The Power of Choice in Random Walks: An Empirical Study

Chen Avin and Bhaskar Krishnamachari
Department of Electrical Engineering, University of Southern California
Los Angeles, CA 90089,USA.
avin@cs.ucla.edu, bkrishna@usc.edu

ABSTRACT

In recent years different authors have proposed the used of random-walk-based algorithms for varying tasks in the networking community. These proposals include searching, routing, self-stabilization, and query processing in wireless networks, peer-to-peer networks and other distributed systems. This approach is gaining popularity because random walks present locality, simplicity, low-overhead and inherent robustness to structural changes. In this work we propose and investigate an enhanced algorithm that we refer to as *random walks with choice*, in which at each step, instead of selecting just one neighbor, the walk moves to the next node after examining a small number of neighbors sampled at random. Our empirical results on random geometric graphs, the model best suited for wireless networks, suggest a significant improvement in important metrics such as the cover time and load-balancing properties of random walks. To obtain more robust and systematic results, we investigate random walks with choice on networks with a square grid topology. For this case, our simulations indicate that there is an unbounded improvement in cover time even with a choice of only two neighbors. We also observe a large reduction in the variance of the cover time, and a significant improvement in visit load balancing.

1. INTRODUCTION

A random walk on a graph is the process of visiting the nodes of the graph in some sequential random order. The walk starts at some fixed node, and at each step it moves to a neighbor of the current node chosen randomly. The random walk is called *simple* when the next node is chosen uniformly at random from the set of neighbors. In a networking context, random walks result when messages are sent at random from device to device. Since this process presents locality, simplicity, low-overhead and robustness to structural changes, applications based on random walk techniques are becoming more and more popular in the networking community. In recent years different authors have proposed the used of random walk for querying/searching, routing, self-stabilization in wireless networks, peer-to-peer networks, and other distributed systems [12, 28, 8, 27, 5, 15, 1].

For example, for a query processing task in wireless sensor networks, a base station can issue a query with some description, such as "return the maximum temperature in the network". The token then takes a random walk in the network and updates its answer at each node; after visiting enough nodes, or after enough time, the query trace its way back to the base station with the answer.

One of the main reasons that random walk techniques are so appealing for networking application is their robustness to dynamics. Many wireless and mobile networks are subject to dramatic structural changes created by sleep modes, channel fluctuations, mobility, device failures, and other factors. Thus, topology driven algorithms are at a disadvantage for such networks as they need to maintain data structures (e.g. pointers to cluster heads, routing tables and spanning trees) and so have to handle recovery mechanisms for critical points of failure (e.g. cluster heads, nodes close to the root in a spanning tree). Consequently, algorithms that require no knowledge of network topology, such as the random walk, are at an advantage. In random walks there are no critical points of failure; on the contrary, all the nodes are equally unimportant at all times so long as the probability of a node failing during the short time it holds the message is considered negligible.

While at first glance, the process of a token wondering randomly in the network may seem overly simplistic and highly inefficient, many encouraging results that proves its comparability with other approaches have been obtain over the years. Two basic properties of random walk need to be evaluated in order to bound the efficiency of this approach: *cover time* and *partial cover time*. The cover time C_G of a graph G is the expected time taken by a simple random walk to visit all nodes in G and the partial cover $C_G(c)$ is the expected time to visit a constant fraction c of the nodes. These properties are relevant to a wide range of algorithmic applications [17, 31, 20, 5, 15], and various methods of bounding the cover time of graphs have been thoroughly investigated [4, 22, 3, 10, 9, 32]. Several bounds on the cover time of random walks on different classes of graphs have been obtained with many positive results [10, 9, 18, 19, 11].

One such example are random geometric graphs, which are most suitable for modeling wireless networks. A random geometric graph is a graph $\mathcal{G}(n, r)$ resulting from placing n points uniformly at random on the unit square and connecting two points iff their Euclidean distance is at most r. In the last few years random geometric graphs have been used as a fundamental model for randomly-deployed wireless adhoc and sensor networks. Recently it has been proven that, when $r = \Theta(r_{\text{con}})$ then *w.h.p.*[1] $\mathcal{G}(n, r)$ has optimal cover time of $O(n \log n)$ and optimal partial cover time of $O(n)$ [6] where r_{con} growing as $O(\sqrt{\frac{\log n}{\pi n}})$ is the critical radius to guarantee connectivity *w.h.p.*

Improving the cover time without losing the locality, simplicity and robustness of the random walk is an important goal that is directly related to the performance and energy usage of a random-walk-based query mechanism. There are other properties of the walk also that need to be addressed in order to improve overall application performance. One is reducing the variance of cover time (i.e. preventing queries that take a very long time), and another is balancing the load on the nodes (i.e. number of visits) by the time of cover, which will increase the system lifetime, in case of battery-constrained wireless sensor networks. In this paper, we take a step in this direction by offering and investigating a new way to improve upon these desire properties by combining random walks with a probabilistic tool known as the *power of choice*.

The essential idea behind the power of choice is to make some decision process more efficient by selecting the best among a small number of randomly generated alternatives. The most basic results about the power of choice are as follows: suppose one throws n balls into n bins one by one, where at each time the next bin is chosen independently and uniformly at random. It is well known that the most loaded bin at the end of the process will have about $\frac{\log n}{\log \log n}$ balls *w.h.p.* [24]. Consider the following change to the above scheme involving choice. At each step, instead of one bin, we choose a constant $d \geq 2$ bins independently and uniformly and put the ball in the bin with the minimum number of balls. In a somewhat-surprising result by Azar *et al.* [7], it has been shown that with this change, the most loaded bin will have $\frac{\log \log n}{\log d} + \Theta(1)$ balls with high probability. So with only a little more work a each step, (choosing two bins instead of one) we see a large improvement. Notice that increasing d farther yields only a constant factor improvement giving diminishing returns. Since it was first offered, the idea of the power of choice has spread in different directions with new results and applications to hashing, load balancing in distributed systems and low-congestion routing, among others [23].

In this work we propose (for the first time, to our knowledge) to combine the power of choice with random walks. We introduce the *Random Walk with Choice, RWC(d)*, in which, instead of selecting one neighbor at each step, the walk selects d neighbors uniformly at random and then chooses to steps to the least visited node among them (a related modification to random walks called *Vertex-Reinforced Random Walks*, VRRW, was proposed in [25, 30] and studied outside of the context of cover time. In VRRW the walk prefers the *most* visited nodes, without choice.). Note that simple random walk with choice consumes a bit more memory and more energy (communication) at each step. This is because we need to keep track of visits at each node and need to consider and choose between d nodes at each step. The question we wish to explore is whether there will be some substantial gain from making this change.

