How to Explore a Fast-Changing World

(Cover Time of a Simple Random Walk on Evolving Graphs)

Chen Avin[1], Michal Koucký[2], and Zvi Lotker[1]

[1] Communication Systems Engineering, Ben Gurion University of the Negev, Israel
{avin,zvilo}@cse.bgu.ac.il
[2] Institute of Mathematics, Academy of Sciences of the Czech Republic
koucky@math.cas.cz

Abstract. Motivated by real world networks and use of algorithms based on random walks on these networks we study the simple random walks on *dynamic* undirected graphs with fixed underlying vertex set, i.e., graphs which are modified by inserting or deleting edges at every step of the walk. We are interested in the expected time needed to visit all the vertices of such a dynamic graph, the *cover time*, under the assumption that the graph is being modified by an oblivious adversary. It is well known that on connected *static* undirected graphs the cover time is polynomial in the size of the graph. On the contrary and somewhat counter-intuitively, we show that there are adversary strategies which force the expected cover time of a simple random walk on connected dynamic graphs to be exponential. We relate this result to the cover time of static directed graphs. In addition we provide a simple strategy, the *lazy* random walk, that guarantees polynomial cover time regardless of the changes made by the adversary.

1 Introduction

A random walk on a graph is a simple process of visiting the nodes of the graph in some random sequential order. The walk starts at some fixed node, and at each step it moves to a neighbor of the current node chosen at random. The random walk is called *simple* when the next node is chosen uniformly at random from the set of neighbors. In the context of communication networks (e.g., Internet, wireless ad-hoc networks and sensor networks) and information networks (e.g., peer-to-peer file sharing networks and distributed databases), a random walk on a network (graph) will result when messages are sent at random from device to device.

Since this process presents locality, simplicity, low memory-overhead and robustness to changes in the network structure applications based on random-walk techniques are becoming more and more popular in the networking community. In recent years, different authors have proposed the use of random walk for a large variety of tasks and networks; to name but a few: querying in sensor and ad-hoc networks [18, 5, 1], searching in peer-to-peer networks [12], gossiping [16], PageRank and search engines on the web [13].

One of the main reasons that random walk techniques are so appealing for networking application is their robustness to dynamics. Many communication networks are subject to dramatic structural changes created by mobility, sleep modes, channel fluctuations, device failures, nodes joining or leaving the system and other factors. Topology-driven algorithms are at a disadvantage for such networks, since they incur high overhead to maintain up-to-date topology and routing information such as routing tables, clusters and spanning trees. In contrast, algorithms that require no knowledge of network topology, such as the random walk, are at an advantage.

While at first glance, the process of a token wandering randomly in the network may seem overly simplistic and highly inefficient, many encouraging results prove that it is comparable to other approaches that have been used over the years. One important property of random walks on graphs that needs to be evaluated to study the efficiency of the approach is the *cover time* [2]. The *cover time* C_G of a graph G is the expected time (measured by number of steps or in our case by the number of messages) taken by a simple random walk to visit all the nodes in G. Methods on bounding the cover time of graphs have been thoroughly investigated with the major result being that cover time is always at most polynomial for undirected graphs. More precisely, it has been shown by Aleliunas *et al.* in their seminal work [3] that C_G is always $O(mn)$, where m is the number of edges in the graph and n is the number of nodes. Tighter bounds for many classes of graphs have been established and they can be found in the extensive literature on the subject.

Since real-world networks change over time researches have recently started to study random walks on such dynamic graphs. Motivated by robotic exploration of the Web, Cooper and Frieze [7] studied the question of covering a graph that grows over time. They considered a particular model of so-called *web graphs* and showed that a simple random walk on the graph fails to visit a constant fraction of nodes if a new node appears and is connected to the graph after every constant number of steps of the walk.

Motivated by sensor networks we consider a similar question on a different model of dynamic graphs. We consider dynamic graphs with fixed number of nodes where connections between the nodes appear and disappear over the time. The question that we study is the cover time of such graphs.

