

Coalescing-Branching Random Walks on Graphs

CHINMOY DUTTA, *Twitter*

GOPAL PANDURANGAN, *University of Houston*

RAJMOHAN RAJARAMAN, *Northeastern University*

SCOTT ROCHE, *Northeastern University*

We study a distributed randomized information propagation mechanism in networks we call the *coalescing-branching* random walk (cobra walk, for short). A cobra walk is a generalization of the well-studied “standard” random walk, and is useful in modeling and understanding the Susceptible-Infected-Susceptible (SIS)-type of epidemic processes in networks. It can also be helpful in performing light-weight information dissemination in resource-constrained networks. A cobra walk is parameterized by a *branching factor* k . The process starts from an arbitrary vertex, which is labeled *active* for step 1. In each step of a cobra walk, each active vertex chooses k random neighbors to become active for the next step (“branching”). A vertex is active for step $t + 1$ only if it is chosen by an active vertex in step t (“coalescing”). This results in a stochastic process in the underlying network with properties that are quite different from both the standard random walk (which is equivalent to the cobra walk with branching factor 1) as well as other gossip-based rumor spreading mechanisms.

We focus on the *cover time* of the cobra walk, which is the number of steps for the walk to reach all the vertices, and derive almost-tight bounds for various graph classes. We show an $O(\log^2 n)$ high probability bound for the cover time of cobra walks on expanders, if either the expansion factor or the branching factor is sufficiently large; we also obtain an $O(\log n)$ high probability bound for the *partial cover time*, which is the number of steps needed for the walk to reach at least a constant fraction of the vertices. We also show that the cover time of the cobra walk is, with high probability, $O(n \log n)$ on any n -vertex tree for $k \geq 2$, $\tilde{O}(n^{1/d})$ on a d -dimensional grid for $k \geq 2$, and $O(\log n)$ on the complete graph.

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Theory of computation** → **Distributed algorithms**; **Random walks and Markov chains**; • **Computing methodologies** → **Self-organization**;

Additional Key Words and Phrases: Information spreading, cover time, epidemic processes

ACM Reference Format:

Chinmoy Dutta, Gopal Pandurangan, Rajmohan Rajaraman, and Scott Roche. 2015. Coalescing-branching random walks on graphs. *ACM Trans. Parallel Comput.* 2, 3, Article 20 (October 2015), 29 pages.

DOI: <http://dx.doi.org/10.1145/2817830>

Authors’ addresses: C. Dutta, *Twitter*, San Francisco, CA; email: chinmoy@twitter.com; G. Pandurangan, Department of Computer Science, University of Houston, Houston, TX 77204, USA. Work done while the author was affiliated with the Division of Mathematical Sciences, Nanyang Technological University, Singapore 637371 and Department of Computer Science, Brown University, Providence, RI 02912, USA. email: gopalpandurangan@gmail.com. Supported in part by the following grants: Nanyang Technological University grant M58110000, Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 grant MOE2010-T2-2-082, and the US-Israel Binational Science Foundation grant 2008348; R. Rajaraman and S. Roche, College of Computer and Information Science, Northeastern University, Boston MA 02115, USA; email: {rraj, str}@ccs.neu.edu. Supported in part by NSF grants CNS-0915985, CCF-1216038, and CCF-1422715.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2015 ACM 2329-4949/2015/10-ART20 \$15.00

DOI: <http://dx.doi.org/10.1145/2817830>

1. INTRODUCTION

We study a distributed propagation mechanism in networks called the *coalescing-branching random walk* (*cobra walk*, for short). A cobra walk is a variant of the standard random walk, and is parameterized by a *branching factor*, k . The process starts from an arbitrary vertex, which is initially labeled *active*. For instance, this could be a vertex that has a piece of data, rumor, or a virus. In a cobra walk, for each discrete time step, each active vertex chooses k random neighbors (sampled independently with replacement) to become active for the next step; this is the “branching” property, in which each vertex spawns multiple independent random walks. A vertex is active for step t if and only if it is chosen by an active vertex in step $t - 1$; this is the “coalescing” property (i.e., if multiple walks meet at a vertex, they coalesce into one walk).

A cobra walk generalizes the standard random walk [Lovász 1996; Mitzenmacher and Upfal 2005], which is equivalent to a cobra walk with $k = 1$. Random walks on graphs have a wide variety of applications, including serving as fundamental primitives in distributed network algorithms for load balancing, routing, information propagation, gossip, and search [Das Sarma et al. 2009, 2010; Bui et al. 2006; Zhong and Shen 2006]. Being local and requiring little state information, random walks and their variants are especially well-suited for self-organizing dynamic networks such as Internet overlay, ad hoc wireless, and sensor networks [Zhong and Shen 2006]. As a propagation mechanism, one parameter of interest is the *cover time*, the expected time it takes to cover all the vertices in a network. Since the cover time of the standard random walk can be large— $\Theta(n^3)$ in the worst case, $\Theta(n \log n)$ even for expanders [Lovász 1996]—some recent studies have studied simple adaptations of random walks that can speed up cover time [Adler et al. 2003; Berenbrink et al. 2010; Dimitrov and Plaxton 2005]. Our analysis of cobra walks continues this line of research, with the aim of studying a lightweight information dissemination process that has the potential to improve cover time significantly.

Our primary motivation for studying cobra walks is their close connection to Susceptible Infected Susceptible (SIS)-type epidemic processes in networks. The SIS model (e.g., Durrett [2010], also see Section 1.3) is widely used for capturing the spread of diseases in human contact networks or propagation of viruses in computer networks. Three basic properties of a SIS process are: (a) a vertex can infect one or more of its neighbors (the “branching” property), (b) a vertex can be infected by one or more of its neighbors (the “coalescence” property), and (c) an infected vertex can be cured and then become susceptible to infection at a later stage. Cobra walks satisfy all these properties, whereas standard random walks and other gossip-based propagation mechanisms violate one or more. Also, although there has been considerable work on the SIS model ([Ganesh et al. 2005; Van Mieghem 2011; Givan et al. 2011; Durrett 2010; Parshani et al. 2010; Draief and Ganesh 2011; Berger et al. 2005]), it has been analytically hard to tackle basic *coverage* questions: (1) How long will it take for the epidemic to infect, say, a constant fraction of network? (2) Will every vertex be infected at some point, and how long will this take? Our analysis of cobra walks in certain special graph classes is a step toward a better understanding of such questions for SIS-type processes.

1.1. Our Results and Techniques

We derive near-tight bounds on the cover time of cobra walks on trees, grids, and expanders. These special graph classes arise in many distributed network applications, especially in the modeling and construction of peer-to-peer (P2P), overlay, ad hoc, and sensor networks. For example, expanders have been used for modeling and construction of P2P and overlay networks, grids and related graphs have been used as models for ad

hoc and sensor networks, and spanning trees are often used as backbones for various information propagation tasks.

We begin with an observation that Matthew’s Theorem [Matthews 1988; Lovász 1996] for random walks extends to cobra walks; that is, the cover time of a cobra walk on an n -vertex graph is at most $O(\log n)$ times the maximum hitting time of a vertex. *Hitting time* is the expected time until a walk originating at $u \in V$ reaches $v \in V$ for the first time. For many graphs, the expected cover time of a random walk coincides with the high-probability cover time. This enables us to focus on deriving bounds for the hitting time.

We face two technical challenges in our analysis. First, unlike in a standard random walk, cobra walks have multiple “active” vertices at any step, and in almost all graphs it is difficult to characterize the distribution of the active vertices at any point of time. Second, the combination of the branching and coalescing properties introduces a nontrivial dependence among the active vertices, making it challenging to quantify the probability that a given vertex is made active during a given time period. Surprisingly, these challenges manifest even in tree networks. We present a result that gives a tight bound on the cover time for trees, which we obtain by establishing a recurrence relation for the expected time taken for the cobra walk to cross an edge along a given path of the tree.

—For an arbitrary n -vertex tree, a cobra walk with $k \geq 2$ covers all vertices in $O(n \log n)$ steps with high probability¹ (c.f. Theorem 3.2 of Section 3).

For a matching lower bound, we note that the cover time of a cobra walk in a star graph is $\Omega(n \log n)$ with high probability. We conjecture that the cover time for *any* n -vertex graph is $O(n \log n)$. By exploiting the regular structure of a grid, we establish improved and near-tight bounds for the cover time on d -dimensional grids.

—For a d -dimensional grid, we show that a cobra walk with $k \geq 2$ takes $\tilde{O}(n^{1/d})$ steps, w.h.p. (cf. Theorem 4.1 of Section 4).

We next show that the cover time of a cobra walk on the complete graph is logarithmic in n . Although this result may not be particularly surprising, the method of proof is independently of interest and serves as a “warm-up” and contrast to the proof of our result.

—For K_n , the complete graph on n vertices, w.h.p. a cobra walk covers K_n in $O(\log n)$ time (cf. Theorem 5.1 of Section 5).

Our main technical result is an analysis of cobra walks on expanders, which are graphs in which every set S of vertices of size at most half the number of vertices has at least $\alpha|S|$ neighbors for a constant α , which is referred to as the expansion factor.

—We show that for an n -vertex constant-degree expander, a cobra walk covers a constant fraction of vertices in $O(\log n)$ steps and all the vertices in $O(\log^2 n)$ steps with high probability, assuming that either the branching factor or the expansion factor is sufficiently large (cf. Theorems 6.2 and 6.3 of Section 6).

Our analysis for expanders proceeds in two phases. We show that in the first phase, which consists of $O(\log n)$ steps, the branching process dominates, resulting in an exponential growth in the number of active vertices until a constant fraction of vertices become active, with high probability. In the second phase, although a large fraction of the vertices continues to be active, dependencies caused by the coalescing property

¹By the term “with high probability, we mean with probability $1 - 1/n^c$, for some constant $c > 0$.

prevent us from treating the process as multiple independent random walks, analyzed in Alon et al. [2008] (or even d -wise independent walks for a suitably large d). We overcome this hurdle by carefully analyzing these dependencies and bounding relevant conditional probabilities to establish an $O(\log n)$ bound with high probability on the maximum hitting time, leading to an $O(\log^2 n)$ bound on the cover time.

For establishing our results, it is often convenient to work with branching factor $k = 2$. Since the cover time for larger values of the branching factor is at most the cover time for $k = 2$, all of our results hold for $k \geq 2$.

1.2. Related Work and Comparison

Branching and coalescing processes. There is a large body of work on branching processes (without coalescence) on various discrete and nondiscrete structures [Harris 1963; Madras and Schinazi 1992; Benjamini and Müller 2010]. A study of coalescing random walks (without branching) was performed in Cooper et al. [2012] with applications to voter models. Others have looked at processes that incorporate branching and coalescing particle systems [Arthreya and Swart 2005; Sun and Swart 2008]. However, these studies treat the particle systems as continuous-time systems, with branching, coalescing, and death rates on restricted-topology structures such as integer lattices. To the best of our knowledge, ours is the first work that studies random walks that branch and coalesce in discrete time and on various classes of nonregular finite graphs.

Random walks and parallel random walks. Feige [1993, 1995] showed that the cover time of a random walk on any undirected n -vertex connected graph is between $\Theta(n \log n)$ and $\Theta(n^3)$ with both the lower and upper bounds being achieved in certain graphs; in fact, the two bounds he established are tight to within lower order terms. With the rapidly increasing interest in information (rumor) spreading processes in large-scale networks and the gossiping paradigm (e.g., see [Chen and Pandurangan 2012] and the references therein), there have been a number of studies on speeding up the cover time of random walks on graphs. One of the earliest studies is due to Adler et al. [2003], who studied a process on the hypercube in which, in each round, a vertex is chosen uniformly at random and covered; if the chosen vertex was already covered, then an uncovered neighbor of the vertex is chosen uniformly at random and covered. For any d -regular graph, Dimitrov and Plaxton showed that a similar process achieves a cover time of $O(n + (n \log n)/d)$ [Dimitrov and Plaxton 2005]. For expander graphs, Berenbrink et al. showed a simple variant of the standard random walk that achieves a linear (i.e., $O(n)$) cover time [Berenbrink et al. 2010].

It is instructive to compare cobra walks with other mechanisms to speed up random walks, as well as with gossip-based rumor spreading mechanisms. Perhaps the most related mechanism is that of parallel random walks, first studied in Broder [1989] for the special case where the starting vertices are drawn from the stationary distribution and in Alon et al. [2008] for arbitrary starting vertices. Nearly tight results on the speedup of cover time as a function of the number of parallel walks have been obtained by Elsasser and Sauerwald [2011] for several graph classes including the cycle, d -dimensional meshes, hypercube, and expanders. (Also see Efremenko and Reingold [2009] for results on mixing time.) Although cobra walks are similar to parallel random walks in the sense that at any step multiple vertices may be selecting random neighbors, there are significant differences between the two mechanisms. First, the cover times of these walks are not comparable. For instance, whereas k parallel random walks may have a cover time of $\Omega(n^2 / \log k)$ for any $k \in [1, n]$ [Elsasser and Sauerwald 2011], a 2-branching cobra walk on a line has a cover time of $O(n)$. Second, although the number of active vertices in k parallel random walks is always k , the number of

active vertices in any k -branching cobra walk is continually changing and *may not even be monotonic*. Most importantly, the analysis of cover time of cobra walks needs to address several dependencies in the process by which the set of active vertices evolve; we use the machinery of Markov chains on graph tensor products to obtain the cover time bound for bounded-degree expanders (see Section 6).