For the complete graph (which resembles the balls-in-bins), the analytical result shows that the cover time of *RWC(d)* will be reduced by a factor of d. For general graphs the lack of the Markov property[2] suggest that the analytical results may be harder to obtain. In the current work we therefore turn to a simulation-based study of the behavior of the random walk with choice. Our results demonstrate the power that comes with choice. We observe a consistent improvement in the cover time, cover time distribution and the load balancing at cover time for different graphs and different sizes. A surprising result is that, for 2-dimensional square grid networks, choice seems to improve the cover time and the load on the most visited node by an unbounded factor. Specifically, the cover time of the n nodes mesh is known to be $\Theta(n \log^2 n)$, our simulations shows that with $d = 2$ random walk with choice has lower cover time than the simple random walk on the hyper-cube that is known to have optimal cover time of $\Theta(n \log n)$. We also find improvements in the variance of the cover time, and load balancing of visits.

The rest of the paper is organized as follows: Section 2 gives background and formal definitions. Section 3 presents the *RWC(d)* and proves results for the complete graph. In section 4 we describe the simulation details and discuss the metrics of interest. Section 5, 6 and 7 and presents the results for various graph models. We present our conclusions in Section 8.

2. BACKGROUND AND PRELIMINARIES

2.1 Cover Time and Partial Cover

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

A Random Walk is the process of visiting the nodes of a graph $G(V, E)$ in some sequential random order. The walk starts at some fixed node, and at each step it moves to a neighbor of the current node chosen randomly according to an arbitrary distribution. The *simple random walk*, *SRW*, is a walk where the next node is chosen uniformly at random from the set of neighbors. That is when walk is at the v the probability to move in the next step to u is $P(v, u) = \frac{1}{\delta(v)}$ for $(v, u) \in E$ and 0 otherwise.

The *cover time* C_G of a graph G is the expected time taken by a simple random walk on G to visit all nodes in G. Formally, for $v \in V$ let C_v be the expected number of steps for the simple random walk starting at v to visit all the nodes

[1]Event $\mathcal{E}_n$ occurs with high probability (*w.h.p.*) if probability $P(\mathcal{E}_n)$ is such that $\lim_{n\to\infty} P(\mathcal{E}_n) = 1$.

[2]In the simple random walk the next step is independent of past steps, in the random walk with choice this is not the case.

in G, and the cover time of G is $C_G = \max_v C_v$. The *cover time* of graphs and methods of bounding it have been extensively investigated [22, 3, 10, 9, 32, 4]. Results for the cover time of specific graphs vary from the *optimal cover time* of $\Theta(n \log n)$ associate with the complete graph to the worst case of $\Theta(n^3)$ associate with the lollipop graph [14, 13]. The known best cases correspond to dense, highly connected graphs, on the other end when connectivity decreases and bottlenecks exist in the graph, the cover time increases. In this paper we consider three types of graphs:

1. **meshes**: G_n^2 - the 2 dimensional mesh (i.e. grid on the torus) of size n. Its is known to have non-optimal cover time of $\Theta(n \log^2 n)$ [10].

2. **hyper-cubs**: H_n - the hyper-cube which is the d-dimensional mesh of size n with $d = \log_2(n)$. H_n is known to have an optimal cover time [24].

3. **random geometric graphs**: $\mathcal{G}(n, r)$ - for $r \geq \sqrt{8} r_{\text{con}}$ $\mathcal{G}(n, r)$ has optimal cover time *w.h.p.* [6].

The *partial cover time* [5] is the expected time taken by a random walk to visit a constant fraction of the nodes and is define formally as follow: For $0 \leq c \leq 1$, let $C_G(c)$ be the expected time taken by a simple random walk on G to visit $\lfloor cn \rfloor$ of the nodes of G. Let H_{uv} be the *hitting time*, the expected time for a random walk starting at u to arrive to v for the first time and let $H_{\max}$ be the maximum H_{uv} over all ordered pairs of nodes. In [5] it was proven that for any graph G, and $0 \leq c < 1$ we have $C_G(c) = \Theta(H_{\max})$. This implies the following interesting results: for graphs in which $H_{\max} = n$, the partial cover becomes linear in n and we consider it to be *optimal partial cover*; known graphs of this type are the complete graph, the star, the hyper-cube, the 3-dimensional mesh and random geometric graph which have been added to this list recently. On the other hand, for the 2-dimensional mesh, the maximum hitting time is $\Theta(n \log n)$ [32] so partial cover becomes $\Theta(n \log n)$.

Note that the analytical results about cover and partial are about the expected time. Less is known about the distribution of the cover time, but it been observed that random walk based algorithm usually have “heavy tailed” distribution, meaning that with non-negligible probability we should expect some very long cover times [16].

2.2 Load Balancing and the Stationary Distribution

The probabilistic rules by which a random walk operates are defined by the corresponding *Markov chain*. Let $\mathfrak{M}$ be a Markov chain over state space Ω and probability transition matrix P (i.e. $P(x, y)$ is the probability to move from x at time t to y at time $t + 1$). In such terms, the stationary distribution of $\mathfrak{M}$, if such exists, is then defined as the unique probability vector π such that

$$\pi P = \pi$$

It is well known that the simple random walk $\mathfrak{M} = (\Omega, P)$ over a connected graph $G = (V, E)$ has a stationary distribution π such that, for any node $q \in V$ [21],

$$\pi(q) = \frac{\delta(q)}{2m} \tag{1}$$

Further, when the underlying graph G is δ regular, the stationary distribution is the uniform distribution [21]

$$\pi(q) = \frac{\delta}{2m} = \frac{1}{n} \ \forall q \in \Omega$$

where $n = |\Omega| = |V|$.

At stationary distribution, it is clear that the random walk has optimal load-balancing qualities for regular graphs G. Similarly, it is clear that the faster the random walk on a regular graph converges to stationarity, the greater its load-balancing qualities. The efficiency with which a random walk of $\mathfrak{M}$ may be used to sample over state space Ω with respect to stationary distribution π is precisely given by the rate at which the distribution of the states at time t converges to π as $t \to \infty$. In order to speak of convergence of probabilities, one must have a notion of distance over time. Let x be the state at time $t = 0$ and denote by $P^t(x, \cdot)$ the distribution of the states at time t. The *variation distance* at time t with respect to the initial state x is defined to be [29, 26]

$$\Delta_x(t) = \max_{S \subseteq \Omega} |P^t(x, S) - \pi(S)| = \frac{1}{2} \sum_{y \in \Omega} |P^t(x, y) - \pi(y)|$$

Here we will be using the variation distance at time of cover to evaluate the load balancing of the random walk. In general the variation distance is used to determine the *mixing time* (i.e. the time in which the chain is ϵ close to stationary) of the chain and if the chain is *rapidly mixing*, namely the mixing time is $O(\text{poly}(\log n))$.

