Available online at www.sciencedirect.com

ScienceDirect

Computer Networks 52 (2008) 44–60

Computer Networks

www.elsevier.com/locate/comnet

The power of choice in random walks: An empirical study ☆

Chen Avin [a,*,1], Bhaskar Krishnamachari [b]

[a] *Department of Communication Systems Engineering, Ben Gurion University of The Negev, Beer Sheva, 84105, Israel*

[b] *Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, United States*

Available online 29 September 2007

Abstract

In recent years random-walk-based algorithms have been proposed for a variety of networking tasks. 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 this algorithm, instead of selecting just one neighbor at each step, 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 energy consumption and the cover time and load-balancing properties of random walks. We also systematically 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.

Keywords: Power of choice; Random walks; Wireless networks

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. 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 use of random walk for querying/searching, routing and self-stabilization in wireless networks, peer-to-peer networks, and other distributed systems [12,29,8,28,5,15,1].

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

☆ This work has been supported in part by NSF through grants CNS-0435505 and CNS-0347621, and by Bosch RTC.

[*] Corresponding author.

E-mail addresses: avin@cse.bgu.ac.il (C. Avin), bkrishna@usc.edu (B. Krishnamachari).

[1] The work was done in part while the first author was at USC.

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.09.012

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 traces 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. Topology-driven algorithms are at a disadvantage for such networks as they incur high overhead to maintain up-to-date topology and routing information and also have to provide recovery mechanisms for critical points of failure (e.g., cluster heads, nodes close to the root in a spanning tree). By contrast, 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 wandering randomly in the network may seem overly simplistic and highly inefficient, many encouraging results that prove its comparability with other approaches have been obtained over the years. Two basic properties of random walks 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 [18,32,21,5,15], and various methods of bounding the cover time of graphs have been thoroughly investigated [4,23,3,10,9,33]. An *optimal cover time* is a cover time of $\Theta(n \log n)$, the same order as the cover time of the complete graph. Several bounds on the cover time of random walks on different classes of graphs have been obtained with many positive results [10,9,19,20,11].

One class of graphs that has received particular attention in this context is the class of random geometric graphs which are most suitable for modeling wireless networks. A random geometric graph is a graph $G(n, r)$ resulting from placing n points uniformly at random on the unit square and connecting two points if and only if 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 ad-hoc and sensor networks. Recently it has been proven that, when $r = \Theta(r_{con})$ then *w.h.p.*[2] $G(n, r)$ has optimal cover time of $O(n \log n)$ and optimal partial cover time of $O(n)$ [6] where r_{con} growing as $O\left(\sqrt{\frac{\log n}{\pi n}}\right)$ is the critical radius to guarantee connectivity *w.h.p.* [17].

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 that also 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). 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 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.* [25]. 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 *w.h.p.* So with only a little more work at each step, (choosing two bins instead of one) we see a large improvement. Notice that increasing d further 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

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

and applications to hashing, load balancing in distributed systems, and low-congestion routing, among others [24].

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 step to the least visited node among them.[3] Note that the 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[4] 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 mesh networks (i.e., the 2-dimensional grid on a torus), choice seems to improve the cover time and the load on the most visited node by an unbounded factor. Specifically, while the cover time of the n nodes mesh is known to be Hðnlog^{2}nÞ, our simulations show 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 Hðn log nÞ. 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. Sections 5–7 present 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 $n =$ jVj and $m =$ jEj. For v 2 V let NðvÞ ¼ fu 2 VjðvuÞ 2Eg be the set of neighbors of v and d(v)=jN(v)j the degree of v. A d-regular graph is a graph in which the degree of all the nodes is d.