The works of Das Sarma et al. [2009, 2010] presented distributed algorithms for performing a standard random walk in sublinear time (i.e., in time sublinear in the length of the walk). In particular, the algorithm of Das Sarma et al. [2010] performs a random walk of length ℓ in $\tilde{O}(\sqrt{\ell D})$ rounds with high probability on an undirected network, where D is the diameter of the network. The high-level idea behind this algorithm is to perform several short walks in parallel and then stitch them carefully. However, this speed-up comes with a drawback: The message complexity of the faster algorithm is much worse compared to the naive sequential walk, which takes only ℓ messages. In contrast, we note that the speed-up in cover time given by a cobra walk over the standard random walk comes only at the cost of a slightly worse message complexity.

Gossip-based mechanisms. Gossip-based information propagation mechanisms have also been used for information (rumor) spreading in distributed networks.² Gossip-based algorithms have also been used successfully to *design* efficient distributed algorithms for a variety of problems in networks, such as information dissemination, aggregate computation, constructing overlay topologies (e.g., see Chen and Pandurangan [2012] and the references therein). In the most typical rumor-spreading models, gossip involves either a push step, in which vertices that are aware of a piece of information (being disseminated) pass it to random neighbors, or a pull step, in which vertices that are unaware of the information attempt to extract the information from one of their randomly chosen neighbors, or some combination of the two. In such models, the knowledgeable vertices or the ignorant vertices participate in the dissemination problem in *every* round (step) of the algorithm. The main parameter of interest in many of these analyses is the number of rounds needed until all the vertices in the network get to know the information.

The rumor-spreading mechanism that is most closely related to cobra walks is the basic push protocol, in which in every step, every informed vertex selects a random neighbor and pushes the information to the neighbor, thus making it informed. (In the push-pull version, unlike cobra, a vertex can choose a random neighbor and can get information.) Feige et al. [1990] show that the push process completes in every undirected graph in $O(n \log n)$ steps, with high probability. This paper also presented optimal upper bounds of the push process in various graph classes, including random graphs, bounded degree graphs, and the hypercube. Since then, the push protocol and its variants (in particular, the push-pull protocol) have been extensively analyzed both for special graphs, as well as for general graphs in terms of their expansion properties (see, e.g., Chierichetti et al. [2010a, 2010b, 2011], Giakkoupis and Sauerwald [2012], Giakkoupis [2011], Fountoulakis et al. [2012], and Fountoulakis and Panagiotou [2010]).

Although cobra walks and push-based rumor spreading share the property that *multiple* vertices are active in a given step (unlike the case in a standard random walk), the two mechanisms differ significantly. While the set of active vertices in rumor spreading is monotonically nondecreasing, this is not so in cobra walks, an aspect that makes the analysis challenging especially with regard to full coverage. (Note that in a push

²Sometimes, in the literature, “gossiping” has been used for *all-to-all* communication and “broadcasting” or “rumor spreading” for *one-to-all* communication.

process, once a node is active it remains active, unlike in cobra.) However, we note that in any graph, the (expected) cover time of the push process is no worse than k (where k is the branching factor) times the cover time of the cobra walk process. This can be easily established by simulating one step of the cobra walk by k (independent) steps of the push process. Thus, at least for constant k , the cover time of push is asymptotically no worse than that of cobra walk. However, the message complexity of the push protocol can be substantially different from that of cobra. A simple example is the star network, which the push protocol covers in $\Theta(n \log n)$ steps with a message complexity of $\Theta(n^2 \log n)$, whereas the 2-branching cobra walk has both cover time and message complexity $\Theta(n \log n)$. This can be extended to show similar results for star-based networks that have been proposed as models for Internet-scale networks [Comellas and Gago 2005] and the references therein.

1.3. Applications

As mentioned at the outset, cobra walks are closely related to the SIS model in epidemics, but they may be easier to analyze using tools from random walk and Markov chain analyses. Whereas the persistence time and epidemic density of SIS-type epidemic models are well studied [Ganesh et al. 2005; Kessler 2008; Van Mieghem 2011], to the best of our knowledge, the time needed for a SIS-type process to affect a large fraction (or the whole) of the network has not been well-studied. The SIS model considered in these studies is typically in a continuous time setting. For example, the work of Ganesh et al. [2005] considers a model in which, at any time, infected vertices infect their neighbors with rate β , and vertices, once infected, recover at rate δ (δ is set to 1 without loss of generality). It is important to note that this defines a Markov process in which the absorbing state 0 can be reached from any starting state; thus, an epidemic always dies out. The main result of Ganesh et al. [2005] is that if the ratio β/δ is less than $1/\lambda_{max}$, the largest eigenvalue of the adjacency matrix of the underlying graph, then the epidemic dies out quickly (i.e., in $O(\log n)$ time). On the other hand, if this ratio is larger than the isoperimetric constant, then the epidemic will last for a long time (i.e., at least $\Omega(e^{n^\alpha})$). A cobra walk can be considered a discrete time variant of the continuous SIS model, with a difference. In a cobra walk, the epidemic does not die out because there is at least one vertex that remains active in the network. Thus, while it is not interesting to study the time to extinction in a cobra walk, it is relevant to study how long it takes to infect the whole or a fraction of the network; the amount of infected vertices in steady state (if it exists) is also worth studying. Our results and analyses of cobra walks can be generalized to understand the time taken for an epidemic process in an SIS-type model to spread in a network. By varying the branching factor and the time that a vertex remains infected, the process can also be viewed as a generalized rumor-spreading model, with applications in both epidemiology and information dissemination.

Cobra walks can also serve as a lightweight information dissemination protocol in networks, similar to the push protocol. As pointed out earlier, in certain types of networks, the message complexity incurred by a cobra walk to cover a network can be smaller than that for the push protocol. This can be useful, especially in infrastructure-less anonymous networks, where vertices don't have unique identities and may not even know the number of neighbors. In such networks, it is difficult to detect locally when coverage is completed.³ If vertices have a good upper bound on n (the network size), however, then vertices can terminate the protocol after a number of steps equal to

³In networks with identities and knowledge of neighbors, a vertex can locally stop sending messages when all neighbors have the rumor. This reduces the overall message complexity until cover time.

the estimated cover time. In such a scenario, message complexity is also an important performance criterion.

2. PRELIMINARIES

Let G be a connected graph with vertex set V and edge set E , and let $|V| = n$. We define a coalescing-branching (cobra) random walk on G with branching factor k starting at some arbitrary $v \in V$ as follows: At time $t = 0$, we place a pebble at v . Then, in the next and every subsequent time step, every pebble in G clones itself $k - 1$ times (so that there are now k indistinguishable pebbles at each vertex that originally had a pebble). Each pebble independently selects a neighbor of its current vertex uniformly at random and moves to it. Once all pebbles make their one-hop moves, if two or more pebbles are at the same vertex, they coalesce into a single pebble, and the next round begins. In a cobra-walk, a vertex may receive a pebble an arbitrary number of times.

For a time step t of the process, let S_t be the **active set**, the set of all vertices of G that have a pebble. We will use two different definitions of the neighborhood of S_t : Let $N(S_t)$ be the **inclusive** neighborhood, the union of the set of neighbors of all vertices in S_t (which can include members of S_t itself). Let $\Gamma(S_t)$ be the **noninclusive neighborhood**, which is the union of the set of neighbors of all vertices of S_t such that $S_t \cap \Gamma(S_t) = \emptyset$.

Let the expected **maximum hitting time** h_{max} of a cobra walk on G be defined as the $\max_{u,v \in V} \mathbb{E}[h_{u,v}]$, where $h_{u,v}$ is the time it takes for the first pebble arising from a cobra walk starting at vertex u to first reach v .

We are interested in two different notions of cover time, which we define as the time until all vertices of G have been visited by a cobra walk at least once. Let τ_v be the minimum time t such that, for a cobra walk starting from v , $\forall u \in V - v$, $u \in S_t$ for some $t \leq \tau_v$ that may depend on u . Then we define the **cover time** of a cobra walk on G to be $\max_{v \in V} \tau_v$. We define the **expected cover time** to be $\max_{v \in V} \mathbb{E}[\tau_v]$. Note that in the literature for simple random walks, cover time usually refers to the expected cover time. In this article, we will show high-probability bounds on the cover time.

In Section 6, we prove results for cobrawalks on expanders. In this article, we use a spectral definition for expanders and then use Tanner's theorem to translate that to neighborhood and cut-based notions of expanders.

Definition 2.1. A **d -regular ϵ -expander** is a d -regular graph whose adjacency matrix has eigenvalues α_i such that $|\alpha_i| \leq \epsilon d$ for $i \geq 2$, where $\epsilon \in (0, 1)$.

It is known that any d -regular ϵ -expander G with n vertices approximates the complete graph $H = K_n$, with weight d/n on each edge, in the following sense: For all $x \in \mathbb{R}^n$, $(1 - \epsilon)x^T L_H x \leq x^T L_G x \leq (1 + \epsilon)x^T L_H x$, where L is the Laplacian matrix associated with the adjacency matrix of the corresponding graph [Spieman 2012]. Tanner's theorem [Tanner 1984] then implies the following lower bound on neighborhood size in any d -regular ϵ -expander. Note that the δ defined in the theorem may be *any* value, not just a constant.

THEOREM 2.2 [TANNER 1984]. *Let G be a d -regular graph ϵ -expander. For all $\delta > 0$ and $S \subseteq V$ such that $|S| = \delta n$, we have $|N(S)| \geq \frac{|S|}{\epsilon^2(1-\delta)+\delta}$.*

In the analysis of random walks it is often important to qualify whether a graph is non-bipartite. In this analysis, all expander graphs are non-bipartite. This is implicit

in the definition of the eigenvalues of the adjacency matrix of G , which require $|\alpha_i| \leq \epsilon d < d$.

3. TREES

A useful tool in bounding the cover time for simple random walks is Matthew's theorem [Matthews 1988; Lovász 1996], which bounds the expected cover time of a graph by the maximum expected hitting time $h_{u,v}$ between any two vertices u and v multiplied by a $O(\log n)$ factor. Here, we show that this result can be extended to cobra walks.

THEOREM 3.1 (MATTHEW'S THEOREM FOR COBRA WALKS). *Let G be a connected graph on n vertices. Let W be a cobra walk on G starting at an arbitrary vertex. Then, the cover-time of W on G , $C(G)$, is bounded from above by $O(h_{\max} \log n)$ in expectation and with high probability.*

PROOF. We follow the proof outlined in Lovász [1996]. We first show the following claim.

CLAIM. Let β be the expected number of steps before a cobra walk visits more than half of the vertices. Then, $\beta \leq 2h_{\max}$.

PROOF OF CLAIM. Let ρ_v be the time when vertex v is first visited by the cobra walk. Arrange the ρ_v 's in increasing order. Let $\ell = \lceil (n+1)/2 \rceil$. Then, the time β when the cobra walk reaches (for the first time) more than half of the vertices is the ℓ -th element in the above order of the ρ_v s. Hence,

$$\sum_{v \in V} \rho_v \geq (n - \ell + 1)\beta.$$

Taking expectation on both sides we have,

$$b = E[\beta] \leq \frac{1}{n - \ell + 1} \sum_{v \in V} E[\rho_v] \leq \frac{nh_{\max}}{n - \ell + 1} \leq 2h_{\max},$$

since $\sum_{v \in V} E[\rho_v] \leq nh_{\max}$ and $\frac{n}{n-\ell+1} \leq 2$. Hence the claim.

Our claim says that in $2h_{\max}$ steps, the cobra walk will visit more than $n/2$ vertices; by a similar argument, in another $2h_{\max}$ steps, the cobra walk will visit half of the rest of the vertices. Repeating the argument $\log_2 n$ times, it follows that all vertices will be visited in expected $O(h_{\max} \log n)$ steps.

We next show that the given bound on the cover time also holds with high probability. Let's partition the visit of the cobra walk into $\log_2 n$ different stages according to the number of vertices it visits: The first $n/2$ (different) vertices (first stage), then the next $n/4$ vertices (second stage), and so on. By our claim, each stage finishes in at most $2h_{\max}$ steps in expectation. By Markov's inequality, this implies that, with probability at least $1/2$, each stage finishes in at most $4h_{\max}$ steps. If the cobra walk finishes a stage in at most $4h_{\max}$ steps, then we call it a "success"; the probability of this happening is at least $1/2$. For the sake of analysis, assume that if the cobra walk fails to complete a particular stage in at most $4h_{\max}$ steps, then it repeats this stage again (this can only increase the cover time). Call each such repetition a trial. Accordingly, the cobra walk needs $\log_2 n$ successes to finish all stages. In a total of $12 \log_2 n$ trials, the expected number of successes is at least $6 \log_2 n$. Applying a Chernoff bound [Mitzenmacher and Upfal 2005], the probability that the number of successes in $12 \log_2 n$ trials is less than $\log_2 n$ is less than $1/n^2$. Thus, with high probability, the number of repetitions is bounded by $O(\log n)$ overall. Since each repetition takes $4h_{\max}$ steps, the cover time is $O(h_{\max} \log n)$ with high probability. \square

Matthew's theorem for cobra walks is used in proving the cover time for trees and grids.