3. RANDOM WALKS WITH CHOICE

The *balls in bins* scenario can be described as a random walk on the complete graph K_n (with the addition of one self-loop for each node). The most loaded bin correspond to the most visited node after n steps of the walk. The idea we set to investigate here is to generalize choice at each step to random walks on arbitrary graphs. Formally we define the Random Walk with d Choice *RWC(d)* as the following process: let $c^t(v)$ be the number of visits to v up to time t. When the walk reaches v at time t it does the following:

RWC(d) at node v, at time t

1. Select d nodes from $N(v)$ independently and uniformly at random (with replacement).

2. Step to node u that minimizes $\frac{c^{t+1}(u)}{\delta(u)}$ (break ties in an arbitrary way).

Few remarks are in place: If the graph is regular, the walk steps to the least visited node; if not, the walk steps to the node that is farther a way from its stationary distribution $\pi(u)$. Clearly for $d = 1$ this is the simple random walk. For

$d > 1$ the Markov property doesn't hold anymore since the current step depends on past steps.

The last property is what seems to make the analytical results harder to obtain. We can regain the Markov property by changing the state space to one in which each state is a vector of size $n+1$ that holds the number of visits at time t for each node and the current node. This is a direction we have been following to prove theoretical results similar in flavor to those obtained for the balls in bins problem, but it is still challenging. In this work, we will focus, instead, on providing some preliminary observations obtained through careful simulations.

Our overall goal, as mentioned above, is to use choice in order to reduce the cover time and to obtain better load balancing. At first it may not be clear that choice will have any asymptotic affect on the cover time. The complete graph is easy to analyze, and we can show a constant factor improvement in this case. Let $C_G^{\mathcal{A}}$ denote the cover time of algorithm $\mathcal{A}$ on the graph G.

LEMMA 1. *For a constant $d \geq 2$ and the complete graph K_n (with self-loops) the cover time $C_{K_n}^{RWC(d)}$ is*

$$C_{K_n}^{RWC(d)} = \frac{C_{K_n}^{SRW}}{d}(1 + o(1))$$

PROOF. Let h_n be the harmonic sum $h_n = \sum_1^n i^{-1} \approx \log n$. It is well known that $C_{K_n}^{\text{SRW}} = nh_{n-1}$. For simplicity we present the case of $d = 2$ and we will follow the proof for the simple random walk, *SRW*, from [2]. Let C^m be the first time at which m distinct nodes have been covered. For each step after time C^m we will step to a visited node with probability $(\frac{m}{n})^2$ (sampling twice from visited nodes), so we will hit a new node with probability $\frac{n^2-m^2}{n^2}$ and the expected time to hit such a node is $E(C^{m+1} - C^m) = \frac{n^2}{n^2-m^2}$. The cover time will be:

$$C_{K_n}^{\text{RWC}(d)} = \sum_{m=1}^{n-1} E(C^{m+1} - C^m) = \sum_{m=1}^{n-1} \frac{n^2}{n^2 - m^2}$$

taking $m = n - x$ we get:

$$\begin{aligned}
C_{K_n}^{\text{RWC}(d)} &= \sum_{x=1}^{n-1} \frac{n^2}{n^2 - (n-x)^2} \\
&= \sum_{x=1}^{n-1} \frac{n^2}{2nx - x^2} \\
&= \frac{n}{2} \sum_{x=1}^{n-1} (\frac{1}{x} + \frac{1}{2n - x}) \\
&= \frac{n}{2} h_{n-1} + \frac{n}{2}(h_{2n-1} - h_n) \\
&= \frac{n}{2} h_{n-1} + \frac{n}{2}(\log(\frac{2n-1}{n})) \\
&= \frac{C_{K_n}^{\text{SRW}}}{2} + o(C_{K_n}^{\text{SRW}}) \qquad \square
\end{aligned}$$

Intuitively, since the complete graph has the lowest cover time for the simple random walk, *SRW*, over all graphs, it will have the lowest cover time for *RWC* as well. It follows then that for any graph that has optimal cover time we can expect at most a constant factor improvement in the cover time (regardless of the order of improvement in the load balancing). What will be the results of choice in a non-optimal graph? In the next sections we will explore this on the the 2-dimensional grid that is known to have a non-optimal cover time of $\Theta(n \log^2 n)$.

4. SIMULATION SET UP

We run our simulations on three types of graphs: (i) the random geometric graph $\mathcal{G}(n, r)$, (ii) the 2-dimensional mesh grid - G_n^2 and (iii) the hyper cube - H_n. The random geometric graphs have been widely used to model link connectivity and protocol behavior in randomly deployed wireless networks. The grid mesh (with wrap around of boundaries into a torus to avoid edge-effects) provides a deterministic graph which also has geometric locality and is also used to model carefully deployed wireless sensor networks. Note that both the grid and the hyper-cube are regular graph with a uniform stationary distribution. For the random geometric graph, $\mathcal{G}(n, r)$, we used $n = 900$ and $r = 2r_{\text{con}} = 0.0981$. On the grid we run the simulation for $n = \{100x^2 \mid x = 1, 2, 3, \ldots, 10\}$. For the hyper-cube we used $n = \{2^x \mid x = 7, 8, \ldots 13\}$. For each graph we execute the *RWC(d)* for $d = 1, 2, 3$ and in each case we average over 1000 runs. The results for the grid of sizes $n = 100, 400, 900$ and $\mathcal{G}(900, 0.0981)$ are an exception; they are based on 10,000 walks for each case, and used in particular to obtain histograms for the cover time.

4.1 Metrics and Questions of Interest

We set out to consider and explore the following metrics and related questions:

1. **Cover time progress up to full cover**: What is the improvement in cover time and partial cover for *RWC(d)*, $d \geq 2$? When dealing with cover time we normalize the number of steps by dividing out by n, the graph size. This allows us to compare different graphs sizes on one figure.

2. **Asymptotic behavior and asymptotic improvement**: The cover time improvement ratio for a constant $d \geq 2$ is defined to be:

$$I_d(n) = \frac{C_G^{\text{RWC}(d-1)}}{C_G^{\text{RWC}(d)}}$$

What is the asymptotic behavior of $I_d(n)$ for the different graphs? We know that for the complete graph K_n, $I_d(n) = d - o(1)$. Note that if $I_d(n) = \omega(1)$ (i.e. an order larger than a constant) then $C_G^{\text{RWC}(d)}$ is of a lower order than $C_G^{\text{RWC}(d-1)}$. Similarly we define the improvement ratio for the partial cover (e.g. 50%) and ask the same question.