1.1 Overview of Our Results

We show that somewhat counter-intuitively, there are dynamic graphs of the type given above that have exponential cover time when explored by a simple random walk. (For the sake of clarity let us say that our examples are deterministic but oblivious to the actual random walk.) Moreover, we show that a random walk on any directed graph G can be simulated (in a way we define later) by a random walk on an undirected dynamic graph that we construct from G; this gives yet another justification to our previous claim. Our examples are also valid when we allow the random walk to make more than a single step between each graph change. Indeed, we can allow up-to $n^{1-\epsilon}$ steps before making each change and still obtain an exponential cover time. Although one could question whether our graphs could appear in a real-word scenario we do not consider these graphs to be far-fetched: for example a particular implementation of sensor networks with links (network interfaces) going to sleep periodically or nodes switching communication frequencies could exhibit such behavior.

In addition to these negative results we also show several positive results. Most importantly we show that a *lazy* random walk (also known as a *max-degree* random walk in literature [15]) does not suffer from these issues. We define as a lazy random walk a walk that picks each adjacent edge with probability $1/d_{\max}$, where $d_{\max}$ is the maximum degree of the graph, and with the remaining probability it stays at the current vertex. We show that a lazy random walk covers any connected dynamic graph in time polynomial in the size of the graph. Furthermore, we also show that when the dynamic graph itself is obtained by sampling from a certain probability distribution, a simple random walk will also cover such a graph in expected polynomial time.

2 Models and Preliminaries

2.1 Random Walks on Graphs

Let $G(V, E)$ be an undirected graph, with V the set of nodes and E the set of edges. Let $n = |V|$ and $m = |E|$. For $v \in V$, let $N(v) = \{u \in V \mid (v, u) \in E\}$ be the set of neighbors of v and $d(v) = |N(v)|$ the degree of v. A d-regular graph is a graph in which the degree of all the nodes is d.

The *simple random walk* on a graph G is a walk on G where the next node is chosen uniformly at random from the set of neighbors of the current node, i.e., when the walk is at node v, the probability to move in the next step to u is $P(v, u) = \frac{1}{d(v)}$ for $(v, u) \in E$ and 0 otherwise.

The *hitting time*, H_{uv}, is the expected time for a random walk starting at u to arrive at v for the first time, and the *commute time*, C_{uv}, is the expected time for a random walk starting at u to first arrive at v and then return to u. Let $H_{\max}$ be the maximum hitting time over all the pairs of nodes in G.

The *cover time* C_G of a graph G is the expected time taken by a simple random walk on G to visit all the nodes in G. Formally, for $v \in V$, let C_v be the expected number of steps needed for the simple random walk starting at v to visit all the nodes in G, and the cover time of G is $C_G = \max_v C_v$. The *cover time* of graphs and methods of bounding it have been extensively investigated [17]. Results for the cover time of specific graphs vary from the *optimal cover time* of $\Theta(n \log n)$ associated with the complete graph to the worst case of $\Theta(n^3)$ associated with the lollipop graph [9, 8].

2.2 Evolving Graphs Model

The most general model to describe a dynamic network is called the Evolving Graph model. We will use a similar definition as in [14, 11, 10].

Definition 1 (Evolving Graphs) *Let $\mathcal{G} = G_1, G_2, \ldots$ be an infinite sequence of graphs on the same vertex set V. We call this sequence an* evolving graph. *We say that $\mathcal{G}$ has the graph property X if every graph G_i in the sequence has the property X.*

In simple words at time i the structure of the evolving graph $\mathcal{G}$ is G_i. For an integer $\tau \geq 1$, an evolving graph $\mathcal{G}$ is *evolving with rate* $\frac{1}{\tau}$ if for all $i \geq 1$, $G_i \neq G_{i+1}$ implies, $G_{i+1} = G_{i+1+j}$ for all $j \in \{0, \ldots, \tau - 1\}$.

A simple random walk on evolving graph $\mathcal{G}$ is defined as follows: assume that at time i the walker is at node $v \in V$, and let $N(v)$ be the set of neighbors of v in G_i, then the walker moves to one of its neighbors from $N(v)$ uniformly at random.

The strength and the weakness of the above model have the same origin, its generality. On the positive side, it captures many interesting scenarios of dynamic networks, but on the other hand, most natural problems are NP-complete such as finding strongly connected components and the equivalence of minimum spanning tree [10].

2.3 Constructive Evolving Graphs Model

Evolving graphs do not capture the underlying mechanism of how (or why) the graph evolves. In many situations the evolving graph itself is a product of some random process. (For example this is the case of web graphs considered in [7].) We will use the following definition to capture the underlying process in the case it is a Markov chain. A special case of such graphs is considered in Section 5.