THEOREM 3.2. *For any tree, the cover time of a cobra walk (with branching factor $k \geq 2$) starting from any vertex is $O(n \log n)$ with high probability.*

We prove our main result by calculating the maximum hitting time of a cobra walk on a tree T and then applying Matthew's theorem. Cobra walks on trees are especially tractable because they follow two nice properties. Since a tree has a unique path between any two vertices, when analyzing progression of the cobra walk from a source vertex to a target vertex, we only need keep track of the pebble closest to the target (which may change in each time step). In addition, the fact that there is one simple path between any two vertices limits the number of collisions we need to keep track of, a property that is not true for general graphs and makes cobra walks harder to analyze on them. For this section, we fix the branching factor $k = 2$. For $k > 2$ but still constant, the cover time would not be asymptotically better. This is because, as we will see, our analysis involves showing stochastic dominance of a biased random walk with transition properties similar to that of tokens moving from an active vertex in a cobra walk. As such, having a branching factor larger than 2 boosts the bias probabilities, but only by a constant. This therefore does not affect the asymptotic hitting time result and hence not the cover time.

The general idea behind the proof is as follows. We consider the longest path in the tree. Along each vertex in this path, except for the first and last, there will be a subtree rooted at that vertex. If a cobra walk's closest pebble to the endpoint is at vertex l , the walk from this point can either advance with at least one pebble, or it cannot advance by either backtracking along the path, going down the subtree rooted at l , or both. We show via a stochastic dominance argument that a biased random walk from l , whose transition probabilities are tuned to be identical to cobra walk's, will next advance to $l + 1$ in a time that is dominated primarily by the size of the subtree at l . This is done by analyzing the return times in the nonadvancement scenarios just listed. Thus, summing up over the entire walk, the hitting time is dominated by a linear function of the size of the entire tree.

In Lemma 3.3, we bound the return time of a cobra walk to the root of a tree.

LEMMA 3.3. *Let T be a tree of size n . Pick a root, r , and let r have d children. Then a cobra walk on T starting at r will have a return time to r of $4n/d$.*

PROOF. To show that the Lemma holds for a cobra walk, we will actually show that it holds for a biased simple random walk with transition probabilities modified to resemble those of a cobra walk. We then use a stochastic dominance argument to show that the return time of the cobra walk dominates the return time of the biased simple random walk.

For this simple random walk, we start at r and assume that r has d children. In the first step, the walk picks one of the children of r , r_i . Let $(d_i + 1)$ be the degree of r_i (meaning that r_i has one parent, r , and d_i children). Then we define transition probabilities as follows: p is the probability of returning to r in the next step, and q is the probability of continuing down the tree to a child of r_i . They are given as: $p = (1 - (\frac{d_i}{d_i+1})^2)$, $q = (\frac{d_i}{d_i+1})^2$, $\frac{q}{p} = \frac{(d_i)^2}{(2d_i+1)}$. Note that these are the exact same probabilities that a cobra walk at vertex r_i would have for sending (not sending) at least one (any) pebbles back to the root.

The rest of the proof follows by mathematical induction. Consider a tree T that has only two levels. Starting from r , the return time, 2, is constant, thus the relationship holds. For the inductive case, assume that the hypothesis holds. Denote $ret(T)$ the

return time to the root of t and $N(v)$ to be the children of v in the rooted tree T . For a root r with children r_i , denote T_i to be the induced subtree rooted at r_i . Then:

$$\begin{aligned}
 \mathbb{E}[\text{ret}(T)] &= 1 + \frac{1}{d} \sum_{r_i \in N(r)} 1 + \mathbb{E}[\text{number of visits to } r_i \text{ until walk returns to } r] \cdot \mathbb{E}[\text{ret}(T_i)] \\
 &\leq 2 + \frac{1}{d} \sum_{r_i \in N(r)} \frac{q}{p} \cdot \frac{c|T_i|}{d_i} \\
 &= 2 + \frac{c}{d} \sum \frac{d_i^2}{2d_i + 1} \cdot \frac{|T_i|}{d_i} \\
 &= 2 + \frac{c}{d} \sum \frac{d_i}{2d_i + 1} |T_i| \\
 &\leq 2 + \frac{c}{2d}(n-1) < \frac{4n}{d}
 \end{aligned}$$

for $c = 4$, where $\text{ret}(T_i) \leq 4|T_i|/d_i$ by the induction hypothesis. Note that the q/p that appears in the second line of the equations represents the mean of a negative binomial random variable describing the number of heads needed in a series of coin tosses until the first tails is observed. Here, heads indicates **not** moving one hop up the tree toward the root, and tails is equivalent to doing so.

Next, we show the stochastic dominance of the return time of the biased random walk to the root over the return time of a cobra walk to the root. Let X be the random variable representing the number of time steps until a biased (single) random walk starting at r returns to r . Let Y be the random variable associated with the return time of a cobra walk starting at r to r . Note that X and Y are defined on the same probability space Ω ; namely, the integers ≥ 2 .

For $X, Y = 2$ note that $\Pr[X = 2] \leq \Pr[Y = 2]$, since the random walk will occupy only one child of r whereas the cobra walk will occupy one or two children. If it occupies two children, the probability that at least one token returns to r is greater than in the single-vertex case. Similarly, for higher values of X, Y , the vertex occupied by the random walk token will always be a subset of the vertices occupied by the tokens closest to r in the cobra walk. As such, the probability of the cobra walk moving one step closer to return will be at least the probability of the random walk's token doing so, and applying this recursively we have $\Pr[X \geq x] \geq \Pr[Y \geq x]$. (Note that x can only take on even values, as T is a bipartite graph). \square

Next, we show an upper bound on the expected amount of time it will take until a cobra walk moves one vertex closer toward the target vertex along the path from the source to the target.

LEMMA 3.4. *Fix a path in a tree T made up of vertices $1, \dots, l, (l+1), \dots, t$. Then, the expected time it takes for a cobra walk starting at vertex l to get to $l+1$ with at least one token can be bounded as:*

$$h_{l,(l+1)} \leq \frac{3}{2} + \frac{12}{5} \sum_{i=l}^2 \left(\frac{1}{5}\right)^{l-i} |T_i|, \quad (1)$$

where T_l is the induced subtree formed by vertex l and its neighbors not in $\{l-1, l+1\}$ and all of their respective descendants.

Informally, we prove that the one-step hitting time is bounded from above by the expected hitting time of the worst-case scenario that either both pebbles go back along

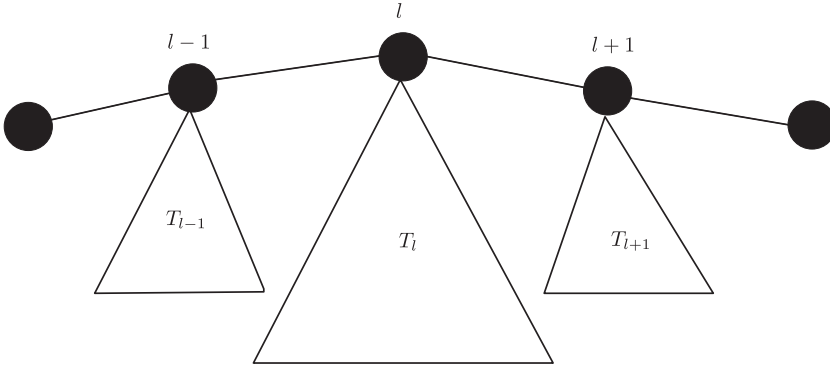


Fig. 1. Local topology of tree for Lemma 3.4.

the path or both go down the subtree rooted at l . We then establish a simple recurrence relation.

PROOF. Vertex l has one edge to the vertex $l - 1$, one edge to vertex $l + 1$, and d_l additional neighbors. T_l is the induced subtree of T formed by l , the d_l neighbors of l that are not in $\{l - 1, l + 1\}$, and all other vertices connected to the d_l neighbors of l .

- Probability of a pebble going from l to $l + 1 = p = (1 - \frac{(d_l+1)}{(d_l+2)})^2$.
- Probability of a pebble not going from l to $l + 1 = 1 - p = q$.
- Probability of a cobra walk sending both pebbles from l to $l - 1$ conditioned on it not sending any pebbles from l to $l + 1 = q'_l = (\frac{1}{(d_l+1)^2})$.
- Probability of a cobra walk sending at least one pebble to the subtree T_l conditioned on its not sending any pebbles to $l + 1 = q''_l = (\frac{(d_l)}{(d_l+1)})^2 + 2(\frac{d_l}{(d_l+1)^2}) = \frac{d_l^2 + 2d_l}{(d_l+1)^2}$.

Note that, conditioned on a pebble not advancing to vertex $l + 1$, there are three disjoint events:

- (A) Both pebbles go to $l - 1$,
- (B) one pebble goes to $l - 1$ and one pebble goes into subtree T_l , and
- (C) both pebbles go into T_l .

We define an alternate event B' as the event that there is only one pebble at l , and it goes to a child in T_l . Therefore, it is not technically in the space of cobra walk actions. However, this modified cobra walk stochastically dominates the original cobra walk since it corresponds to a subset of the possible cobra walks actions from l : The case where both pebbles move to the same vertex in T_l . If we let R be the time until first return of the cobra walk to l conditioned on no pebble going to $l + 1$ from l initially, we wish to show that $E[R|B] \leq E[R|B']$ and that $E[R|C] \leq E[R|B']$. What is the relationship between B and B' ? Consider two random variables, X and Y , and let X be the time until first return of a pebble that travels from l to $l - 1$, and let Y be the time until first return of a pebble that travels into T_l . Then $R|B$ is just another random variable, $U = \min(X, Y)$. Since $U \leq Y$ over the entire space, $E[U] \leq E[Y]$, and clearly $R|B'$ is equivalent to Y . Thus, $E[R|B] \leq E[R|B']$. It is also easy to see that $E[R|B'] \geq E[R|C]$. Thus, by the law of total expectation we have:

$$\begin{aligned}
 E[R] &= E[R|A] \Pr(A) + E[R|B] \Pr(B) + E[R|C] \Pr(C) \\
 &\leq E[R|A] \Pr(A) + (\Pr(B) + \Pr(C))E[R|B'] \\
 &= E[R|A] \Pr(A) + (1 - \Pr(A))E[R|B'].
 \end{aligned}$$

Then, the hitting time can be expressed as:

$$\begin{aligned} h_{l,l+1} &\leq p + q(E[R] + h_{l,l+1}) \\ \Rightarrow (1 - q)h_{l,l+1} &\leq p + q(E[R]) \\ \Rightarrow h_{l,l+1} &\leq 1 + \frac{q}{p}(q'_l(1 + h_{l-1,l}) + q''_l r(T_l)). \end{aligned}$$

Note that $q/p = \frac{(d_l+1)^2}{(2d_l+3)}$. Since $r(T_l) \leq 4|T_l|/d_l$ by Lemma 3.3, we continue with:

$$\begin{aligned} h_{l,l+1} &\leq 1 + \frac{(d_l+1)^2}{(2d_l+3)} \frac{1}{(d_l+1)^2} (1 + h_{l-1,l}) + \frac{(d_l+1)^2}{(2d_l+3)} \frac{(d_l^2 + 2d_l)}{(d_l+1)^2} \frac{4|T_l|}{d_l} \\ &\leq 1 + \frac{1}{5}(1 + h_{l-1,l}) + \frac{12}{5}|T_l| \\ &= \frac{6}{5} + \frac{12}{5}|T_l| + \frac{h_{l-1,l}}{5}, \end{aligned}$$

for $d_l \geq 1$.

We next apply this formula to $h_{l-1,l}$ and continue the recurrence to get the bound in its expanded form:

$$h_{l,l+1} \leq \frac{6}{5} \sum_{i=0}^l \left(\frac{1}{5}\right)^i + \frac{12}{5} \left(|T_l| + \left(\frac{1}{5}\right) |T_{l-1}| + \left(\frac{1}{5}\right)^2 |T_{l-2}| + \dots + \left(\frac{1}{5}\right)^{l-2} |T_2| \right),$$

where we use the fact that $h_{0,1} \leq 1$. We simplify the equation to obtain the desired result.

$$h_{l,l+1} \leq \frac{3}{2} + \frac{12}{5} \sum_{i=l}^2 \left(\frac{1}{5}\right)^{l-i} |T_i|. \quad \square$$

Note that we stop at T_2 , since for vertex 1 on the path, having a tree rooted at it would violate the assumption that $1, \dots, l$ is the path with maximal hitting time.

We are finally ready to prove our main result for the tree, Theorem 3.2, that the cobra walk cover time of an arbitrary tree occurs in $O(n \ln n)$ steps.

PROOF. By Matthew's theorem for cobra walks, $C(G) \leq O(\log n)h_{max}$. We just need to prove that h_{max} is at most linear in the size of the tree.

