_j that holds a.a.s., we need to prove that a.a.s. there exists no winning strategy for the sludge.

We will use a combinatorial game-type argument. The greens will play *greedily* (that is, they will always protect nodes adjacent to the sludge). Suppose that the sludge occupies node $v \in L_{si+1}$ for some $i \in \mathbb{N}$ (at time $t = si + 2$ he moves from v to some node in L_t) and he has a strategy to win from this node, provided that no node in the next layers is protected by the greens. We will call such a node *sludge-win*. Note that during the time period between $si+2$ and $s(i+1)$, the greens can protect $d-2$ nodes at a time, so they can direct the sludge leaving him exactly one node to choose from at each time-step. Therefore, if there is a node of in-degree at least 2 in any of these layers, the greens can force the sludge to go there and finish the game in the next time-step. This implies that all nodes within distance $s-2$ from v (including v itself) must have in-degree 1 and so the graph is locally a tree. However, at time-step $s(i+1)+1$, the greens can protect $d-3$ nodes, one less than in earlier steps. If the in-degree of a node reached at this layer is at least 3, then the greens can protect all out-neighbours and win. Further, if the in-degree is 2 and there is at least one out-neighbour that is not sludge-win, the greens can force the sludge to go there and win by definition of not being sludge-win. Finally, if the in-degree is 1, the sludge will be given 2 nodes to choose from. However, if there are at least two out-neighbours that are not sludge-win, the greens can "present" them to the sludge and regardless of the choice made by the sludge, the greens win.

We summarize now the implications of the fact that $v \in L_{si+1}$ is sludge-win. First of all, all nodes within distance $s-2$ are of in-degree 1. Nodes at the layer $L_{s(i+1)}$ below v have in-degree at most 2. If $u \in L_{s(i+1)}$ has in-degree 2, then all of the $d-2$ out-neighbours are sludge-win. If $u \in L_{s(i+1)}$ has in-degree 1, then all out-neighbours except perhaps one node are sludge-win. Using this observation, we characterize a necessary condition for a node $v \in L_1$ to be sludge-win. For a given $v \in L_1$ that can be reached at time 1, we define a *sludge-cut* to be the following cut: examine each node of L_{si}, and proceed inductively for $i \in \mathbb{N}^+$. If $u \in L_{si}$ has out-degree $d-1$, then we cut away any out-neighbour and all nodes that are not reachable from v (after the out-neighbour is removed). The node that is cut away is called an *avoided node*. After the whole layer L_{si} is examined, we skip $s-1$ layers and move to the layer $L_{s(i+1)}$. We continue until we reach the sink, the layer $L_j = L_{si'}$ for some i' (we stop at L_j without cutting any further). The main observation is that if the sludge can win the game, then the following claim holds.

Claim. There exists a node $v \in L_1$ and a sludge-cut such that the graph left after cutting is a $(d-1, d-2)$-regular graph, where each node at layer L_{si}, $1 \le i \le i'-1$ has out-degree $d-2$, and all other nodes have out-degree $d-1$. In particular, for any $1 \le i \le i'-1$ the graph induced by the set $\bigcup_{t=si}^{s(i+1)-1} L_t$ is a tree.

It remains to show that a.a.s. the claim does not hold. (Since there are at most d nodes in L_1 it is enough to show that a.a.s. the claim does not hold for a given node in L_1.) Fix $v \in L_1$. The number of avoided nodes at layer L_{si+1} is at most the number of nodes in L_{si} (after cutting earlier layers), which is at most

$$\ell_i = (d-1)^{si-1} \left(\frac{d-2}{d-1} \right)^{i-1} = (d-1)^{(s-1)i} (d-2)^{i-1}.$$

In particular, ℓ, the number of nodes in the sink after cutting, is at most $\ell_{i'} \le n$. It can be shown that a.a.s. $\ell > n^\alpha$ for some $\alpha > 0$.

Fix $n^\alpha \le \ell \le \ell_{i'} \le n$. We need to show that for this given ℓ the claim does not hold with probability $1 - o(n^{-1})$. Since each node in $L_{si'}$ has in-degree at most 2, the number of nodes in $L_{si'-1}$ is at most 2ℓ (as before, after cutting). Since the graph between layer $L_{s(i'-1)}$ and $L_{si'-1}$ is a tree, the number of nodes in $L_{si'}$ is at most $2\ell/(d-1)^{s-1}$, which is an upper bound for the number of avoided nodes at the next layer $L_{si'+1}$. Applying this observation recursively we obtain that the total number of avoided nodes up to layer si' is at most $4(d-1)^{-s+1}\ell$. To count the total number of sludge-cuts of a given graph, observe that each avoided node corresponds to one out of $d-1$ choices. Hence, the total number of sludge-cuts is at most

$$(d-1)^{4(d-1)^{-s+1}\ell}. \tag{1}$$

We now estimate the probability that the claim holds for a given $v \in L_1$ and a sludge-cut. To obtain an upper bound, we estimate the probability that all nodes in the layer $L_{si'-1}$ are of in-degree 1. Conditioning on the fact that we have ℓ nodes in the last layer, we find that the number of nodes in $L_{si'-1}$ is at least $\frac{\ell}{d-1}$. Let i' be large enough such that we are guaranteed by Lemma 1 that the number of edges between the two consecutive layers is at least $c_0 n(1-\varepsilon/2)$. Hence, the probability that a node in $L_{si'-1}$ has in-degree 1 is at most