Definition 2 (Markovian Evolving Graphs) *Let the* space set $\mathbf{G}$ *be a set of graphs with the same set V of nodes, let $G_1 \in \mathbf{G}$ and let P be a probability transition matrix on $\mathbf{G}$. A* Markovian evolving graph $\mathcal{M} = (\mathbf{G}, G_1, P)$ *is an evolving graph $\mathcal{M}_1, \mathcal{M}_2, \ldots$ obtained by the Markov chain given by P with the initial state G_1. Thus for any sequence of graphs $G_2, G_3, \cdots \in \mathbf{G}$ and any $t > 1$,* $\Pr[\mathcal{M}_t = G_t \mid \mathcal{M}_{t-1} = G_{t-1}]$ *is given by the appropriate entry of P.*

It is clear that a Markovian evolving graph is a random variable. By a random walk on a Markovian evolving graph we understand a random walk on the outcome of the random variable. When considering the expected cover time of a random walk on a Markovian evolving graph we consider the expectation over the choice of both the evolving graph and the random walk.

3 Exponential Hitting Time of Evolving Graphs

In this section we address the cover time of the simple random walk on evolving graphs by studying the maximum hitting time. Clearly the cover time must be at least as large as the maximum hitting time. First, we mention some technical issues. On static graphs the cover time is finite only for connected graphs. This is not the case for evolving graphs as we will see in Section 5. For simplicity though we restrict our discussion mostly to evolving graphs in which every graph in the sequence $\mathcal{G}$ is connected. (In the Markovian model we require that every graph in the set $\mathbf{G}$ is connected, and call $\mathbf{G}$ *connected* if that is the case). Moreover we require that all graphs have a self-loop for each of the nodes. This is simply a technical condition to avoid pathological cases such as the walk switching forever between two nodes. In the case of static graphs this is a standard way of enforcing ergodicity. An evolving graph $\mathcal{G}$ that has the above properties we call an *explorable* evolving graph.

Now one can easily claim the following for explorable evolving graphs (a similar claim can be made for Markovian evolving graphs):

Claim 3 *Let $\mathcal{G}$ be an explorable evolving graph, then the cover time of $\mathcal{G}$ is bounded by $n^{O(n)}$.*

We outline the argument here. Let V be the set of vertices of $\mathcal{G}$. Fix two vertices u and v from V. For $i \geq 1$, let V_i be the set of vertices that could be visited within first i steps of a simple random walk on $\mathcal{G}$ starting from u. Since $\mathcal{G}$ is connected it must be the case that for each i, $V_i \subsetneq V_{i+1}$ unless $V_i = V$. In particular, V_n must contain all vertices of $\mathcal{G}$ and in particular v. Thus, the probability of reaching v starting from u is at least n^{-n}. Indeed, this is true for any two vertices u and v and starting from any time t. A standard argument now implies that the cover time is at most $n^{O(n)}$.

Requiring connectivity at each step of the evolving graph may look like a very strong condition that should imply polynomial cover time and maximum hitting time. Surprisingly we show that this is not the case.

Theorem 4 *There exists an explorable evolving graph $\mathcal{G}$, such that the maximum hitting time of the simple random walk on $\mathcal{G}$ is $\Omega(2^n)$.*

One can think of this result in the following way: consider a random walk on an evolving graph that is controlled by an oblivious *adversary* that is deciding what will be the next graph at each time step. In such a case the adversary, although unaware of the random walk location, can force the walk to step exponential number of steps before exploring the whole graph. We give below the basic details of the proof.

Proof of Theorem 4. Let G_1 be the star of size n (with the addition of a self-loop at each node) where nodes called $1, 2, \ldots, n-2$ and n are always the leafs and node called $n-1$ is always the center. The random walk starts at the node originally called 1 and we will bound the hitting time to the node called n. The adversary is the following *deterministic* process: At each time step vertices $1, \ldots, n-1$ will trade their places, i.e., the adversary changes the edges by changing the names of the nodes. The adversary uses the following renaming strategy: for $1 \leq i \leq n-1$, node i changes its name to $i + 1 (\bmod\, n-1)$. Note that node n does not change its name, nodes $1, \ldots, n-2$ increase their name by one, node $n-1$ becomes 1.