3. **Cover time distribution**: Will $d \geq 2$ change the variance of the cover time? Does it eliminate or minimize the long tail of the cover time distribution of the simple random walk? How else does it change the distribution?

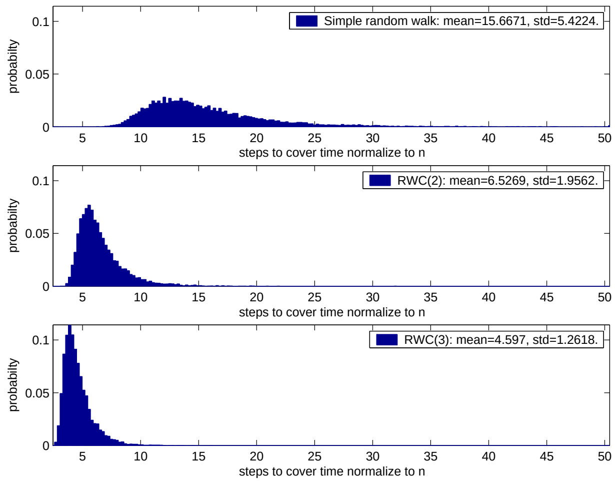


Figure 1: The distribution of the cover time on $\mathcal{G}(900, 0.0981)$ as an histogram from 10000 runs

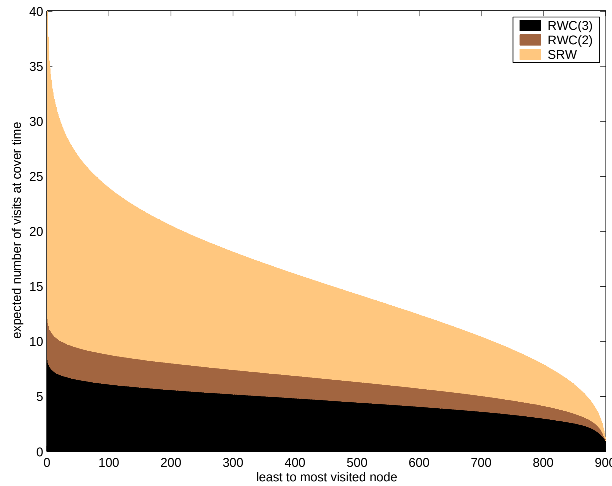


Figure 2: The load balance at cover time for random geometric graph $\mathcal{G}(900, 0.0981)$

Table 1: Statistical data for the cover time distribution of $\mathcal{G}(900, 0.0981)$ (normalized by n)

Type	mean	std	median	max	95%
SRW	15.66	5.42	14.07	67.59	25.99
RWC(2)	6.52	1.95	6.04	32.03	10.25
RWC(3)	4.59	1.26	4.29	16.21	7.03

4. **Load balancing**: Load balancing is of crucial interested in energy-limited wireless networks where such protocols may be implemented. To measure load balancing, we check the effect of choice on the most visited node. Let $L_G^{\mathrm{RWC}(d)}$ be the expected number of visits to the most visited node at cover time. For a graph G and a constant $d \geq 2$ we define the improvement ratio of the most visited node as:

$$L_d(n) = \frac{L_G^{\mathrm{RWC}(d-1)}}{L_G^{\mathrm{RWC}(d)}}$$

Finding $L_d(n)$ was the original result for the power of choice in the balls in bins (complete graph), do we have a similar effect on the grid? Note one difference in our setting — the original result is for the load balancing time $t = n$, while here we consider the load balancing at time of cover. Next we extend this to all the nodes: at cover time we order the nodes from the most loaded to the least loaded (which always has 1 visit at cover time) and average over all runs. This yields the expected number of visits to the $i'th$ most visited node.

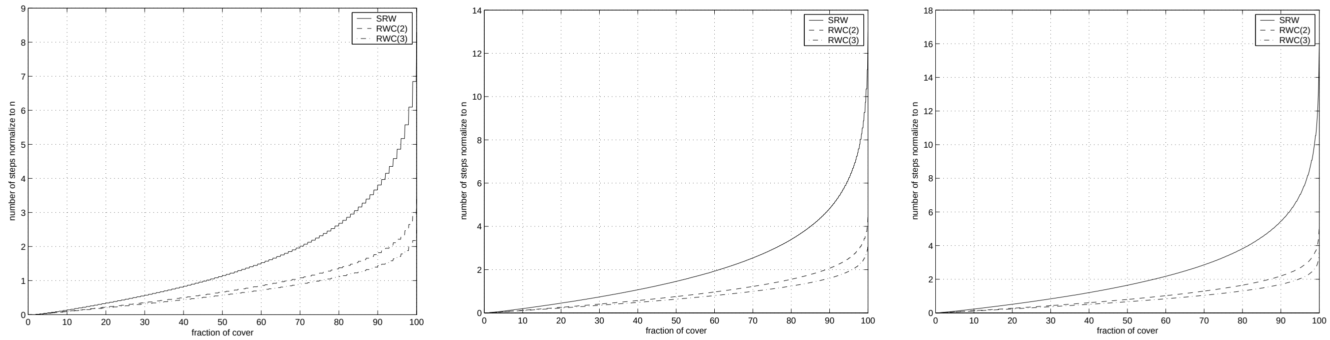
5. **Speed of mixing time**: At each step t we take the probability to be at node v as $\frac{c^{t+1}(u)}{t}$ and find the variation distance of this $t'th$-step distribution from the stationary distribution. What is the affect of choice on the variation distance and the corresponding mixing time (the time at which the variation distance goes below some ϵ)?

5. RANDOM GEOMETRIC GRAPHS

As mention earlier random geometric graphs are the most popular graph model for random wireless networks. To give a flavor of the improvements achieved by random walk with choice, we present data from $10,000$ random walk runs on an instance of $\mathcal{G}(900, 0.0981)$.