Let P be the path for which $h_{u,v}$ is maximized, and let the path consist of the sequence of vertices $1, 2, \dots, t$. As in the proof of the single-step hitting time, we note that for all but the first and last vertices on P there is a subtree T_l of size $|T_l|$ rooted at each vertices. Because $h_{1,t} \leq h_{1,2} + h_{2,3} + \dots + h_{l-1,t}$, we obtain the desired result from Lemma 3.4 as follows:

$$\begin{aligned} h_{1,t} &\leq \frac{3}{2}t + \frac{12}{5} \sum_{j=2}^{t-1} \left[|T_j| \sum_{i=0}^{\infty} \left(\frac{1}{5}\right)^i \right] \\ &\leq \frac{3}{2}t + \frac{12}{5} \frac{5}{4} \sum_{j=2}^{t-1} |T_j| \leq \frac{9n}{2}. \quad \square \end{aligned}$$

4. COVER TIME FOR GRIDS

To show a cover time bound for the d -dimensional grid (in which each dimension of the grid is of length $n^{1/d}$ and d is a constant), we also apply the technique of finding the maximum of the expected hitting times between any vertices $u, v \in G$ and then

applying Matthew's bound. However, unlike the tree, we do not simply consider the pebble closest to the "target" vertex. Rather, we focus on making progress one dimension at a time toward the target (with regard to the coordinate of that dimension). Within this dimension, we only follow the pebble closest to the target and show that it is making a biased random walk toward its goal. Simultaneously, with regard to other dimensions, it is making an unbiased random walk and thus remains close to its starting position in that coordinate. Repeating this $\log \log n$ times for each dimension, we show that it takes $O(n^{1/d} \log^c n)$ time in expectation for a cobra walk starting at any vertex to first reach any other vertex in the grid. Applying Matthew's bound and multiplying by another factor of $\log n$ gets us coverage in $\tilde{O}(n^{1/d})$ time. We show the following theorem for a cobra walk with branching factor $k = 2$; clearly, the theorem holds for any $k \geq 2$.

THEOREM 4.1. *Let G be a finite d -dimensional grid on n nodes for some constant d , without wrap-around edges. Then, the cover time of a cobra walk on G is $O(n^{1/d} \log^{c+1} n)$ with high probability for branching factor $k = 2$ and some constant $c \in \Theta(1)$.*

PROOF. We want to calculate how long it takes, in expectation, for a cobra walk that starts at vertex $s = (0, \dots, 0)$ to have at least one of its pebbles reach vertex $\omega = (n^{1/d}, \dots, n^{1/d})$. Unlike in the case of the cobra walk on the tree, we do not focus our attention on the pebble (out of all pebbles) closest to ω , since there are multiple paths between s and ω . However, we do something similar: Imagine focusing on the progress of the cobra walk with regard to only one of the dimensions of the grid, say dimension i . As the cobra walk progresses, we look at the pebbles' i -th coordinates and keep track of the pebble closest to ω 's i -th coordinate. We call this pebble the *lead pebble*. We pick the lead pebble in each step in a manner similar to how we pick it for the tree. Suppose the lead pebble is at vertex v , whose i -th coordinate is y . In the next (branching) step of the cobra walk, if at least one of the lead pebble's offspring moves to $y + 1$, we make this pebble the lead pebble. If both of the offspring pebbles move to $y - 1$ (back toward s with regard to this dimension), then we place the lead pebble at the adjacent vertex to v with the $y - 1$ coordinate. Finally, if neither one of these two events occur, this means both pebbles have moved laterally in the grid (and thus make no progress backward or forward in the i -th coordinate). We flip a fair coin and pick one of the two pebbles as lead. It is worth pointing out for use later that the edge this pebble chooses to move along is chosen uniformly at random from all possible edges orthogonal to dimension i .

Returning to our focus on the lead pebble's progress on dimension i , it should be clear that what we have described here is our lead pebble taking a biased random walk when we focus solely on the changes in the i -th coordinate (i.e., we are projecting onto a line). It has the following transition probabilities: the probability p_+ of a $+1$ motion is given as $p_+ = 1 - \frac{(2d-1)^2}{2d} = \frac{4d-1}{4d^2}$, the probability p_- of a -1 motion is $1/4d^2$, and the probability p_0 of staying in place is the remainder of the probability mass: $p_0 = \frac{d-1}{d}$. This projection as a biased random walk on the line will later allow us to use a concentration bound to calculate the probability of the walk making a certain amount of progress for a walk of a given length.

Although we are following the lead pebble's movement along its i -th coordinate, it is of course moving around the grid along its other coordinates. However, as noted earlier, if the lead pebble moves along an edge orthogonal to dimension i , that edge is chosen uniformly at random from the set of all orthogonal edges. Hence, with respect to every dimension $j \neq i$, the lead pebble is making a (lazy) unbiased walk. Thus, we can also use a concentration bound to show that the lead pebble's j -th coordinate does not drift very far, for all other dimensions j . This conveniently will allow us to focus on the lead pebble's progress one dimension at a time and show that it makes progress toward the target in that dimension while remaining (relatively) still in the other dimensions

and, more importantly, not drifting very far backward and thus undoing any progress it might have made earlier.

More formally, we break our analysis into segments. In each segment, our goal is to have the cobra walk starting at s reach ω and do so within $O(n^{1/d})$ steps. We will show that, in any segment, this will happen with probability $q \in \Omega(1/\log^c n)$ for some constant c . We can think of each segment as being an independent trial, with probability q of success. Thus, in expectation, after $O(\log^c n)$ trials, one success will have occurred. This gives us a maximum hitting time of a cobra walk of $O(n^{1/d} \log^c n)$. Applying Matthews's bound gives us a cover time of $O(n^{1/d} \log^{c+1} n) = \tilde{O}(n^{1/d})$ for the d -grid.

We now describe what happens in a segment. Each segment is divided into a number of phases. The first phase is of length $O(dn^{1/d})$, the second phase $O(dn^{1/2d})$, and in general the i -th phase is of length $O(dn^{1/d2^{i-1}})$. All phases have identical structure (except for their duration) except for the last phase, which is treated separately and is of constant duration. Within each phase, there are d sub-phases, one for each dimension of the grid. So for some phase M of length $O(dD)$ (where $D = n^{1/d2^{j-1}}$, for the j th phase), label the sub-phases M_1, M_2, \dots, M_d . In sub-phase M_i , we allow the lead pebble selection to be governed by the rules of advancement for dimension i as described earlier. For the other $d - 1$ sub-phases, when the lead pebble selection is governed by a dimension other than i , the lead pebble's i -th coordinate is taking an unbiased random walk.

We now directly bound the probability p_1 , that for each sub-phase M_i of duration $O(dD)$ the lead pebble moves toward ω in the i -th coordinate by at least $D - 10d\sqrt{D}/2$, and p_2 , the probability that for the other $d - 1$ sub-phases the lead pebble's i -th coordinate does not drift by more than $\sqrt{D}/2$ steps.

Bound for p_1 . Let the sub-phase M_i last for K independent random walk steps. Let the result of each step t be a random variable X_t that takes on value 1 if a forward step is made, -1 if a backward step is made, and 0 if it stays in place. Then $\Pr[X_t = 1] = p_+$, $\Pr[X_t = -1] = p_-$, and $\Pr[X_t = 0] = p_0$ as defined earlier. Let $X = \sum_{t=1}^K X_t$. Since $X_t \geq -1$, we can use the following version of the Chernoff bound (see Theorems 2.8 and 2.9 in Chung et al. [2006]):

$$\Pr[X \leq E[X] - \lambda] \leq e^{\frac{-\lambda^2}{2(|X|^2 + \lambda/3)}}, \quad (2)$$

where $\|X\| = \sqrt{\sum_{t=1}^K E[X_t^2]}$. It is easy to verify that $E[X_t^2] = 1/d$ and that $E[X_t] = (2d - 1)/2d^2$. We want to make expected progress of D steps in the i -th coordinate. Setting $E[X] = D$ requires $K = \frac{2d^2}{2d-1}D$. Note that $\|X\|^2 = K/d$. To achieve our desired bound we set $\lambda = 10d\sqrt{D}/2$. Thus:

$$\begin{aligned} \Pr[X \leq E[X] - 10d\sqrt{D}/2] &\leq e^{\frac{-100d^2D}{8(K/d + \sqrt{D}/6)}} \\ &= e^{\frac{-600d^2D}{48K + 8d\sqrt{D}}} \\ &= e^{\frac{-600d^2D}{48D2d^2/(2d-1) + 8d\sqrt{D}}} \\ &= e^{\frac{-600dD}{96dD/(2d-1) + 8\sqrt{D}}} \\ &\leq e^{-4d} = 1 - p_1 \end{aligned}$$

Bound for p_2 . Recall that in sub-phase M_i , the lead pebble is taking an unbiased lazy random walk with regard to all other coordinates besides i . Rather than calculate the exact probabilities of $+1$, -1 , 0 movements of the walk when projected onto dimension j , we note that such a lazy unbiased walk is stochastically dominated by an unbiased, non-lazy random walk. Thus, we will calculate a concentration bound on this walk instead.

Let X_t take on the value 1 if a $+1$ step is made at time t , and value -1 if a step is made in the opposite direction. Then, we analyze this walk over $(d-1)K = (d-1)2d^2/(2d-1) \cdot D$ steps, and we would like to show that:

$$\Pr[X \leq E[X] - 10d^{1.5}\sqrt{D}/2] \leq (1 - p_2).$$

Here, of course, $X = \sum_{t=1}^{(d-1)K} X_t$. Thus $E[X] = 0$, $E[X_t^2] = 1$, and $\|X\|^2 = (d-1)\frac{2d^2}{(2d-1)}D$. Then, we can calculate:

$$\begin{aligned} \Pr[X \leq -10d^3\sqrt{D}/2] &\leq e^{-\frac{100d^3D}{8\left(\frac{(d-1)2d^2}{(2d-1)}D + \frac{\sqrt{D}}{6}\right)}} \\ &\leq e^{-\frac{100d^3}{8\left(\frac{(d-1)2d^2}{(2d-1)} + \frac{1}{6}\right)}} \leq e^{-4d} = (1 - p_2). \end{aligned}$$

Thus, during phase M , the probability that the lead pebble's i -th coordinate moves "right" by at least $D - O(\sqrt{D})$ steps is lower bounded (via union bound) by $p_1 + p_2 \geq 1 - 2e^{-4d}$. The probability that this happens for all dimensions is lower bounded (via union bound over all dimensions) by $1 - 2de^{-4d} = \Theta(1)$. Recall that each phase runs for a length that is the square root of the phase before it. By these calculations, with each phase, we grow progressively closer (within another square root factor) to ω with some constant probability. Indeed, after $O(\log \log n)$ phases, we will be within a constant distance from ω in each coordinate dimension, with probability at least $(\Theta(1))^{d \log \log n}$. If we come within a constant distance, then for the last phase, with some constant probability, the remaining distance will be covered in a constant number of steps, and the lead pebble will arrive at ω . Thus, each phase succeeds with probability in $\Omega(1/\log^{\Theta(1)} n)$, and the result follows. \square

For the special case where $d = 1$ (i.e., the line), we note that the expected cover time is $O(n)$, since by starting the cobra walk at 0 and only keeping track of the pebble closest to vertex n , we have a biased random walk with probability $3/4$ of advancing toward n . This will reach n in a linear number of steps. Repeating $\log n$ times would give us a high-probability bound.

5. CLIQUES

In this section, we analyze the cover time of a cobra walk on the complete graph of n vertices, K_n . Although the result ($O(\log n)$ cover time with high probability) is not surprising and the method of proof somewhat basic, we present it here for several reasons. First, it serves to illustrate a different method of proving results about cobra walks. Previously, for the case of the grid and the tree, the backbone of the analysis involved analyzing the progress of the closest token to the target of the longest path in the graphs. Other tokens were generally ignored. In this section, and particularly in the section dealing with expanders, the structure of the graph loses its simplicity and we have to keep track, in some way, of all of the tokens. We do this by studying the

active set at each time t in the progression of the cobra walk, where by active set S_t we mean the set of vertices of G at time t where a token from the previous round has landed. The second and third reasons for studying cobra walks on K_n are related: As an easier illustration of a new proof technique, it serves as a warm-up to the analysis of expanders and yet also serves as an illustrative contrast. We will point out the various places where the assumptions that can be made for K_n do not hold for expanders and thus necessitate the development of more advanced techniques.

We state the main result for K_n :

THEOREM 5.1. *Let $G = K_n$, the complete graph on n vertices. Then, a cobra walk (with branching factor $k = 2$) starting from any vertex in K_n will cover the entire graph with high probability in $O(\log n)$ time.*

The key to understanding the coverage of K_n is to understand what happens at an arbitrary step t , given that the active set is already of size $s = |S_t|$. For simplicity, assume that every vertex in K_n also has a self-loop. Let us also assume that $|S_t| \ll n$. According to the rules of the cobra walk, each active vertex will hold two pebbles at the beginning of the round. Each one of those pebbles will independently choose one of the n vertices of K_n to move to in the next step. Thus, the cardinality of S_{t+1} , the active set at the beginning of round $t + 1$ can range from 1, in the unlikely event that every single token chose to converge on the same vertex, to a maximum size $2s$, corresponding to the event where every token independently chooses a unique vertex of K_n .