The only way to reach node n is through the center. By induction on $i = 2, \ldots, n-2$ one can see the following. Unless we have already reached the center

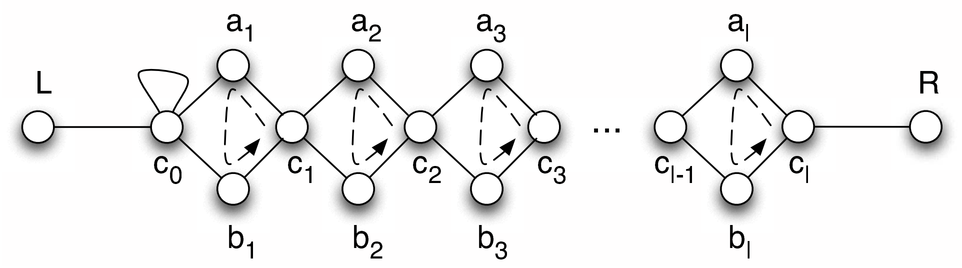

Fig. 1. the gadget $\mathcal{H}_\ell$ (dashed lines show node transformations)

of the star the only way to be at the leaf named i after the adversary move is to be at the leaf named $i-1$ before the adversary renaming. That implies we must have used a self-loop at that random step. Hence, to get to the leaf named $n-2$ we must have had a sequence of $n-3$ random steps all taking a self-loop. To get to the center we have to stay at leaf $n-2$. All in all to be at the center after the adversary move the random walk must have made a sequence of $n-2$ consecutive self-loop steps. That happens with probability 2^{-n+2} in a sequence of $n-2$ consecutive steps. Therefore the expected time before we observe the random walk to make such a sequence of steps is $\Omega(2^n)$. □

We would like to point out that all graphs in $\mathcal{G}$ are isomorphic and rapidly mixing (the cover time of each of them is in fact $O(n \log n)$). This fact shows that common tools like spectral analysis cannot be applied naïvely to dynamic graphs.

3.1 Simulating Directed Graphs

One way to understand the results of the previous section is by relating random walks on explorable evolving graphs to random walks on static directed graphs. In fact we can simulate a simple random walk on a directed graph G by a careful choice of evolving graph $\mathcal{G}$. We will use the following gadget $\mathcal{H}$ to replace every directed edge of G. For $\ell > 0$, the gadget $\mathcal{H}_\ell$ is a sequence of graphs $H_\ell^0, H_\ell^1, H_\ell^2, H_\ell^0, H_\ell^1, H_\ell^2, H_\ell^0, \ldots$ with vertices L, R, s_0 and $s_{i,j}$, for $i = 1, \ldots, \ell$ and $j = 0, 1, 2$. The graph H_ℓ^k is obtained from the graph in Fig. 1 by mapping vertices $L \to L$, $R \to R$, $s_0 \to c_0$ and $s_{i,k} \to c_i$, $s_{i,k+1 \bmod 3} \to b_i$, $s_{i,k+2 \bmod 3} \to a_i$, $i = 1, \ldots, \ell$. (We deviate here from our convention of having self-loops at every node for the sake of simplicity of the analysis. As it will be clear in the next section with minor modification of bounds our claims would be true even if we would add a self-loop to every node.) The main property of a simple random walk on $\mathcal{H}$ is summarized in the following lemma.

Lemma 5 *Let $\ell > 0$, $\mathcal{H}_\ell = H_\ell^0, H_\ell^1, H_\ell^2, H_\ell^0, H_\ell^1, H_\ell^2, H_\ell^0, \ldots$ and $\epsilon = \ell(1/2)^\ell + (3/4)^\ell$. Consider a simple random walk on $\mathcal{H}_\ell$. If the walk starts at vertex L then the probability of returning to L before visiting R is at least $1-\epsilon$. Moreover if* *the walk starts at vertex R then the probability of returning to R before visiting L is at most ϵ.*

We omit the proof of this lemma due to space constraints. Thus the gadget $\mathcal{H}_\ell$ has essentially the same effect for a simple random walk as a directed edge from R to L with a self-loop at L. Given a directed graph G with a self-loop at every vertex we can replace all its directed edges between different vertices by a copy of $\mathcal{H}_\ell$ to obtain a sequence of graphs $\mathcal{G}$ on which a simple random walk will simulate a simple random walk on G (up-to some error ϵ). Of course, replacing several edges incoming to a vertex by the gadget will introduce several self-loops to that vertex. To avoid that we can collapse the vertices c_0 from these gadgets into one thus obtaining an equivalent of one self-loop. (This collapse will affect ϵ slightly but no more than by a factor polynomial in the number of replaced edges.) We also remove the original self-loops from the graph G.