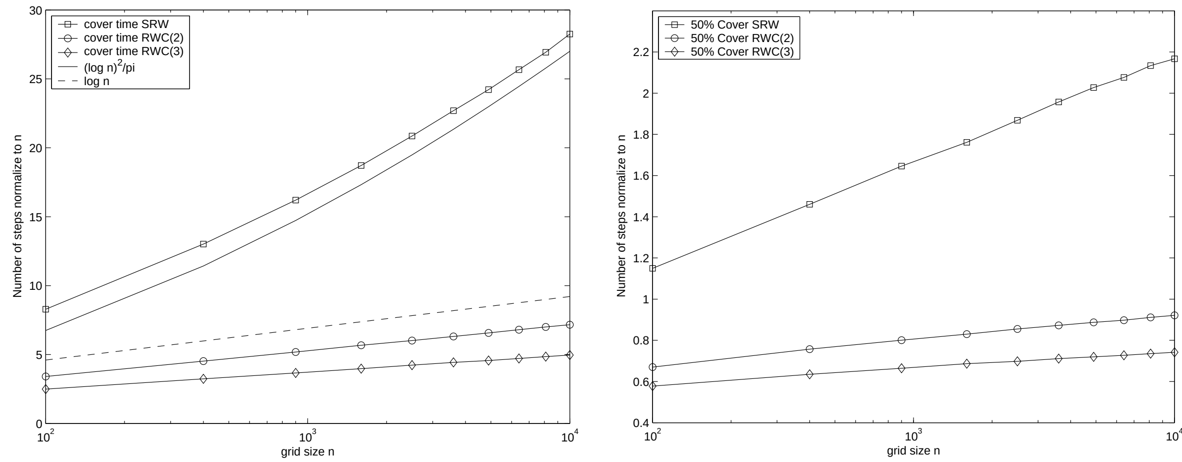
Fig 1 compares the histograms of cover times for the simple random walk with random walk with choices 2 and 3. Table 1 presents statistical data of these distributions. Note that the x axis in all three figure is set fo be from the minimum cover time of the random walk with choice 3 to the time that is larger than 99.9% of the cover times of the simple walk. The strong affect of choice on the distribution is clear from the figure and the table. It seems as the choice eliminates the heavy tail of the distribution and makes it more concentrated around its mean. This property is extremely important in practice as one want to avoid very long random-walk-based queries even if this happens only occasionally. In centralized random walk application (e.g. solving satisfiability) the heavy tail can be eliminated by rapid restarts of new walks in the case where retrieving an answer takes too long. In distributed systems there is a problem: while the base station can issue a repeated query if the random walk doesn't return fast (i.e. rapid restart), it cannot terminate the long walk which is somewhere in the network, and it will continue to move and consume energy.

The expected load balancing at cover time is shown in Fig 2. From left to right, the figure shows the expected number of visits to the $i'th$ most visited node at cover time. The first node on the left is the most visited node and rightmost node always has one visit at cover time. Note that the total number of visits (or the area under each curve) is the expected cover time. Again we can clearly see the reduction in cover time as a result of choice. Moreover, not only is there a large improvement in the most visited node, but the visits are distributed much more evenly. Intuitively it looks like the use of choice pushes "down" the most loaded bin causing the load to be distributed more evenly.



(A) (B) (C)

Figure 3: The expected cover time progress on a grid (normalized by grid size) for the simple random walk, and choice of 2 and 3. (A) 100 nodes (B) 400 nodes (C) 900 nodes



(A) (B)

Figure 4: The cover time and 50% cover time for different grid sizes. (A) cover time (B) 50% cover time

To obtain more systematic results, and to eliminate the noise created by the generation of random graphs, we next turn to determininstic mesh grids (with wrap-around into a torus to avoid edge effects). This allows us to more carefully check the asymptotic behavior of choice across different graphs size.

6. GRIDS

6.1 Expected Cover and Partial Cover Time

Fig 3 present the expected cover time progress up to full cover for meshes of size $n = 100, 400, 900$. The results are based on $10,000$ runs. In all cases we can see the improvement in cover and partial cover times as well as the diminishing returns type of behavior. Choice of 2 gives a large improvement compare to the simple random walk, but for $d = 3$ the gain is much smaller. We analytically know that for the simple random walk the partial cover is an order less than the cover time (i.e. $O(n \log n)$ instead of $O(n \log^2 n)$), meaning that most of the time in the cover process is "wasted" on the last few nodes. We observe that the same type of behavior is also presented by random walk with choice.

6.1.1 Different grid sizes

Fig 4 is one of the most significant figures in this work, and its results are surprising. Part (A) shows the expected cover time for meshes varying from size 100 to 10000. Note that the x axis is on log scale and the y axis is the expected number of steps to cover normalized by n. As we expected the simple random walk gives a cover time of $\Theta(n \log^2 n)$ which results in $\Theta(\log^2 n)$ curve in the figure. More interesting is the cover time $C_{G_n^2}^{\mathrm{RWC}(2)}$, of the random walk with choice of 2. It seems to have a lower order of $O(\log n)$ which implies a cover time of order $O(n \log n)$. This suggests that a choice of 2 on the 2-dimensional mesh achieves *optimal cover time*, the same order of cover as the complete graph. Selecting $d = 3$ doesn't seems to offer significant further improvement (in any case improvement in order is impossible if indeed $d = 2$ already gives optimal cover time). Part (B) of Fig 4 displays the expected time to cover 50% of the graph. For the simple random walk we know that partial cover is $O(n \log n)$ and this what the figure shows. For the choice of 2, it is harder to conclude what is the order, but it doesn't seem to be optimal partial cover time. Recall that optimal partial cover time is linear which should result in a constant line since we normalize by n. The partial cover of the random walk with choices $d = 2, 3$ is not a constant and therefore does

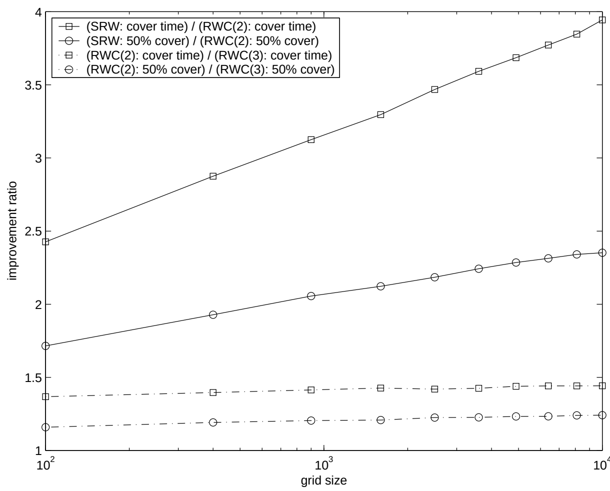


Figure 5: The improvement ratio of cover time for the choices of 2 and 3 for different grid sizes

not appear optimal. Some interpolation suggests a behavior of $O(n \log \log n)$ but this is highly speculative.