We show that when the size of the active set is of no more than δn , for a constant $\delta < 1$ to be calculated later, with high probability the active set will grow by at least a constant factor $(1 + c)$ in each round up until it hits size δn . Once it reaches this active set size δn , it will not go below this size again, and the rest of the graph will be covered in $O(\log n)$ steps through random sampling.

LEMMA 5.2. *Let a cobra walk on K_n at time t have active set S_t , with cardinality $|S_t|$, and let $C = 10 < |S_t| \leq \delta n$. Then, with probability at least $1 - 1/n^3$, active set $|S_{t+1}| \geq (1 + c)|S_t|$.*

PROOF. As mentioned earlier, in moving from step t to $t + 1$, each of the s vertices in the active set in time t gives rise to two tokens. Each of these tokens then makes an independent choice among the n vertices and moves to that vertex at time $t + 1$. Since our analysis is concerned with active sets, we want to find an expression for the probability that $|S_{t+1}| = M$ for any $M \in [1, 2s]$. This involves counting the size of the sample space subject to the requirement that the $2s$ tokens select M vertices. Hence, we have $M! \binom{2s}{M}$, where $\binom{2s}{M}$ is the Stirling number of the second kind, which counts the number of ways to place $2s$ unlabeled balls into M unlabeled bins. Here, our tokens are unlabeled but our bins (vertices) are not, and we multiply through by a $M!$ to account for the labeling. Therefore, the total probability that $2s$ tokens will choose M vertices is given by:

$$\Pr[|S_{t+1}| = M] = \binom{n}{M} \frac{M! \binom{2s}{M}}{n^{2s}}.$$

Taking advantage of the identity $M! \binom{n}{M} = M^n - \sum_{i=1}^{M-1} \frac{M!}{i!} \binom{n}{i}$, we can bound the probability of having an active set of size M as: $\Pr[|S_{t+1}| = M] < \binom{n}{M} \frac{M^{2s}}{n^{2s}}$. We noted earlier that $s \leq \delta n$, and for the following calculation, we want to calculate the probability when $M = (1 + c)s = \hat{s}$, for a constant $c \in (0, 1)$. Thus, we also require that $\hat{s} < n/2$. We have:

$$\Pr[|S_{t+1}| = \hat{s}] \leq \binom{n}{\hat{s}} \frac{\hat{s}^{2s}}{n^{2s}} \leq \frac{n^{\hat{s}}}{\hat{s}^{\hat{s}}} e^{\hat{s}} \frac{\hat{s}^{2s}}{n^{2s}} = n^{\hat{s}-2s} \hat{s}^{2s-\hat{s}} e^{\hat{s}}.$$

We next need to show that $\Pr[|S_{t+1}| = \hat{s}] < 1/n^4$ for all s, \hat{s} in the valid range. When $s, \hat{s} = (1+c)s$ are constants, the $n^{(c-1)s}$ term dominates. For $c = 0.5$, if $s > 10$, the inequality holds for sufficiently large n . On the other hand, when $s = \delta n$, replacing $(1+c) = \epsilon$ for simplicity, we have:

$$\begin{aligned} n^{\hat{s}-2s} \hat{s}^{2s-\hat{s}} e^{\hat{s}} &\leq n^{\epsilon\delta n-2\delta n} (\epsilon\delta n)^{2\delta n-\epsilon\delta n} e^{\epsilon\delta n} \\ &= (\epsilon\delta)^{(2-\epsilon)\delta n} e^{\epsilon\delta n} \\ &\leq (e)^{-6(2-\epsilon)\delta n} e^{\epsilon\delta n} \\ &= e^{-(12-7\epsilon)\delta n} \\ &= e^{-1.5\delta n} = e^{-n/e^6} \end{aligned}$$

if we let $\epsilon = 1.5$ and δ is chosen such that $\epsilon\delta = e^{-6}$. Due to the presence of $-n$ in the exponent of the last line, it should be clear that $\Pr[|S_{t+1}| = \epsilon n]$ is much less than n^{-4} .

We are, of course, interested in the probability that $|S_{t+1}| < (1+c)s = \epsilon s$, which can be given as $\sum_{k=1}^{\epsilon s} \Pr[|S_{t+1}| = k] \leq \epsilon s \cdot \binom{n}{\epsilon s} \frac{(\epsilon s)^{2s}}{n^{2s}} \leq \epsilon s \left(\frac{\epsilon s}{n}\right)^{(2-\epsilon)s} e^{\epsilon s}$. The second inequality comes from the fact that middle term $\binom{n}{\epsilon s} \frac{(\epsilon s)^{2s}}{n^{2s}}$ is increasing over the range $[1, \epsilon s]$ as long as $\epsilon s \ll 1/2$. For values of s at the lower end of the range $[C, \dots, \delta n]$, when $s = C$, by this reasoning, we have that the total probability is less than n^{-3} . We next show that this holds for the entire range by proving that the function $f(s) = \epsilon s \left(\frac{\epsilon s}{n}\right)^{(2-\epsilon)s} e^{\epsilon s}$ is decreasing over the range $[C, \delta n]$. The first derivative of the expression with regard to s is:

$$f'(n, s, \epsilon) = -\epsilon e^{\epsilon s} \left(\frac{\epsilon s}{n}\right)^{(2-\epsilon)s} \left[(\epsilon - 2)s \ln\left(\frac{\epsilon s}{n}\right) - 2s - 1 \right]. \quad (3)$$

This will be negative when the quantity in brackets is positive. This is equivalent to:

$$\begin{aligned} (\epsilon - 2)s \ln\left(\frac{\epsilon s}{n}\right) - 2s - 1 &> 0 \\ (\epsilon - 2)s [\ln(\epsilon) + \ln(s) - \ln(n)] &> 2s + 1 \\ (2 - \epsilon)[\ln(n) - \ln(\epsilon) - \ln(s)] &> 2 + 1/s. \end{aligned}$$

Clearly, this will hold when s is small, and since the function $g(s) = \ln n - \ln \epsilon - \ln s$ is a convex function on the range $[C, \delta n]$, it will not achieve a value greater than $\max g(C), g(\delta n)$. Thus, we only need to show that the inequality above also holds for $s = \delta n$. As before, we set $\epsilon = 1.5$ and $\delta = 1/(\epsilon e^6)$:

$$\begin{aligned} (2 - \epsilon) \left[\ln(n) - \ln(\epsilon) - \left(\ln \frac{n}{\epsilon e^6} \right) \right] &> 2 + \frac{1}{n/(\epsilon e^6)} \\ (2 - \epsilon) [\ln(n) - \ln \epsilon - \ln n + \ln \epsilon + 6] &> 2 + \frac{\epsilon e^6}{n} \\ 3 &> 2 + \frac{1.5e^6}{n} \end{aligned}$$

for sufficiently high values of n . Therefore, function $f(s)$ is decreasing and is less than $1/n^3$ for all values of s in range $[C, \delta n]$, proving the lemma. \square

Now we are ready to complete the proof of Theorem 5.2.

PROOF. There are three different phases we need to analyze. The first phase is when the active set is in the range $[1, \dots, C)$. Here, however, it is easy to show that the active set grows with a constant probability, and thus it grows in a sufficient number of continuous steps to achieve a value of C with constant probability. Thus, we only

need $O(\log n)$ such steps to achieve a size C with high probability. The next phase is covered by Lemma 5.2. When the active set is between $[C, \delta n]$, with probability $1 - n^{-3}$, it grows by a factor of 1.5 in each step. Thus, with high probability, it will grow for $\log n$ steps in a row, achieving an active set of size δn . Once the active set reaches size δn , with high probability, it will not go below δn for more than $1/2$ of the next $2 \log n$ steps. Meanwhile, each token of the cobra walk can choose any vertex of K_n uniformly at random. Therefore each vertex, in each time step, has at least a $\delta n/n = \delta$ probability of being selected by a token in each round. Thus, within another $O(\log n)$ steps, each vertex will have been visited by at least one token with high probability, completing the proof. \square

6. EXPANDERS

For expander graphs, we prove a high probability cover time of $O(\log^2 n)$. The proof is divided into two phases. In the first phase, we show that a cobra walk starting from any vertex will reach a constant fraction of the vertices in logarithmic (in $|V|$) time with high probability. In the second phase, we create a process that stochastically dominates the cobra walk and show that this new process will cover the entire rest of the graph again in polylogarithmic time with high probability.

The main result of this section can be stated formally as follows:

THEOREM 6.1. *Let G be any d -regular ϵ -expander with ϵ, δ not depending on n (the number of vertices in G), with $\delta < \frac{1}{2}$, and ϵ , a sufficiently small constant such that*

$$\frac{1}{\epsilon^2(1-\delta) + \delta} > \frac{d(de^{-k} + (k-1)) - \frac{k^2}{2}}{d(e^{-k} + (k-1)) - \frac{k^2}{2}}. \quad (4)$$

Then, with high probability, a cobra walk with branching factor $k = 2$ and starting from any vertex in G will cover G in $O(\log^2 n)$ time.

Recall that d -regular ϵ -expanders are defined in Definition 2.1, and Theorem 2.2 places a lower bound on the size of neighborhoods in such an expander.

We also note that the condition in Theorem 2.2 is satisfied if ϵ is sufficiently small for a fixed k . However, k can also be viewed as an adjustable parameter. Increasing k would allow for graphs with worse expansion to be covered by this result.

In the case where $k = 2$, this condition holds for strong expanders, such as the Ramanujan graphs, which have $\epsilon \leq 2\sqrt{d-1}/d$ and random d -regular graphs for d sufficiently large.

As described earlier, the proof of Theorem 6.1 has two parts. The first part, Phase I, handles the behavior of the cobra walk on G in the period between its start at a single vertex in G and the time when there are a linear number of active vertices (i.e., a constant fraction of the vertices). We show that this phase lasts no longer than $O(\log n)$ time.

The second part, Phase II, shows that once the cobra walk has reached a state in which a large fraction of vertices are active, the rest of the vertices will be hit by the walk within at most $O(\log^2 n)$ rounds. The method of showing this is somewhat indirect, in that the bound is shown for a somewhat similar process that stochastically dominates a cobra walk. The main idea, however, is that any vertex in G will be hit by the cobra walk within $O(\log n)$ time with constant probability. Thus, after $\Theta(\log n)$ runs of $O(\log n)$ steps, any (single) vertex will be hit with high probability. Performing a union bound over all vertices and choosing constants appropriately achieves the $O(\log^2 n)$ bound for the whole graph.

We next state the two lemmas associated with Phase I and Phase 2, respectively. Combining these two lemmas proves Theorem 6.1

LEMMA 6.2. *Let G be any d -regular ϵ -expander with ϵ, δ not depending on n (number of vertices in G), with $\delta < \frac{1}{2}$, and ϵ satisfying 4. Then, in time $O(\log n)$, with high probability, a cobra walk on G with branching factor k will attain an active set of size δn .*

LEMMA 6.3. *Let G be as above, and let W be a cobra walk on G that at time T has reached an active set of size δn . Then, with high probability, in an additional $O(\log^2 n)$ steps every vertex of G will have been visited by W at least once.*

6.1. Phase 1

To prove Lemma 6.2, we prove that active sets up to a constant fraction of V are growing at each step by a factor greater than 1. To do this, we show first that the size of the active set is growing by an exponential factor in expectation (c.f. Lemma 6.4). We then use a standard martingale argument to show that this growth occurs in any single step with high probability (c.f. Lemma 6.5). Finally, to complete the proof of Lemma 6.2, we show that there are a sufficient number of growth steps within the first $O(\log n)$ rounds. To do this, we view the size of the active set itself as a random variable and show that it is dominated by a random process derived from a negative binomial random variable that itself satisfies the $O(\log n)$ bound.

LEMMA 6.4. *Let G be any ϵ -expander with ϵ, δ satisfying the conditions of Theorem 6.2. Then, for any time $t \geq 0$, the cobra walk on G with active set S_t such that $|S_t| \leq \delta n$ satisfies $E[|S_{t+1}|] \geq (1 + \nu)|S_t|$ for some constant $\nu > 0$.*

PROOF. We instead show that the portion of $N(S_t)$ not selected by the cobra walk is sufficiently small, $E[|N(S_t) - S_{t+1}|] \leq |N(S_t)| - (1 + \nu)|S_t|$, and the result of the lemma will follow immediately.