If we perform a simple random walk on $\mathcal{G}$ and we restrict ourselves to observing only visits to the vertices of the original graph G we will observe essentially the same probability distribution as of a simple random walk on G. In particular, if we choose $\ell = n^{k+1}$, for $k > 1$ and n being the size of G, then the probability of observing an edge being traversed in the opposite direction in the first 2^{n^k} steps is at most $2^{-O(n^{k+1})}$. Since for example the maximal hitting time on any strongly connected directed graph is bounded by $2^{O(n \log n)}$ this error is negligible.

4 Slowly Evolving Graphs

The previous section has shown that there are evolving graphs for which a simple random walk essentially fails as a means of exploring it. All our examples so far considered graphs that evolve at rate one. This would not really be a typical case in a real-world application. The rate at which graphs evolve is usually slower compared to unit operations such as sending a packet. So could it be the case that a simple random walk covers in polynomial time all graphs evolving at lower rate? In this section we show that this is not the case. Namely for any constant $0 < \epsilon < 1$ and an integer n large enough, we provide an example of an evolving graph on $O(n)$ vertices that evolves at rate $\frac{1}{n^{1-\epsilon}}$ so that a simple random walk needs expected time $2^{\Omega(n^\epsilon)}$ to cover the graph. Indeed the graph is essentially the gadget from the previous section with the speed of evolution slowed down.

Let F^i_ℓ be the graph H^i_ℓ from the previous section modified by adding possibly several self-loops to each vertex so that the probability of staying at the same vertex is precisely one half. (So in particular vertices of degree two will receive two self-loops and vertices of degree four will receive four of them. We remark that our claim would be true even without these self-loops but in some cases for trivial reasons. So to capture the most general situation we introduce the loops.)

For $0 < \epsilon < 1$ and an integer $n \geq 2^{1/(1-\epsilon)}$, we define an evolving graph $\mathcal{G}^\epsilon_n$ to consist of repeated sequence $F^0_{2n}, F^0_{2n}, \dots, F^0_{2n}, F^1_{2n}, F^1_{2n}, \dots, F^1_{2n}, F^2_{2n}, \dots, F^2_{2n},$

where each block of consecutive F_{2n}^i's consists of $n^{1-\epsilon}$ copies of F_{2n}^i. Clearly $\mathcal{G}_n^\epsilon$ evolves at rate $\frac{1}{n^{1-\epsilon}}$. We claim:

Theorem 6. *The cover time of* $\mathcal{G}_n^\epsilon$ *is* $2^{\Omega(n^\epsilon)}$.

In order to prove the theorem we analyze a concept that we call a *random walk on a line with a drift.* A random walk on a line with a drift is a simple random walk on a line of size n where each $\ell = n^{1-\epsilon}$ steps, a step biased towards the same direction is taken. We show that such a walk requires in expectation an exponential number of steps to traverse the line in the direction opposite to the bias. Due to space limitations we omit the detailed analysis from this version (see [6] for the proof).

5 Polynomial Cover Time of Dynamic Graphs

We turn our attention to cases where the cover time of evolving graphs is "good", i.e., polynomial. Our first example is of a simple Markovian case.

Definition 7 (Bernoulli evolving graph) *Let* $\mathbf{G}$ *be a set of graphs with the same set* V *of nodes and let* $\bar{P}$ *be a probability distribution over* $\mathbf{G}$. *A Bernoulli evolving graph* $\mathcal{B} = (\mathbf{G}, \bar{P})$ *is a Markovian evolving graph in which the rows of the transition matrix* P *are identical and equal to* $\bar{P}$ *and the initial graph* G_1 *is taken at random according to* $\bar{P}$, *i.e., the random graphs* G_i, *are i.i.d.*

We show that the bound for the cover time of the simple random walk on Bernoulli evolving graphs is very similar to the bound of static graphs; essentially when the process is time invariant and the graph is always connected then the bound of Aleliunas *et al.* [3] can be extended to dynamic graphs.