Fig. 5 presents the improvement ratios $I_2(n)$ and $I_3(n)$ for cover time achieved by random walk with choice. Clearly we notice that the improvement ratio for $d = 2$ is non-constant, supporting the claim that there is an unbounded improvement in the cover time. On the other hand $I_3(n)$ behaves as a constant, demonstrating concretely the diminishing returns. The results are similar for the 50% cover time; there seems to be an unbounded improvement ratio for a choice of 2, and a constant for a choice of 3. Nevertheless, the improvement ratio for partial cover is smaller than the one for cover time.

6.2 Cover Time Distribution

As in the case of the random geometric graph, Fig. 6 shows a consistent behavior in term of the cover time distribution for different grid sizes. For all three sizes choice is reducing the cover time as well as the variance. Choice eliminates the heavy tail and seems to change the distribution envelop. The statistical data is summarize in Table 2.

6.3 Load Balancing

Reproducing the load balance figure of the random geometric graph, Fig 7 presents the same behavior for different grid sizes. The effect of choice is seen clearly as flattening the load on nodes. Note that the y axis is normalized such that the max bin in each sub-figure is 1. The original results on the power of choice were stated in term of the most visited bin (after n balls, or n random walk steps), proving a non constant improvement on the ratio between the most visited node in the random walk with choice of 2 compare to the simple random walk. Does something similar happens on grids at cover time?

Fig 8 gives a positive answer to this question. It presents the improvement ratios $L_2(n)$ and $L_3(n)$. Our experiments show that the radio between choice of 2 and the simple walk, $L_2(n)$, is unbounded and seems to be of the order

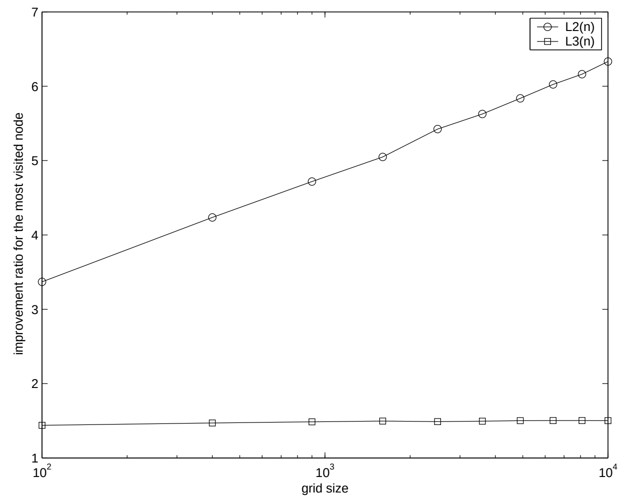


Figure 8: The improvement ratios, $L_2(n)$ and $L_3(n)$, of the most visited node in SRW, RWC(2) and RWC(3) for different grid sizes

of $O(\log n)$. On the other end, $L_3(n)$, the improvement between choice of 2 and 3 is a constant, similarly to what we observed for the improvement ratio of the cover time. This figure, as before, confirms our observation that the addition of choice has a large effect on the final outcome of the random walk.

6.4 Mixing Rate

As stated before, the mixing time is another key metric of interest. Since our graphs are regular, the stationary distribution of the random walk is the uniform distribution which, by definition, balances the load (number of visits) at mixing time. Therefore when the mixing time is smaller, the faster $P^t(x, \cdot)$ (i.e. the distributions of states at time t) coverage to the uniform distribution, and we should expect a better load balancing at cover time. Fig 9 plots the expected variation distance at step t until cover time (presented as fraction of cover) between $P^t(x, \cdot)$ and the uniform distribution. For the three grids we observe the impact of choice on the rate by which the variation distance decreases. At the start of the random walk, many new nodes are been visited, which decreases the variation distance “fast”; later, when discovering new nodes takes longer, the rate in which the variation distance decrease is “slower”. From Fig 9 it seems as choice extends the time for which the walk often discovers new nodes and the variation distance decreases fast. This behavior results in a smaller variation distance at cover time for random walks with choice.

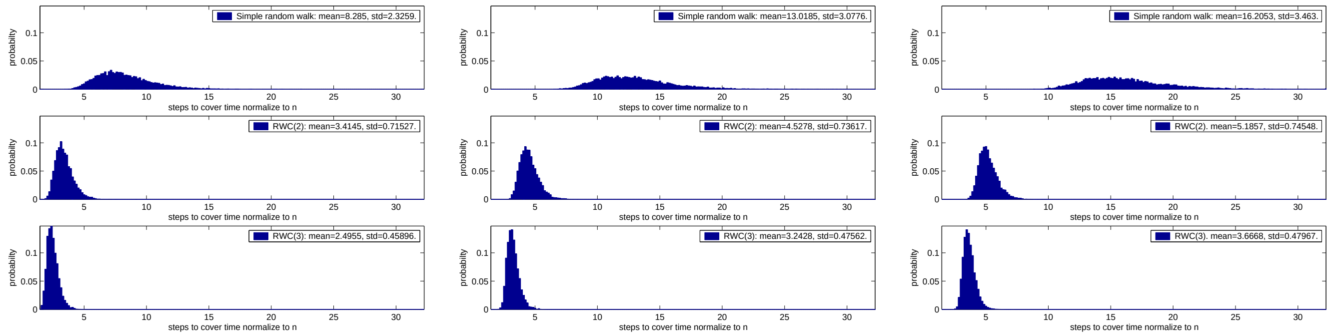
7. COMPARING THE GRID AND THE HYPER-CUBE

In order to validate our results from Fig. 4 on the order of the cover time of the random walk with choice on grids we set out to compare those results with the cover time of a graph which has optimal cover time. We repeated the same set of experiment on the hyper-cube, H_n, and compared the results with the grids.

Fig 10 shows the cover time for hyper-cube and grid of dif-

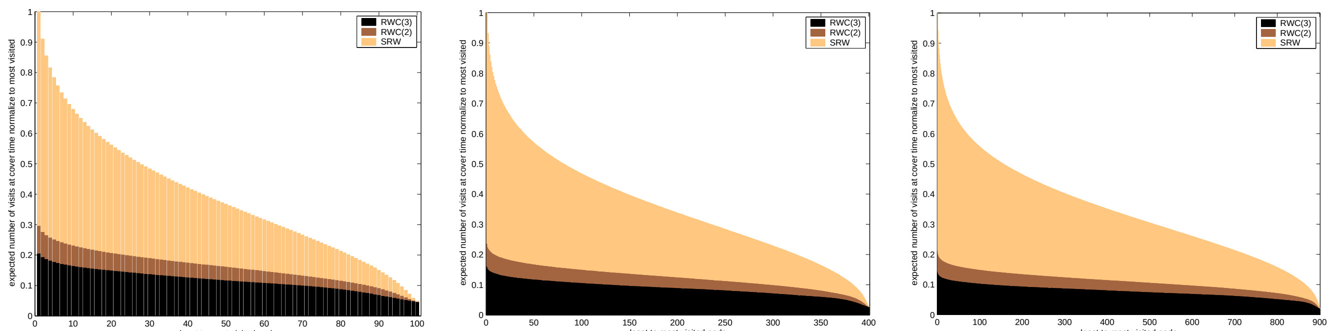
Table 2: **Mean and variance of cover times on grids (normalized by n)**