$$\left(1 - \frac{1}{n}\right)^{c_0 n(1-\varepsilon/2)} = (1+o(1))e^{-c_0(1-\varepsilon/2)} \le e^{-c_0(1-\varepsilon)}, \tag{2}$$

where $\varepsilon > 0$ can be arbitrarily small by taking i' large enough. Let p_ε be the probability in (2). We derive that $j = si' \ge \log_{d-1} n + C'$, where $C' = C'(\varepsilon, s) > 0$ is a large enough constant. Conditioning under $v \in L_{si'-1}$ having in-degree 1, it is harder for $v' \in L_{si'-1}$ to have in-degree 1 than without this condition, as more edges remain to be distributed. Thus, the probability that all nodes in $L_{si'-1}$ have the desired in-degree is at most

$$p_\varepsilon^{\frac{\ell}{d-1}} = \exp\left(-c_0(1-\varepsilon)\frac{\ell}{d-1}\right). \tag{3}$$

Thus, by taking a union bound over all possible sludge-cuts (the upper bound for the number of them is given by (1)), the probability that the claim holds is at most

$$\left((d-1)^{4(d-1)^{-s+1}} \left(e^{-c_0(1-\varepsilon)}\right)^{\frac{1}{d-1}}\right)^\ell$$

which can be made $o(n^{-1})$ by taking ε small enough, provided that s is large enough so that

$$(d-1)^{4(d-1)^{-s+2}} e^{-c_0} < 1.$$

By considering the extreme case for the probability of having in-degree one when $d = 3$ we obtain that

$$e^{-c_0} \le e^{-\frac{0.895}{3}d} \le e^{-0.29d}$$

for $d \ge 3$ (see Table 1). It is straightforward to see that $s \ge 4$ will work for any $d \ge 3$, and $s \ge 3$ for $d \ge 5$. $\square$

5 Conclusions and future work

We introduced a new stochastic model for DAGs, and analyzed the vertex-pursuit game Seepage in the model. We focused on two cases: random regular DAGs and random power law DAGs. In the d-regular case, our main result was Theorem 1, which demonstrated that the green number is close to $d-2$ throughout the process. One interpretation of this is that an effective strategy to disrupt regular DAGs is to do so near the source (as it takes roughly the same resources for all j). In the random power law DAG case, we give bounds on the green number in Theorem 2. In the power law case the green number is smaller for large j. This reinforces the view that intercepting a message in a hierarchical social network following a power law is more difficult close to levels near the source. More work remains to be done in the regular case: in particular, we did not derive tight bounds on the green number for values of j between $\log_{d-1} n - \Theta(\log\log n)$ and $\log_{d-1} n + O(1)$. In addition, it would be interesting to analyze Seepage in a model of sequences different from regular and power law ones.

References

1. J.A. Almendral, L. López, M.A.F. Sanjuán, Information flow in generalized hierarchical networks, *Physica A* **324** (2003) 424–429.
2. A. Bonato, *A Course on the Web Graph*, Graduate Studies in Mathematics Series, American Mathematical Society, Providence, Rhode Island, 2008.
3. A. Bonato, R.J. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, Providence, Rhode Island, 2011.
4. J.T. Chayes, B. Bollobás, C. Borgs, O. Riordan, Directed scale-free graphs, In: *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.
5. F.R.K. Chung, L. Lu, *Complex graphs and networks*, American Mathematical Society, Providence RI, 2006.
6. N.E. Clarke, S. Finbow, S.L. Fitzpatrick, M.E. Messinger, R.J. Nowakowski, Seepage in directed acyclic graphs, *Australasian Journal of Combinatorics* **43** (2009) 91–102.
7. R. Diestel, *Graph theory*, Springer-Verlag, New York, 2000.
8. J.D. Farley, Breaking Al Qaeda cells: a mathematical analysis of counterterrorism operations (A guide for risk assessment and decision making), *Studies in Conflict & Terrorism* **26** (2003) 399–411.
9. J.D. Farley, *Toward a Mathematical Theory of Counterterrorism*, The Proteus Monograph Series Jonathan David Farley, Stanford University, 2007.
10. M. Gupte, S. Muthukrishnan, P. Shankar, L. Iftode, J. Li, Finding hierarchy in directed online social networks, In: *Proceedings of WWW'2011*.
11. A. Gutfraind, A. Hagberg, F. Pan, Optimal interdiction of unreactive Markovian evaders, In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Hoeve, Willem-Jan van; Hooker, John N. (Eds), (Springer Berlin / Heidelberg) 2009.
12. K. Ikeda, S.E. Richey, Japanese network capital: the impact of social networks on japanese political participation, *Political behavior* **27** (2005) 239–260.
13. L. López, J.F.F. Mendes, M.A.F. Sanjuán, Hierarchical social networks and information flow, *Physica A: Statistical Mechanics and its Applications* **316** (2002) 695–708.
14. Twitaholic. Accessed January 10, 2012. `http://twitaholic.com/`.
15. D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.
16. N.C. Wormald, Models of random regular graphs, *Surveys in Combinatorics*, 1999, J.D. Lamb and D.A. Preece, eds. London Mathematical Society Lecture Note Series, vol 276, pp. 239–298. Cambridge University Press, Cambridge, 1999