Theorem 8 *For any explorable Bernoulli evolving graph,* $\mathcal{B} = (\mathbf{G}, \bar{P})$, *the cover time of the simple random walk on* $\mathcal{B}$ *is* $O(n^3 \log n)$ *and the maximum hitting time is* $O(n^3)$.

The property that $\mathbf{G}$ is *connected* is not necessary to obtain a polynomial bound on the cover time as the following statement shows. (We omit proofs of both of these theorems due to space limitations.)

Theorem 9 *Let* $\mathcal{B} = (\mathbf{G}, \bar{P})$ *be a Bernoulli evolving graph,* $\mathbf{G}$ *be the set of all maximum matching of the complete graph (any such graph is disconnected) and* $\bar{P}$ *is the uniform distribution over* $\mathbf{G}$. *The cover time of the simple random walk on* $\mathcal{B}$ *is the same as the cover time of the complete graph,* $n \log n(1 + o(1))$.

5.1 *d*-regular Dynamic Graphs

It is known that simple random walks on regular, connected, non-bipartite static graph have cover time of $O(n^2)$ [17]. Interestingly, it turns out that a similar result holds true for regular, connected, non-bipartite evolving graphs.

Theorem 10. *For any d-regular connected non-bipartite evolving graph $\mathcal{G}$ the cover time of the simple random walk on $\mathcal{G}$ is $O(d^2n^3\ln^2 n)$.*

We will need the following lemma proof of which omitted is due to space limitations:

Lemma 11 *Let G be an undirected d-regular (multi)graph on n vertices and $p = (p_1, \ldots, p_n)$ be a probability distribution on its vertices. Let A_G be the transition matrix of a simple random walk on G. Then:*

1.

$$\left\| pA_G - \frac{\mathbb{I}}{n} \right\|_2^2 \leq \left\| p - \frac{\mathbb{I}}{n} \right\|_2^2 .$$

2. If G is connected non-bipartite

$$\left\| pA_G - \frac{\mathbb{I}}{n} \right\|_2^2 \leq \left(1 - \frac{1}{d^2n^2}\right) \left\| p - \frac{\mathbb{I}}{n} \right\|_2^2 .$$

Here $\mathbb{I}$ stands for a vector of ones of an appropriate dimension.

As an immediate corollary to the previous lemma we obtain:

Corollary 12 *Let $\mathcal{G} = G_1, G_2, \ldots$ be a sequence of d-regular graphs on the same vertex set $V = \{1, \ldots, n\}$. For integers $0 \leq \ell \leq t$ let at least ℓ of the graphs $G_1, \ldots, G_t$ be non-bipartite connected. If p_0 is the initial probability distribution on V and we perform a simple random walk on $\mathcal{G}$ starting from p_0, then the probability distribution p_t of the walk after t steps satisfies:*

$$\left\| p_t - \frac{\mathbb{I}}{n} \right\|_2^2 \leq \left(1 - \frac{1}{d^2n^2}\right)^{\ell} \left\| p_0 - \frac{\mathbb{I}}{n} \right\|_2^2 .$$

A technique similar to [4] gives the following lemma.

Lemma 13 *Let $Y_0, Y_1, Y_2, \ldots$ be a sequence of random variables with range $V = \{1, \ldots, n\}$ satisfying for all $u, v \in V$ and $i > 0$, $\Pr[Y_i = u | Y_{i-1} = v] \geq 1/2n$. If $t = \min\{i;\ \{Y_0, Y_1, \ldots, Y_i\} = V\}$ then the expectation $E[t] \leq 3n \ln n + O(\sqrt{n} \ln n)$.*

Proof. For every $\ell > 0$ and every $v \in V$, $\Pr[v \notin \{Y_{\ell+1}, \ldots, Y_{\ell+3n\ln n}\}] < (1 - 1/2n)^{3n\ln n} < e^{-(3/2)\ln n} = n^{-3/2}$. Thus, $\Pr[\exists v \in V;\ v \notin \{Y_{\ell+1}, \ldots, Y_{\ell+3n\ln n}\}] < n \cdot n^{-3/2} = 1/\sqrt{n}$. For each integer $k \geq 0$, if we set $\ell = k \cdot 3n \ln n$ then the probability that $Y_{\ell+1}, \ldots, Y_{\ell+3n\ln n}$ does not cover whole V is at most $1/\sqrt{n}$. Thus the expected k before V is covered is at most $1/(1 - 1/\sqrt{n}) = 1 + O(1/\sqrt{n})$. Hence the expected cover time of V is bounded by $E[t] \leq 3n \ln n + O(\sqrt{n} \ln n)$. □

Now, we can prove Theorem 10.