Walk Type	$n = 100$			$n = 400$			$n = 900$		
	mean	std	99%	mean	std	99%	mean	std	99%
SRW	8.285	2.325	15.420	13.018	3.077	22.625	16.205	3.463	26.460
RWC(2)	3.414	0.715	5.560	4.527	0.736	6.780	5.185	0.745	7.404
RWC(3)	2.495	0.458	3.870	3.242	0.475	4.692	3.666	0.476	5.082



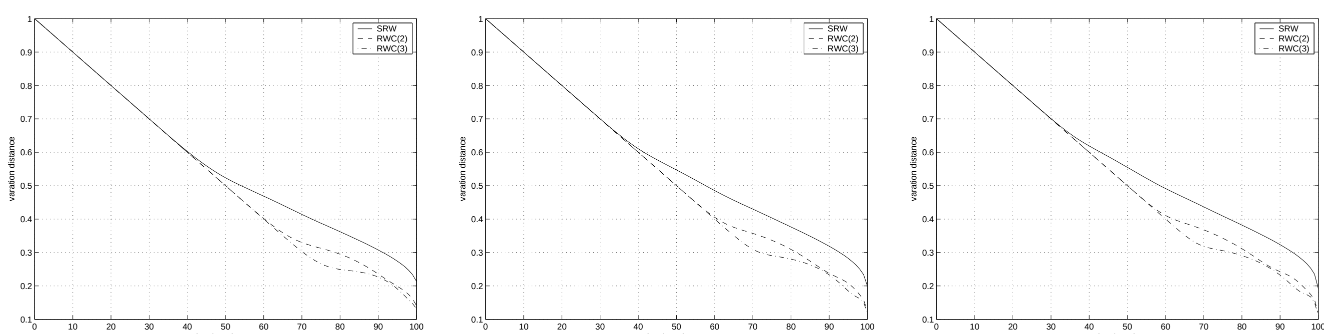
(A) (B) (C)

Figure 6: The distribution of the cover time on a grid as an histogram from 10000 runs for the simple random walk, and choice of 2 and 3. (A) 100 nodes (B) 400 nodes (C) 900 nodes



(A) (B) (C)

Figure 7: The expected load balance as number of visits at cover time on a grid. Average over 10000 runs for the simple random walk, and choice of 2 and 3. (A) 100 nodes (B) 400 nodes (C) 900 nodes



(A) (B) (C)

Figure 9: The variation distance progress until cover time on a grid average over 10000 runs for the simple random walk, and choice of 2 and 3. (A) 100 nodes (B) 400 nodes (C) 900 nodes

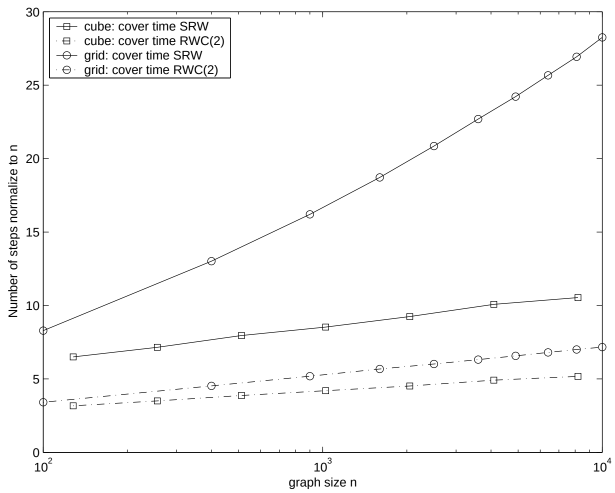


Figure 10: The cover time of the different sizes cube and grid

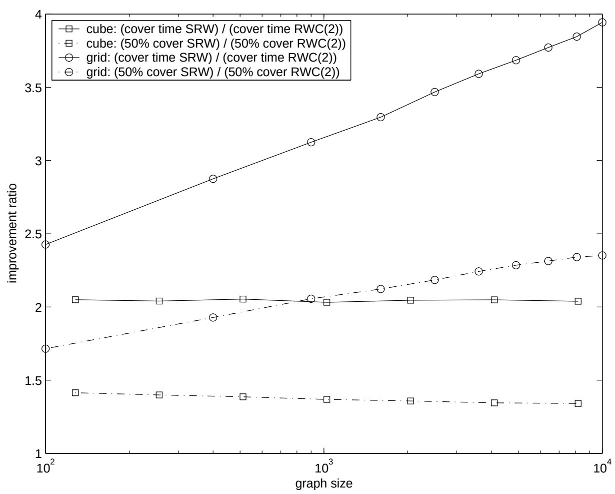


Figure 11: The improvement ratio of cover time for the choice of 2 in the hyper-cube and grid for different sizes

ferent sizes. The cover time of the simple random walk on the grid, $C^{\mathrm{SRW}}_{G^2_n}$, behaves as $O(\log^2 n)$ as we saw before. Regarding the hyper-cube, $C^{\mathrm{SRW}}_{H_n}$, behaves as $O(\log n)$ as we know analytically. The interesting result here is that the cover time of the walk with choice 2 on the grid, $C^{\mathrm{RWC(2)}}_{G^2_n}$, is less than the cover time of the simple random walk on the hyper-cube, $C^{\mathrm{SRW}}_{H_n}$. When interpolating this two line as straight lines (on the log scale), $C^{\mathrm{RWC(2)}}_{G^2_n}$ has a slightly smaller slop than $C^{\mathrm{SRW}}_{H_n}$. This gives, yet another, evidence for the order improvement of cover time on the grid with choice of 2.

We proved that the improvement ratio for the complete graph is constant. Since the hyper-cube has cover time of the same order of the complete graph we expect that similar results will apply to it. Fig 11 validates this intuition. It compares the improvement ratio of the cover time of the grid and the hyper-cube. We stated earlier that the improvement ratio for the grid is unbounded for both the cover and partial cover times. On the other hand for the hyper-cube, the figure shows that both for the cover and partial cover time the improvement ratio is constant. This is the case because both the cover and partial cover are optimal for the hyper-cube.