For each vertex $u \in N(S_t)$, define X_u as an indicator random variable that takes value 1 if $u \notin S_{t+1}$ and 0 otherwise. Then, $\Pr[X_u = 1] = (1 - 1/d)^{kd_u}$, where d_u is the number of neighbors u has in S_t . Thus:

$$E[|N(S_t) - S_{t+1}|] = \sum_{u \in N(S_t)} X_u = \sum_{u \in N(S_t)} \left(1 - \frac{1}{d}\right)^{kd_u} \leq \sum_{u \in N(S_t)} e^{-\frac{kd_u}{d}}.$$

Because $\sum_{u \in N(S_t)} d_u = d|S_t|$ and we are working with a convex function, we have that $\sum e^{-\frac{kd_u}{d}}$ is maximized when all the values of d_u are equal to either 1 or d , with the exception of possibly one d_u to act as the remainder. Let R_1 be the number of vertices in $N(S_t)$ where $d_u = 1$, and let R_2 be the number of vertices where $d_u = d$. We have the following system of equations:

$$R_1 + R_2 = |N(S_t)| \tag{5}$$

$$R_1 + dR_2 = d|S_t|. \tag{6}$$

Solving for R_1 and R_2 , we get:

$$R_1 = \frac{d}{d-1} (|N(S_t)| - |S_t|) \tag{7}$$

$$R_2 = \frac{1}{d-1} (d|S_t| - |N(S_t)|). \tag{8}$$

We now want to show that

$$\begin{aligned} E[|N(S_t) - S_{t+1}|] &\leq R_1 e^{-\frac{k}{d}} + R_2 e^{-k} = \frac{d}{d-1} (|N(S_t)| - |S_t|) e^{-\frac{k}{d}} + \frac{1}{d-1} (d|S_t| - |N(S_t)|) e^{-k} \\ &\leq |N(S_t)| - (1 + \nu) |S_t|. \end{aligned}$$

Rearranging, we want to show that

$$|N(S_t)| \left(1 - \frac{d}{d-1} e^{-\frac{k}{d}} + \frac{1}{d-1} e^{-k} \right) + |S_t| \left(\frac{d}{d-1} e^{-\frac{k}{d}} - \frac{d}{d-1} e^{-k} - 1 \right) \geq \nu |S_t|.$$

We let $\alpha = \frac{1}{\epsilon^2(1-\delta)+\delta}$, then $|N(S_t)| \geq \alpha |S_t|$ (by Theorem 2.2), and we can divide through by $|S_t|$ in the preceding expression. Since the first quantity in parenthesis is positive, and we don't care what ν is as long as it's a positive constant, we are down to needing: $\alpha(1 - \frac{d}{d-1} e^{-\frac{k}{d}} + \frac{1}{d-1} e^{-k}) + (\frac{d}{d-1} e^{-\frac{k}{d}} - \frac{d}{d-1} e^{-k} - 1) > 0$. Rearranging, we want

$$(\alpha - 1) \left(1 - \frac{d}{d-1} e^{-\frac{k}{d}} \right) - \frac{d-\alpha}{d-1} e^{-k} > 0.$$

Taking the second-order Taylor approximation $e^{-\frac{k}{d}} \leq 1 - \frac{k}{d} + \frac{k^2}{2d^2}$, (6.1) will be satisfied if

$$(\alpha - 1) \left(1 - \frac{d}{d-1} \left(1 - \frac{k}{d} + \frac{k^2}{2d^2} \right) \right) - \frac{d-\alpha}{d-1} e^{-k} > 0,$$

which will be true for

$$\frac{1}{\epsilon^2(1-\delta)+\delta} = \alpha > \frac{d(de^{-k} + (k-1)) - \frac{k^2}{2}}{d(e^{-k} + (k-1)) - \frac{k^2}{2}}. \quad \square$$

Next, we use a martingale argument to show that the number of vertices in S_t is concentrated around its expectation.

LEMMA 6.5. *For a cobra walk on a d -regular ϵ -expander that satisfies the conditions in Lemma 6.4, at any time t*

$$\Pr[|S_{t+1}| - E[|S_{t+1}|] \leq -\tau |S_t|] \leq e^{-\frac{\tau^2 |S_t|}{2k}} \quad (9)$$

PROOF. Arbitrarily index the the vertices of S_t , $i = \{1, \dots, |S_t| = m\}$. Let (Z_i^j) be a sequence of random variables ranging over the indices i and $j = \{1, \dots, k\}$, where $Z_i^j = v$ indicates the i^{th} element of S_t that has chosen vertex v to place its j^{th} pebble. Define A as the random variable that represents the size of S_{t+1} . Then,

$$X_i^j = E[A | Z_1^1, \dots, Z_1^k, \dots, Z_i^1, \dots, Z_i^j]$$

is the Doob martingale for A , with $X_m^k = |S_{t+1}|$. Since $X_i^j - X_i^{j-1} \leq 1$ and $X_i^1 - X_{i-1}^1 \leq 1$ for all i, j , Azuma's inequality yields:

$$\Pr[|S_{t+1}| - E[|S_{t+1}|] \leq -\tau |S_t|] \leq e^{-\frac{\tau^2 m^2}{2km}} = e^{-\frac{\tau^2 m}{2k}}. \quad \square \quad (10)$$

We now complete the proof of Lemma 6.2. Using the bound of Lemma 6.5, we show that, with high probability, we will cover at least δn of the vertices of G with a cobra walk in logarithmic time by showing that the active set for some $t = O(\log n)$ is of size at least δn .

The key to this proof is to view a cobra walk on G as a Markov process over a state space consisting of all the possible sizes of the active set. In this interpretation, all configurations of pebbles in a cobra walk in which i vertices are active are equivalent.

The goal is to show that this new Markov process will reach a state corresponding to an active set of size δn quickly with high probability. To prove this, we first show that it is dominated by a restricted Markov chain over the same state space in which any negative growth in the size of the active set is replaced with a transition to the initial state (in which only one vertex is active). We then in turn show that the restricted walk is dominated by an even more restricted walk in which the probability of negative growth is higher than in the first restricted walk, bounded from below by a constant and no longer dependent on the size of the current state. We then show that the goal of the lemma is achieved even in this walk by relating the process to a negative binomial random variable.

PROOF. We view a cobra walk on G as a random walk W over the state space consisting of all the possible sizes of the active set: $S(W) = \{1, \dots, n\}$. We then define a Markov process M_1 that stochastically dominates W : Let $\tau = \nu/2$, where ν is the expected growth factor of the active set as shown in Lemma 6.4. The states of M_1 , $S(M_1)$, are the same as W 's, but the transitions between states differ. Each $i \in S(W)$ can have out-arcs to many different states, but the corresponding $i \in S(M_1)$ has only two transitions.

With probability $p_i = 1 - e^{-\frac{\nu^2 i}{8k}}$ transition to state $(1 + \nu/2)i$, and with probability $1 - p_i$ transition to state 1. Note that p_i is derived from Lemma 6.5.

In M_1 , each transition probability is still a function of the current state i , and, as mentioned earlier, we would like to eliminate this dependence. Thus, define M_2 as a random walk over the same state space. However, we will deal only with a subset of $S(M_2)$: the states: $(1 + \nu/2)^j C$ for $i \in \mathbb{Z}$ and a suitably large constant C . We then have the following transitions for each state in the chain (which will begin once it hits C). Setting $r = \nu^2/8k$, at state $(1 + \nu/2)^j C$: (i) Transition to state $(1 + \nu/2)^{j+1} C$ with probability $p'_j = 1 - e^{-rC(1+\frac{\nu}{2})^j}$. (ii) Transition to state C with probability $1 - p'_j$. This Markov chain oscillates between failure (going to C) and growing by a factor of $1 + \nu/2$. Note that to succeed (i.e., reach a state of at least δn), we need $\Omega(\log n)$ growing transitions.

The probability that in a walk on this state space we “fail” and go back to C before hitting δn is bounded by $1/2$, since $\sum_{i=0}^{\infty} e^{-rC(1+i\frac{\nu}{2})} \leq e^{-rC} \sum_{i=0}^{\infty} e^{irC\frac{\nu}{2}} = \frac{e^{-rC}}{1 - e^{-rC\frac{\nu}{2}}} \leq \frac{1}{2}$, provided that C is sufficiently large as a function of r (which is itself only a function of the branching factor and the constant ν).

Consider each block of steps that end in a failure (meaning that we return to C). Then, clearly, with high probability after $b \log n$ trials, for some constant b , we will have a trial that ends in success (i.e., reaching an active set of size δn vertices). In these $b \log n$ trials, there are exactly that many returns to C . However, looking across all trials that end in failure, there are also only a total of $O(\log n)$ steps that are successful (i.e., involve a growth rather than shrinkage). To see why this is true, note that the probability of a failure after a string of growth steps decreases supralinearly with each step, so that if we know we are in a failing trial, it is very likely that we fail after only a few steps. Thus, there cannot be too many successes before each failure. Indeed, the probability that we fail at step i within a trial can be bounded. Thus, $\Pr[\text{Failure at step } i \mid \text{eventual failure}]$

$$\begin{aligned} &= \frac{\Pr[\text{Failure at step } i]}{\Pr[\text{Eventual failure}]} \\ &= \frac{e^{-rC(1+i\nu/2)}}{\sum_{i=1}^{\infty} \left(\prod_{j=1}^{i-1} (1 - e^{-rC(1+j\nu/2)}) \right) e^{-rC(1+i\nu/2)}} \\ &\geq \frac{1}{\sum_{i=1}^{\infty} e^{-irC\nu/2}} \geq 1 - e^{-rC\nu/2} \end{aligned}$$

and thus the probability of advancing is no more than $e^{-rCv/2}$, also a quantity that does not depend on i . This is a negative binomial random variable with distribution $w(k, p)$, the number of coin flips needed to obtain k heads with heads probability p . Identifying heads with a failure (i.e., returning to C) and tails with making a growth transition, we have a random variable $w(k, p)$, the number of coin flips needed for k failures with probability of failure $p = 1 - e^{-rCv/2}$. It is well known that $\Pr[w(k, p) \leq m] = \Pr[B(m, p) \geq k]$, where $B(m, p)$ is the binomial random variable counting the number of heads within m p -biased coin flips. Thus, $\Pr[w(k, p) > m] = \Pr[B(m, p) < k]$. Setting $k = a \log n$ and $m = b \log n$, we have $\Pr[B(m, p) \leq \mathbb{E}[B(m, p)] - t] = \Pr[B(m, p) < pm - t] \leq e^{-\frac{t^2}{m}}$. We let $k = pm - t$, and, solving for t , we get $t = (pb - a) \log n$. This gives us

$$\Pr[B(m, p) < k] \leq \frac{1}{n^{\frac{(pb-a)^2}{b}}},$$

establishing that there are at most $O(\log n)$ successes within $O(\log n)$ trials ending in failure. Via stochastic dominance, this bound holds for our original cobra walk process. \square

6.2. Phase 2

Once the active set has reached size $\Omega(n)$, we need a different method to show that the cobra walk achieves full coverage in $O(\log^2 n)$ time. We cannot simply pick a random pebble and restart the cobra walk from this point $O(\log n)$ times because we know nothing about the distribution of the δn pebbles after restart, and the restarting method would require the pebbles to be independently identically distributed uniformly across the vertices of G . As a result, we are unable to establish a straightforward bound on h_{max} and invoke Matthew's theorem.

Hence, to prove Lemma 6.3, we develop a different process, which we call W_{alt} , that stochastically dominates the cobra walk. In W_{alt} , no more branching or coalescing occurs, and we also modify the transition probabilities of the pebbles on a vertex-by-vertex basis depending on the number of pebbles at a vertex. For technical reasons, we also make the W_{alt} process lazy in that, in any given round, with probability 1/2, the pebbles all remain in the same location.

Definition 6.6. For any time t and any collection of S pebbles on V (there can be more than 1 pebble at a vertex), define $W_{alt}(t + 1)$ as follows. With probability 1/2, all the pebbles remain in the same location; with probability 1/2, the pebbles move as follows. Let $A \subseteq V$ be the set of all vertices with 1 pebble at time t . Let $B \subseteq V$ be the set of all vertices with exactly 2 pebbles, and let C be the set of all vertices with more than 2 pebbles. Then, (i) for every $v \in A$, the pebble at v uniformly at random selects a vertex in $\Gamma(v)$ (the neighborhood of v not including itself) and moves to it; (ii) for every $v \in B$, each pebble at v uniformly at random selects a vertex in $\Gamma(v)$ and moves to it; (iii) for every $v \in C$, arbitrarily index the pebbles at v . Then, the first two pebbles each (index 1, 2) then pick a neighbor $\in \Gamma(v)$ to move to, uniformly at random. Let u, w be the two neighbors picked by pebbles 1, 2. (Note that $w = u$ is a possible outcome). The remaining pebbles (those with index > 2), still in the same time step, each independently pick u or w with probability 1/2 and move to the vertex they have selected.

The process W_{alt} can be thought of as a kind of coalescing (lazy) random walk in which the threshold for coalescence is three pebbles on a vertex rather than the standard two, with the added condition that the third (and higher) token chooses which of the first two tokens to coalesce with by flipping an unbiased coin.

Furthermore, at any time t , when W_{alt} is not lazy, a vertex with two or more pebbles behaves locally exactly the same in W_{alt} and a cobra walk. The divergence in behavior occurs only for those vertices with one pebble. In W_{alt} , they behave like vertices participating in a simple lazy random walk (or like vertices in a cobra walk in which both pebbles have chosen the same neighbor to move to in the next round). In light of these modifications, the active set of W_{alt} at any time t can be viewed as a (possibly proper) subset of the active set of a cobra walk on G at an earlier time that accounts for the lazy steps of W_{alt} , if both are started from identical initial conditions (i.e., distribution of pebbles). Therefore, the probability that the cover time of W_{alt} is greater than some value x is at least the probability of the cover time of the cobra walk being greater than x . Thus, the cover time of W_{alt} stochastically dominates the cover time of a cobra walk when started from the same initial condition.