Proof of Theorem 10. Let $X_0, X_1, \ldots$ be a random walk on $\mathcal{G}$. For an integer $i \geq 0$, define $Y_i = X_{i \cdot 4d^2n^2 \ln n}$. Pick $u, v \in V$. For $i > 1$, let p_i be the probability distribution of Y_i conditioned on $Y_{i-1} = v$. By Corollary 12, $\left\|p_i - \frac{\mathbb{I}}{n}\right\|_2^2 \leq \left(1 - \frac{1}{d^2n^2}\right)^{4d^2n^2 \ln n} < n^{-4}$. Hence, all coordinates of the vector $(p_i - \frac{\mathbb{I}}{n})$ are in absolute value smaller than $1/n^2$. Thus $\Pr[Y_i = u | Y_{i-1} = v] \geq \frac{1}{n} - \frac{1}{n^2} \geq 1/2n$, provided that $n \geq 2$. Applying Lemma 13 yields the result. □

6 Random Walk Strategy

Consequently to the previous section the following simple strategy for the random walk guarantees that an evolving graph will be covered in expected polynomial time:

Definition 14 (Lazy Random Walk) *At each step of the walk pick a vertex v from $V(G)$ uniformly at random and if there is an edge from the current vertex to the vertex v then we move to v otherwise we stay at the current vertex.*

In effect what this strategy does is that it makes the graph n-regular; every edge adjacent to the current vertex is picked with the probability $1/n$ and with the remaining probability we use one of many self-loops. If we have an a priori upper bound $d_{\max}$ on the maximum degree of the dynamic graph we can achieve a slightly faster cover time. In that case we can reformulate the strategy as follows:

> At each step of the walk with probability $1 - (d(u)/(d_{\max} + 1))$ stay at the current vertex u and with the remaining probability pick uniformly at random one of the neighbors v of the current vertex and move to v.

We call this strategy $d_{\max}$-*lazy random walk*. (In literature such a walk is sometimes called *max-degree* random walk [15].) If the only upper bound on the maximum degree that we have is n then this strategy becomes the previous one. We claim the following as an immediate corollary of Theorem 10:

Theorem 15. *For any connected evolving graph $\mathcal{G}$ with maximum degree $d_{\max}$ the cover time of the $d_{\max}$-lazy random walk on $\mathcal{G}$ is $O(d_{\max}^2 n^3 \ln^2 n)$.*

Indeed these strategies do not even require the dynamic graph to be connected at each step. By Corollary 12 and Lemma 13 as long as the dynamic graph is connected for polynomial fraction of the time, the cover time of a random walk using our strategy will still be polynomial. In that case we can obtain the following generalization of Theorem 15.

Theorem 16. *Let $\mathcal{G} = G_1, G_2, \ldots$ be an evolving graph with maximum degree $d_{\max}$. Let $\epsilon > 0$ be such that for every integer ℓ, at least $\epsilon\ell$ graphs among $G_1, G_2, \ldots, G_\ell$ are connected. Then the cover time of the $d_{\max}$-lazy random walk on $\mathcal{G}$ is $O(\epsilon^{-1} d_{\max}^2 n^3 \ln^2 n)$.*

The constant in the big-O is a universal constant that is independent of $\mathcal{G}$.

We point out that the strategy can be modified so that the a priori knowledge of $d_{\max}$ and n is not necessary. First, we can assume that $d_{\max} = n$. Second, we can try different estimates for n as follows. We start with the estimate $n = 10$. Then we always walk for $n^5 \ln^2 n$ steps as in the n-lazy random walk, where n is the current estimate, and after that we double our estimate of n. One can show that this strategy will provide $O(n^5 \ln^2 n)$ expected cover time of the random walk on a connected evolving graph with n vertices.