8. CONCLUSIONS

It is of fundamental interest to understand and enhance the behavior of simple low-state protocols for wireless networks such as query and routing mechanisms based on random walks. Motivated by the successful use of the power of choice technique for load-balancing problems, we have proposed a novel random walk with choice in this work. In this modified random walk algorithm, at each step the least visited among a set of randomly selected neighbors is chosen as the next node. The intuition behind this idea is that this choice will push the walk to visit less visited areas in the graph in order to improve upon the cover time. Our analytical results for the completed graph shows that when choosing between a constant number of neighbors we will have a constant improvement in the cover time. This suggests that for any graph with cover time on the order of the complete graph we should expect at most a constant improvement in the cover time. In particular, we should expect this to be the case for random geometric graph with radius $r \geq \sqrt{8} r_{\mathrm{con}}$. It is an open question whether for a lower radius the improvement will be unbounded. Our simulation results suggest that the effect of random walk with choice is larger for graphs that have non-optimal cover time. For 2-dimensional grid networks, we observed via simulations that the random walk with choice can offer unbounded improvement in cover time and the number of visits to the most visited node at cover time. We formulate this observation in the following conjecture:

CONJECTURE 1. *For the 2-dimensional mesh, G^2_n,*

$$C^{RWC(2)}_{G^2_n}(n) = o(C^{SRW}_{G^2_n}(n)) = o(n \log^2 n)$$

or in words, the cover time of the random walk with choice 2 is an order less than the simple random walk.

It will be challenging to prove this conjecture as well as other theoretical results for the random walk with choice regarding the distribution of cover time and the load balance at cover time.

At any rate, our simulation results give a strong evidence that incorporating choice into random-walk-based query or routing applications for wireless networks can provide significant performance improvements.

9. REFERENCES

[1] M. Alanyali, V. Saligrama, and O. Sava. A random-walk model for distributed computation in energy-limited network. In *In Proc. of 1st Workshop on Information Theory and its Application*, San Diego, 2006.

[2] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. 1999. Unpublished.

http://stat-www.berkeley.edu/users/aldous/RWG/book.html.

[3] D. J. Aldous. Lower bounds for covering times for reversible Markov chains and random walks on graphs. *J. Theoret. Probab.*, 2(1):91–100, 1989.

[4] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979)*, pages 218–223. IEEE, New York, 1979.

[5] C. Avin and C. Brito. Efficient and robust query processing in dynamic environments using random walk techniques. In *Proc. of the third international symposium on Information processing in sensor networks*, pages 277–286, 2004.

[6] C. Avin and G. Ercal. On the cover time of random geometric graphs. In *Proc. Automata, Languages and Programming, 32nd International Colloquium, ICALP05*, pages 677–689, 2005.

[7] Y. Azar, A. Broder, A. Karlin, and E. Upfal. Balanced allocations. In *Proc. 26th ACM Symp. Theory of Computing*, pages 593–602, 1994.

[8] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *Proc. of the 1st ACM Int. workshop on Wireless sensor networks and applications*, pages 22–31. ACM Press, 2002.

[9] A. Broder and A. Karlin. Bounds on the cover time. *J. Theoret. Probab.*, 2:101–120, 1989.

[10] A. K. Chandra, P. Raghavan, W. L. Ruzzo, and R. Smolensky. The electrical resistance of a graph captures its commute and cover times. In *Proc. of the twenty-first annual ACM symposium on Theory of computing*, pages 574–586. ACM Press, 1989.

[11] C. Cooper and A. Frieze. The cover time of sparse random graphs. In *Proceedings of the fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-03)*, pages 140–147, Baltimore, Maryland, USA, 2003. ACM Press.

[12] S. Dolev, E. Schiller, and J. Welch. Random walk for self-stabilizing group communication in ad-hoc networks. In *Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS'02)*, page 70. IEEE Computer Society, 2002.

[13] U. Feige. A tight lower bound on the cover time for random walks on graphs. *Random Structures and Algorithms*, 6(4):433–438, 1995.

[14] U. Feige. A tight upper bound on the cover time for random walks on graphs. *Random Structures and Algorithms*, 6(1):51–54, 1995.

[15] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *in Proc. 23 Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). to appear*, 2004.

[16] C. P. Gomes, B. Selman, N. Crato, and H. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1-2):67, Feb 2000.

[17] M. Jerrum and A. Sinclair. The markov chain monte carlo method: an approach to approximate counting and integration. In *Approximations for NP-hard Problems, Dorit Hochbaum ed.*, pages 482–520. PWS Publishing, Boston, MA, 1997.

[18] J. Jonasson. On the cover time for random walks on random graphs. *Comb. Probab. Comput.*, 7(3):265–279, 1998.

[19] J. Jonasson and O. Schramm. On the cover time of planar graphs. *Electronic Communications in Probability*, 5:85–90, 2000.

[20] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2003.

[21] L. Lovász. Random walks on graphs: A survey. In *Combinatorics, Paul Erdős is eighty, Vol. 2 (Keszthely, 1993)*, volume 2 of *Bolyai Soc. Math. Stud.*, pages 353–397. János Bolyai Math. Soc., Budapest, 1996.

[22] P. Matthews. Covering problems for Brownian motion on spheres. *Ann. Probab.*, 16(1):189–199, 1988.

[23] M. Mitzenmacher, A. W. Richa, and R. Sitaraman. The power of two random choices: A survey of techniques and results. In R. et al, editor, *In Handbook of Randomized Computing*, volume I, pages 255–312. Kluwer Academic Press, 2001.

[24] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

[25] R. Pemantle. Vertex-reinforced random walk. *Probability Theory and Related Fields*, 92(1):117–136, 1992.

[26] J. S. Rosenthal. Convergence rates for markov chains. *SIAM Rev.*, 37(3):387–405, 1995.

[27] N. Sadagopan, B. Krishnamachari, and A. Helmy. Active query forwarding in sensor networks (acquire). *To appear in Elsevier journal on Ad Hoc Networks*, 2003.

[28] S. D. Servetto and G. Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. In *Proc. of the first ACM Int. workshop on Wireless sensor networks and applications*, pages 12–21. ACM Press, 2002.

[29] A. Sinclair. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1:351–370, 1992.

[30] S. Volkov. Vertex-reinforced random walk on arbitrary graphs. *Annals of Probability*, 29(1):66–91, Jan 2001.

[31] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Robotic exploration, brownian motion and electrical resistance. *Lecture Notes in Computer Science*, 1518:116–130, 1998.

[32] D. Zuckerman. A technique for lower bounding the cover time. In *Proc. of the twenty-second annual ACM symposium on Theory of computing*, pages 254–259. ACM Press, 1990.