Thus, if we can prove that the cover time of W_{alt} on G is $O(\log n)$ with high probability, this will imply that the cover time of the cobra walk on G has the same upper bound as well.

LEMMA 6.7. *Let G be a bounded-degree d -regular ϵ -expander graph, with ϵ sufficiently small to satisfy the conditions in Lemma 6.4. Let there be δn pebbles distributed arbitrarily over V , with at most one pebble per vertex, for a constant $\delta < 1/2$. Starting from this configuration, the cover time of W_{alt} on G is $O(\log^2 n)$, with high probability.*

PROOF. Our proof relies on showing that each vertex in G has a constant probability of being visited by at least one pebble during an epoch of W_{alt} lasting $\Theta(\log n)$ time. Once this has been established, all vertices of G will be covered with high probability after $O(\log n)$ epochs lasting $\Theta(\log n)$ steps each.

Define E_i to be the event that pebble i covers an arbitrary vertex v at time $s = \Theta(\log n)$, where the constant hidden in the Θ notation is chosen to be sufficiently large. We want to prove that the probability that v is covered by at least one pebble, $\Pr[\bigcup_i E_i]$, is constant. For pebbles i and j , we cannot assume that E_i and E_j are independent since the transition probabilities of the walks of i and j will not be independent if they are spatially and temporally co-located. However, we can calculate an upper bound using a second-order inclusion-exclusion approximation:

$$\Pr\left[\bigcup_i E_i\right] \geq \sum_i \Pr[E_i] - \frac{1}{2} \sum_{i \neq j} \Pr[E_i \cap E_j].$$

As a marginal probability, $\Pr[E_i]$ can be viewed as the probability that the (simple) random walk of pebble i hits v at time s . This is justified because if we only observe the movement of pebble i , at any time t , if i is at vertex w , its probability of walking to each of w 's d neighbors is $1/2d$, regardless of whether it is the first, second, or third pebble at w . Since it reduces to analyzing a simple random walk, we only need to look at the elements of zA^s , where A is the stochastic transition matrix of the simple lazy random walk on G and z is a vector with $z(l)$ equal to 1 for l equal to the position of pebble i at the beginning of the epoch and 0 in all other positions. By a standard analysis [Alon et al. 2008, Lemma 4.8], it follows that each coordinate of $A^s z$ differs from $1/n$ by at most $\frac{1}{2n}$ for $s \geq c \log n$, where c is a sufficiently large constant. Thus, $\Pr[E_i] \geq \frac{1}{2n}$.

Next, we establish an upper bound for $\Pr[E_j \cap E_i]$, the joint (and hence nonindependent) walks of pebbles i and j . For the purposes of this analysis, we consider pebble i to be the higher priority pebble: That is, if at any time i and j are co-located at some vertex and the process is not lazy in that round, we assume that i has first priority and chooses a neighbor to move to in the next step independently with uniform probability. We assume that j has third or lower priority and will choose i 's destination with probability $1/2$ and thus with probability $1/2$ choose a neighbor uniformly at random. We

can view the walks of i, j as a random walk on the tensor product $G \times G$. The tensor product has vertex set that is the Cartesian product $V(G) \times V(G)$. Vertex (u, u') of $G \times G$ has an edge to (v, v') if and only if (u, v) and (u', v') are edges of G . Note that the joint walk of i, j on G does not map to a simple random walk on $G \times G$; rather, it maps to a walk on a directed graph formed from $G \times G$ with weights chosen appropriately based on the process W_{alt} . In Lemma 6.8, we show that this walk, viewed as a nonreversible, irreducible Markov chain, has a stationary distribution close to $1/n^2$, that it converges rapidly to the stationary distribution, and, thus, after s steps, the probability that i and j are at the same vertex in G is no more than $2/(n^2 + n) + 1/n^4$.

With this result, we then have:

$$\begin{aligned} \Pr \left[\bigcup E_i \right] &\geq \sum_i \Pr[E_i] - \frac{1}{2} \sum_{i \neq j} \Pr[E_j \cap E_i] \\ &\geq \delta n \frac{1}{2n} - \binom{\delta n}{2} \left(\frac{2}{n^2 + n} + \frac{1}{n^4} \right) \\ &\geq \frac{\delta}{2} - \frac{\delta}{2} (\delta n^2 - n) \left(\frac{2}{n^2 + n} + \frac{1}{n^4} \right) \\ &\geq \frac{\delta}{2} - \frac{\delta^2 n^2}{n^2 + n} - \frac{\delta^2}{n^2} \\ &\geq \frac{\delta}{2} - \delta^2, \end{aligned}$$

which is a positive constant for any $\delta < 1/2$. The last inequality holds because the term $n^2/(n^2 + n) + 1/n^2$ is at most 1 for $n \geq 2$. \square

LEMMA 6.8. *Let G and s be defined as in Theorem 6.7. Let i and j be two pebbles walking on G according to the rules of W_{alt} . If $E_i \cap E_j$ defined as the event in which i and j are both at arbitrary vertex $v \in G$ at time s , then $\Pr[E_i \cap E_j] \leq 1/n^2$.*

PROOF. Let $G \times G$ be the tensor product chain as just defined. We first make some observations about the structure of this graph. We note that there are two types of vertices of $G \times G$. The first type involves all vertices that have the form (u, u) , where $u \in V(G)$. A pebble at (u, u) would correspond to two pebbles occupying u in the paired walk of i, j on G . We label this set S_1 . The cardinality of S_1 is n . The remaining vertices, which we place in the set S_2 , are of the form (u, v) for $u \neq v$. There are $n^2 - n$ such vertices. Also note that, in the undirected graph $G \times G$, each vertex has degree d^2 . Furthermore, every vertex in S_1 will have d neighbors also in S_1 by virtue of G being d -regular.

Many of the spectral properties of G apply to $G \times G$ as well. Primarily, because G is an ϵ -expander, the transition matrix of a simple random walk on G has a constant second eigenvalue $\alpha_2(G)$ bounded away from 0. It is well known [Levin et al. 2009] that $G \times G$ will have $\alpha_2(G \times G) = \alpha_2(G)$, which we refer to as α_2 henceforth.

We now take the undirected graph $G \times G$ and transform it into a directed graph $D(G \times G)$ as follows. For every undirected edge (x, y) in $G \times G$, replace it with two directed edges: $x \rightarrow y$ and $y \rightarrow x$. As mentioned, every vertex in S_1 will have d neighbors also in S_1 , meaning there will be one directed arc $x \rightarrow y$ for every vertex $x \in S_1$ and $y \in N(x), y \in S_1$. We now add an additional d copies of edge $x \rightarrow y$ for every such original edge. Finally, to account for the fact that W_{alt} is a process in which all pebbles stay simultaneously at their respective locations with probability $1/2$, we add as many self-loops at each vertex of $D(G \times G)$ as the number of outgoing edges at the vertex.

It is relevant for our analysis to note that because of the regularity of the subgraph of $G \times G$ induced on S_1 , for every vertex in $D(G \times G)$, the number of out-edges will equal the number of in-edges, and hence $D(G \times G)$ is an Eulerian digraph. Furthermore, we

can now calculate the transition probabilities of a random walk on $D(G \times G)$ where an outgoing edge from vertex x is picked with probability equal to the reciprocal of its out-degree. For all vertices in S_2 , the probability that the walk stays at the vertex is $1/2$; every other transition occurs with probability $1/(2d^2)$. This includes edges from S_2 into S_1 . On the other hand, although the probability that the walk stays at a vertex in S_1 is $1/2$, the probabilities of transitions from a vertex in S_1 to any other vertex is modified: The probability of transitioning from a vertex $x \in S_1$ to one of its $d^2 - d$ neighbors in S_2 is now $1/(4d^2)$, whereas the probability of transitioning to a neighbor in S_1 has become $(d + 1)/(4d^2)$ on account of the multiple edges. We have thus constructed a walk on digraph $D(G \times G)$ that maps exactly to the joint walk of pebbles i, j on G according to the rules of W_{alt} .

Because the walk on $D(G \times G)$ is irreducible, it has a stationary distribution π , which is the (normalized) eigenvector of the dominant eigenvalue of the transition matrix M of the walk on $D(G \times G)$. Furthermore, because $D(G \times G)$ is Eulerian, the stationary distribution of vertex x is exactly given by $d^+(x)/m$, where m is the number of edges. Therefore, there are only two distinct values of the components of the stationary vector: For all $x \in S_1$, $\pi(x) = 2/(n^2 + n)$, while for all $y \in S_2$, $\pi(y) = 1/(n^2 + n)$.

Because G and hence $G \times G$ have such nice spectral properties, and because $D(G \times G)$ represents a minor modification of $G \times G$, it would be reasonable to intuit that $D(G \times G)$ also has some of the same properties. Yet one must proceed with caution when analyzing Markov chains on directed graphs because some properties that hold for chains on undirected graphs do not carry through. However, following closely the work of Chung [2005], we can verify that the random walk on $D(G \times G)$ converges rapidly to its stationary distribution.

For succinctness of notation, denote $\mathcal{D} = D(G \times G)$. Consider the function $F_\pi : E(\mathcal{D}) \rightarrow \Re$ given by $F_\pi(x, y) = \pi(x)P(x, y)$, where $\pi(x)$ is the x -th component of the stationary distribution of the walk on \mathcal{D} , and $P(x, y)$ is the associated transition probability moving from x to y . Then F_π is the **circulation** associated with the stationary vector as shown in Lemma 3.1 of Chung [2005]. Note that a circulation is any such function that satisfies a balance equation: $\sum_{u, u \rightarrow v} F(u, v) = \sum_{w, v \rightarrow w} F(v, w)$.

There is a Cheeger constant for every directed graph, defined as:

$$h(G) = \inf_S \frac{F_\pi(\partial S)}{\min\{F_\pi(S), F_\pi(\bar{S})\}}, \quad (11)$$

where $F_\pi(\partial S) = \sum_{u \in S, v \notin S} F(u, v)$, $F(v) = \sum_{u, u \rightarrow v} F(u, v)$, and $F(S) = \sum_{v \in S} F(v)$ for a set S . Furthermore, Theorem 5.1 of Chung [2005] shows that the second eigenvalue, λ , of the Laplacian of \mathcal{D} satisfies:

$$2h(\mathcal{D}) \geq \lambda \geq \frac{h^2(\mathcal{D})}{2}. \quad (12)$$

The Laplacian of a directed graph is defined slightly differently than for an undirected graph. However, because we will not use the Laplacian directly in our analysis, we refer the reader to Chung [2005] for the definition. We will directly bound the Cheeger constant for \mathcal{D} , and hence produce a bound on the second eigenvalue of the Laplacian. This second bound will then be used to provide a bound on the convergence of the chain to its stationary distribution (see Theorem 6.9).

First, without loss of generality, assume that $F_\pi(S)$ is smaller than its complement. Furthermore, assume that S is the set that satisfies the inf condition in the Cheeger constant. We have $F_\pi(\partial S) = \sum_{x \rightarrow y, x \in S, y \in \bar{S}} \pi(x)P(x, y)$, and $F_\pi(S) = \sum_{x \rightarrow y, x \in S} \pi(x)P(x, y)$. The first sum occurs over all (directed) edges that cross the cut of S . Fortunately, we are already able to provide a lower bound on the size of this cut.

Observing that \mathcal{D} has at least as many edges crossing the cut as a cut on the equivalent set S in $G \times G$, and recalling that $G \times G$ has second eigenvalue α_2 , we can appeal to α_2 being bounded away from zero by a constant (by virtue of G 's expansion, recall) to state that there exists some constant β (depending on α_2) such that there are at least $\beta|S|$ edges crossing the cut. We can thus provide a lower bound for the entire Cheeger constant:

$$\begin{aligned} h(\mathcal{D}) &= \inf_S \frac{F_\pi(\partial S)}{\min\{F_\pi(S), F_\pi(\bar{S})\}} \\ &\geq \frac{\beta|S|P_{\min}\pi_{\min}}{|S|P_{\max}\pi_{\max}} \\ &= \beta \cdot \frac{\frac{1}{4d^2} \frac{1}{(n^2+n)}}{\frac{1}{2} \frac{1}{(n^2+n)}} = \frac{\beta}{4d^2} \end{aligned}$$

We can then apply the lower Cheeger inequality to have $\lambda \geq \frac{\beta^2}{32d^4}$. We now show the rapid convergence (in logarithmic time) of the walk on \mathcal{D} to the stationary distribution. To measure distance from the stationary distribution, we use the χ -square-distance:

$$\Delta'(t) = \max_{y \in V(\mathcal{D})} \left(\sum_{x \in V(G)} \frac{(P^t(y, x) - \pi(x))^2}{\pi(x)} \right)^{1/2}. \quad (13)$$

It can be shown that a rapid convergence for Δ' implies a rapid convergence for the total variational distance, and thus the distribution of a random walk starting anywhere in \mathcal{D} will be close to its stationary distribution for every vertex.