7 Conclusions

In this paper we demonstrate that the cover time of the simple random walk on dynamic graphs is significantly different from the case of static graphs. While the latter was well known to be polynomial, the former is shown here to be exponential on some evolving graphs. Moreover, we show that even if the random walk takes many steps before the graph evolves the cover time can still be exponential.

We prove that in order to accelerate the cover time one can use a *lazy* random walk and reduce the cover time to polynomial. This approach has been used previously on static graphs in order to sample nodes uniformly at random, but contrary to our situation, it can be shown that it cannot accelerate the cover time for static graphs.

To summarize, the main results presented here provide theoretical justification to the wide use of random-walk-techniques in dynamic networks. Nevertheless, one must pay careful attention to the network dynamics when choosing the implementation of the random walk.

Acknowledgements Part of this work was done while Michal Koucký was visiting Ben Gurion University. Michal Koucký was supported in part by grant GA ČR 201/07/P276, project No. 1M0021620808 of MŠMT ČR and Institutional Research Plan No. AV0Z10190503.

References

1. Alanyali, M., Saligrama, V., and Sava, O. A random-walk model for distributed computation in energy-limited network. In *In Proc. of 1st Workshop on Information Theory and its Application* (San Diego, 2006).
2. Aldous, D., and Fill, J. *Reversible Markov Chains and Random Walks on Graphs.* 1999. Unpublished. http://stat-www.berkeley.edu/users/aldous/RWG/book.html.
3. Aleliunas, R., Karp, R. M., Lipton, R. J., Lovász, L., and Rackoff, C. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979).* IEEE, New York, 1979, pp. 218–223.

4. Alon, N., Avin, C., Koucký, M., Kozma, G., Lotker, Z., and Tuttle, M. R. Many random walks are faster than one. In *20th ACM Symposium on Parallelism in Algorithms and Architectures, to appear.* (2008).
5. Avin, C., and Brito, C. Efficient and robust query processing in dynamic environments using random walk techniques. In *Proc. of the Third International Symposium on Information Processing in Sensor Networks* (2004), pp. 277–286.
6. Avin, C., Koucký, M., and Lotker, Z. How to explore a fast-changing world. Tech. Rep. pre-116, Institute of Mathematics of the Academy of Sciences of the Czech Republic, 2007. http://www.math.cas.cz/preprint/pre-116.pdf.
7. Cooper, C., and Frieze, A. Crawling on simple models of web graphs. *Internet Mathematics 1* (2003), 57–90.
8. Feige, U. A tight lower bound on the cover time for random walks on graphs. *Random Structures and Algorithms 6*, 4 (1995), 433–438.
9. Feige, U. A tight upper bound on the cover time for random walks on graphs. *Random Structures and Algorithms 6*, 1 (1995), 51–54.
10. Ferreira, A. Building a reference combinatorial model for manets. *Network, IEEE 18*, 5 (2004), 24–29.
11. Ferreira, A., Goldman, A., and Monteiro, J. On the evaluation of shortest journeys in dynamic networks. In *Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007)* (2007), pp. 3–10.
12. Gkantsidis, C., Mihail, M., and Saberi, A. Random walks in peer-to-peer networks. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies 1* (2004), –130.
13. Henzinger, M., Heydon, A., Mitzenmacher, M., and Najork, M. Measuring index quality using random walks on the Web. *WWW8 / Computer Networks 31*, 11-16 (1999), 1291–1303.
14. Jarry, A., and Lotker, Z. Connectivity in evolving graph with geometric properties. In *DIALM-POMC '04: Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing* (New York, NY, USA, 2004), ACM Press, pp. 24–30.
15. Jerrum, M., and Sinclair, A. Approximating the permanent. *SIAM Journal on Computing 18*, 6 (1989), 1149–1178.
16. Kempe, D., Dobra, A., and Gehrke, J. Gossip-based computation of aggregate information. In *Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science* (2003), pp. 482–491.
17. Lovász, L. Random walks on graphs: A survey. In *Combinatorics, Paul Erdos is eighty, Vol. 2 (Keszthely, 1993)*, vol. 2 of *Bolyai Soc. Math. Stud.* János Bolyai Math. Soc., Budapest, 1996, pp. 353–397.
18. Sadagopan, N., Krishnamachari, B., and Helmy, A. Active query forwarding in sensor networks (acquire). *Journal of Ad Hoc Networks 3*, 1 (January 2005), 91–113.