We next apply Theorem 7.3 from Chung [2005], which we state here for clarity:

THEOREM 6.9. *Suppose a strongly connected directed graph G on n vertices has Laplacian eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$. Then, G has a lazy random walk with the rate of convergence of order $2\lambda_1^{-1}(-\log \min_x \pi(x))$. Namely, after at most $t \geq 2\lambda_1^{-1}(-\log \min_x \pi(x) + 2c)$ steps, we have:*

$$\Delta'(t) \leq e^{-c}.$$

Recall that we provided a constant lower bound for λ , the second-smallest eigenvalue of the Laplacian of \mathcal{D} . Hence, we can apply it to the minimum running time of the random walk on \mathcal{D} to show that after at most

$$s = \frac{32d^4}{\beta^2} (\log(n^2 + n) + 4 \log n^2)$$

steps, we will have $\Delta'(t) \leq \frac{1}{n^4}$. Therefore, for the random walk on \mathcal{D} , after a logarithmic number of steps s (in n , the size of G), a walk that starts at any initial distribution on $V(\mathcal{D})$ will be within $1/n^4$ of the stationary distribution of any vertex. Mapping our analysis back directly to the coupled walk of i, j on G , $\Pr[E_i \cap E_j] \leq 2/(n^2 + n) + 1/n^4$ when pebbles i and j start from any initial position. \square

7. CONCLUSION

We studied a generalization of the random walk—namely, the cobra walk—and analyzed its cover time for trees, grids, complete graphs, and expander graphs. The cobra

walk is a natural random process with potential applications to epidemics and gossip-based information spreading. We plan to explore further the connections between cobra walks and the SIS model and pursue their practical implications. From a theoretical standpoint, there are several interesting open problems regarding cobra walks. The first one is to obtain a tight bound for the cover time of cobra walks on expanders. Our upper bound is $O(\log^2 n)$, whereas the diameter $\Omega(\log n)$ is an easy lower bound. Another pressing open problem is to determine the worst-case bound on the cover time of cobra walks on general graphs; we conjecture that it is $O(n \log n)$ with high probability. It will also be interesting to establish and compare the message complexity of a cobra walk with the standard random walk and other gossip-based rumor spreading processes.

ACKNOWLEDGMENTS

We would like to thank the reviewers for their detailed and insightful comments on earlier versions of this manuscript. We are especially grateful to the reviewer who urged us to revise the proof for the second phase of the process on expanders, which has resulted in a cleaner argument, and to the reviewer who suggested an improved result and proof for the cover time of grids.

REFERENCES

- Micah Adler, Eran Halperin, Richard M. Karp, and Vijay V. Vazirani. 2003. A stochastic process on the hypercube with applications to peer-to-peer networks. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC'03)*. ACM, New York, NY, USA, 575–584. DOI: <http://dx.doi.org/10.1145/780542.780626>
- Noga Alon, Chen Avin, Michal Koucky, Gady Kozma, Zvi Lotker, and Mark R. Tuttle. 2008. Many random walks are faster than one. In *Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures (SPAA'08)*. ACM, New York, NY, USA, 119–128. DOI: <http://dx.doi.org/10.1145/1378533.1378557>
- Siva R. Arthreya and Jan M. Swart. 2005. Branching-coalescing particle systems. *Probability Theory and Related Fields* 131, 3 (2005), 376–414. DOI: <http://dx.doi.org/10.1007/s00440-004-0377-4>
- Itai Benjamini and Sebastian Müller. 2010. On the trace of branching random walks. *arXiv preprint arXiv:1002.2781*.
- Petra Berenbrink, Colin Cooper, Robert Elsässer, Tomasz Radzik, and Thomas Sauerwald. 2010. Speeding up random walks with neighborhood exploration. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1422–1435. <http://dl.acm.org/citation.cfm?id=1873601.1873716>.
- Noam Berger, Christian Borgs, Jennifer T. Chayes, and Amin Saberi. 2005. On the spread of viruses on the internet. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'05)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 301–310. <http://dl.acm.org/citation.cfm?id=1070432.1070475>.
- Andrei Broder. 1989. Generating random spanning trees. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (SFCS'89)*. 442–447. DOI: <http://dx.doi.org/10.1109/SFCS.1989.63516>
- Marc Bui, Thibault Bernard, Devan Sohler, and Alain Bui. 2006. Random walks in distributed computing: A survey. In *Proceedings of the 4th International Conference on Innovative Internet Community Systems (IICS'04)*. Springer-Verlag, Berlin, Heidelberg, 1–14. DOI: http://dx.doi.org/10.1007/11553762_1
- Jen-Yeu Chen and Gopal Pandurangan. 2012. Almost-optimal gossip-based aggregate computation. *SIAM J. Comput.* 41, 3 (2012), 455–483.
- Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010a. Almost tight bounds for rumour spreading with conductance. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing (STOC'10)*. ACM, New York, NY, USA, 399–408. DOI: <http://dx.doi.org/10.1145/1806689.1806745>
- Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010b. Rumour spreading and graph conductance. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1657–1663. <http://dl.acm.org/citation.cfm?id=1873601.1873736>.
- Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2011. Rumor spreading in social networks. *Theoretical Computer Science* 412, 24 (2011), 2602–2610. Selected Papers from 36th International Colloquium on Automata, Languages and Programming (ICALP 2009). DOI: <http://dx.doi.org/10.1016/j.tcs.2010.11.001>.

- Fan Chung. 2005. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics* 9 (2015), 1–19.
- F. R. K. Chung, L. Lu, Conference Board of the Mathematical Sciences, and National Science Foundation (U.S.). 2006. *Complex graphs and networks*. Number no. 107 in CBMS Regional Conference Ser. in Mathematics Series. American Mathematical Society. <http://books.google.com/books?id=BqqDsEKIAE4C>.
- Francesc Comellas and Silvia Gago. 2005. A star-based model for the eigenvalue power law of Internet graphs. *Physica A: Statistical Mechanics and its Applications* 351, 24 (2005), 680–686. DOI: <http://dx.doi.org/10.1016/j.physa.2005.01.003>
- Colin Cooper, Robert Elsässer, Hirotaka Ono, and Tomasz Radzik. 2012. Coalescing random walks and voting on graphs. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing (PODC'12)*. ACM, New York, NY, USA, 47–56. DOI: <http://dx.doi.org/10.1145/2332432.2332440>
- Atish Das Sarma, Danupon Nanongkai, and Gopal Pandurangan. 2009. Fast distributed random walks. In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing (PODC'09)*. ACM, New York, NY, USA, 161–170. DOI: <http://dx.doi.org/10.1145/1582716.1582745>
- Atish Das Sarma, Danupon Nanongkai, Gopal Pandurangan, and Prasad Tetali. 2010. Efficient distributed random walks with applications. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'10)*. ACM, New York, NY, USA, 201–210. DOI: <http://dx.doi.org/10.1145/1835698.1835745>
- Nedialko B. Dimitrov and C. Greg Plaxton. 2005. Optimal cover time for a graph-based coupon collector process. In *Proceedings of the 32nd International Conference on Automata, Languages and Programming (ICALP'05)*. Springer-Verlag, Berlin, Heidelberg, 702–716. DOI: http://dx.doi.org/10.1007/11523468_57
- Moez Draief and Ayalvadi Ganesh. 2011. A random walk model for infection on graphs: Spread of epidemics and rumours with mobile agents. *Discrete Event Dynamic Systems* 21, 1 (2011), 41–61. DOI: <http://dx.doi.org/10.1007/s10626-010-0092-5>
- Rick Durrett. 2010. Some features of the spread of epidemics and information on a random graph. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4491–4498. DOI: <http://dx.doi.org/10.1073/pnas.0914402107>
- Klim Efremenko and Omer Reingold. 2009. How well do random walks parallelize?. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'09/RANDOM'09)*. Springer-Verlag, Berlin, Heidelberg, 476–489. DOI: http://dx.doi.org/10.1007/978-3-642-03685-9_36
- Robert Elsässer and Thomas Sauerwald. 2011. Tight bounds for the cover time of multiple random walks. *Theoretical Computer Science* 412, 24 (2011), 2623–2641. Selected Papers from 36th International Colloquium on Automata, Languages and Programming (ICALP 2009). DOI: <http://dx.doi.org/10.1016/j.tcs.2010.08.010>
- U. Feige. 1993. A Tight Upper Bound on the Cover Time for Random Walks on Graphs. (1993).
- Uriel Feige. 1995. A tight lower bound on the cover time for random walks on graphs. *Random Struct. Algorithms* 6, 4 (July 1995), 433–438. DOI: <http://dx.doi.org/10.1002/rsa.3240060406>
- Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. 1990. Randomized Broadcast in Networks. (1990), 128–137. <http://dl.acm.org/citation.cfm?id=646475.693450>.
- Nikolaos Fountoulakis and Konstantinos Panagiotou. 2010. Rumor spreading on random regular graphs and expanders. In *Proceedings of the 13th International Conference on Approximation, and 14 the International Conference on Randomization, and Combinatorial Optimization: Algorithms and Techniques (APPROX/RANDOM'10)*. Springer-Verlag, Berlin, Heidelberg, 560–573. <http://dl.acm.org/citation.cfm?id=1886521.1886565>.
- Nikolaos Fountoulakis, Konstantinos Panagiotou, and Thomas Sauerwald. 2012. Ultra-fast rumor spreading in social networks. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12)*. SIAM, 1642–1660. <http://dl.acm.org/citation.cfm?id=2095116.2095246>.
- Ayalvadi J. Ganesh, Laurent Massouli, and Donald F. Towsley. 2005. The effect of network topology on the spread of epidemics. In *INFOCOM (2006-10-04)*. IEEE, 1455.1466. <http://dblp.uni-trier.de/db/conf/infocom/infocom2005.html#GaneshMT05>.
- George Giakkoupis. 2011. Tight bounds for rumor spreading in graphs of a given conductance. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011) (Leibniz International Proceedings in Informatics (LIPIcs))*, Thomas Schwentick and Christoph Dürr (Eds.), Vol. 9. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 57–68. DOI: <http://dx.doi.org/10.4230/LIPIcs.STACS.2011.57>
- George Giakkoupis and Thomas Sauerwald. 2012. Rumor spreading and vertex expansion. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12)*. SIAM, 1623–1641. <http://dl.acm.org/citation.cfm?id=2095116.2095245>.

- Or Givan, Nehemia Schwartz, Assaf Cygelberg, and Lewi Stone. 2011. Predicting epidemic thresholds on complex networks: Limitations of mean-field approaches. *Journal of Theoretical Biology* 288 (2011), 21–28. DOI: <http://dx.doi.org/10.1016/j.jtbi.2011.07.015>
- Theodore E. Harris. 1963. *The theory of branching processes*. Springer-Verlag, Berlin. xiv+230 pages.
- David A. Kessler. 2008. Epidemic size in the SIS model of endemic infections. *Journal of Applied Probability* 45, 3 (2008), 757–778. DOI: <http://dx.doi.org/10.1239/jap/1222441828>
- David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. 2009. *Markov chains and mixing times*. Providence, R.I. American Mathematical Society. <http://opac.inria.fr/record=b1128575>. With a chapter on coupling from the past by James G. Propp and David B. Wilson.
- L. Lovász. 1996. Random walks on graphs: A survey. In *Combinatorics, Paul Erdős is Eighty*, D. Miklós, V. T. Sós, and T. Szőnyi (Eds.). Vol. 2. János Bolyai Mathematical Society, Budapest, 353–398.
- Neal Madras and Rinaldo Schinazi. 1992. Branching random walks on trees. *Stochastic Processes and Their Applications* 42, 2 (1992), 255–267. DOI: [http://dx.doi.org/10.1016/0304-4149\(92\)90038-R](http://dx.doi.org/10.1016/0304-4149(92)90038-R)
- Peter Matthews. 1988. Covering problems for Brownian motion on spheres. *The Annals of Probability* 16, 1 (1988), 189–199. DOI: <http://dx.doi.org/10.1214/aop/1176991894>
- Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA.
- Roni Parshani, Shai Carmi, and Shlomo Havlin. 2010. Epidemic threshold for the susceptible-infectious-susceptible model on random networks. *Phys. Rev. Lett.* 104, 25 (June 2010), 258701. DOI: <http://dx.doi.org/10.1103/PhysRevLett.104.258701>
- Daniel Spielman. 2012. Spectral graph theory lecture notes. Available at <http://www.cs.yale.edu/homes/spielman/561>.
- Rongfeng Sun and Jan M. Swart. 2008. The brownian net. *The Annals of Probability* 36, 3 (2008), 1153–1208.
- R. Michael Tanner. 1984. Explicit concentrators from generalized N-gons. *SIAM Journal on Algebraic Discrete Methods* 5, 3 (1984), 287–293.
- Piet Van Mieghem. 2011. The N -intertwined SIS epidemic network model. *Computing* 93, 2–4 (Dec. 2011), 147–169. DOI: <http://dx.doi.org/10.1007/s00607-011-0155-y>
- Ming Zhong and Kai Shen. 2006. Random walk based node sampling in self-organizing networks. *SIGOPS Oper. Syst. Rev.* 40, 3 (July 2006), 49–55. DOI: <http://dx.doi.org/10.1145/1151374.1151386>

Received November 2013; revised December 2014; accepted August 2015