The *simple random walk*, SRW, is a walk where the next node is chosen uniformly at random from the set of neighbors. That is when the walk is at node v the probability to move in the next step to u is Pðv; uÞ ¼$\frac{1}{\text{dðvÞ}}$ for ðv; uÞ 2 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 2 V let C_v be the expected number of steps for the simple random walk starting at v to visit all the nodes in G, and the cover time of G is $C_G = \max_v C_v$. The *cover time* of graphs and methods of bounding it have been extensively investigated [23,3,10,9,33,4]. Results for the cover time of specific graphs vary from the *optimal cover time* of Hðn log nÞ associated with the complete graph to the worst case of Hðn^{3}Þ associated with the lollipop graph [14,13]. The known best cases correspond to dense, highly connected graphs; on the other hand, when connectivity decreases and bottlenecks exist in the graph, the cover time increases. In this paper we consider three types of graphs:

1. *random geometric graphs*: Gðn; rÞ – of size n and radius r. We consider the cases of different sizes and r between $1.5r_{\text{con}}$ and $6r_{\text{con}}$. It is known that for r P $^{\text{p}}$ 8pr_{con}Gðn; rÞhas, *w.h.p.*, optimal cover time [6].
2. *meshes*: G^2_n – the 2-dimensional mesh (i.e., grid on the torus) of size n. It is known to have non-optimal cover time of Hðnlog^{2}nÞ[10].
3. *hyper-cubes*: H_n – the hyper-cube which is the $\log_2(n)$-dimensional grid of size n. H_n is known to have an optimal cover time [25].

The *partial cover time* [5] is the expected time taken by a random walk to visit a constant fraction

[3] A related modification to random walks called *Vertex-Reinforced Random Walks*, VRRW, was proposed in [26,31]. However, in VRRW the walk prefers the *most* visited nodes, without choice. Further, VRRW is not studied in the context of cover time.

[4] In the simple random walk the next step is independent of past steps; this is not the case for the random walk with choice.

of the nodes and is defined formally as follows: 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 (starting from the worst case node). Let H_{uv} be the *hitting time*, the expected time for a random walk starting at u to arrive at 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 $\Theta(n \log n)$[33], so partial cover becomes $\Theta(n \log n)$.

Note that the analytical results about cover and partial cover are about the expected time. Less is known about the distribution of the cover time, but it has been observed that in some contexts random-walk-based algorithms have a "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 $\mathcal{M}$ 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 $\mathcal{M}$, 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 $\mathcal{M} = (\Omega, P)$ over a connected graph $G = (V,E)$ has a stationary distribution π such that, for any node $q \in V$ [22]

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

where m is the number of edges in G. Further, when the underlying graph G is d regular, the stationary distribution is the uniform distribution [22]

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

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 $\mathcal{M}$ may be used to sample over state space Ω with respect to the 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 [30,27]

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

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

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 corresponds to the most visited node after n steps of the walk. The idea we set out 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:

Algorithm 1

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 $\frac{c_t(u)+1}{d(u)}$ (break ties in an arbitrary way).

A few remarks are in place: If the graph is regular, the walk steps to the least-visited neighbor; if not, the walk steps to the node that is farthest away from its stationary distribution $\pi(u)$. Clearly for $d=1$ this is the simple random walk. For $d>1$ the Markov property does not hold anymore since the current step depends on all the 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 effect 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^A denote the cover time of algorithm 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 $C_{K_n}^{\mathrm{RWC}(d)}$ is

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

Proof. Let h_n be the harmonic sum $h_n = \sum_1^n i^{-1} \approx \log n$. It is well known that $C_{K_n}^{\mathrm{SRW}} = nh_{n-1}$. 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 $\left(\frac{m}{n}\right)^2$ (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:

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

taking $m = n - x$ we get:

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

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 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 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 $G(n,r)$, (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 graphs with a uniform stationary distribution. For the random geometric graph, $G(n,r)$, we used $n=900$, 1600, 2500 and the radius r varies from $r=1.5r_{\mathrm{con}}$ to $r=6r_{\mathrm{con}}$ where $r_{\mathrm{con}} = \sqrt{\frac{\log n}{\pi n}}$.

On the grid we run the simulation for $n = \{100x^2 | x = 1, 2, 3, \ldots, 10\}$. For the hyper-cube we used $n = \{2^x | x = 7, 8, \ldots 13\}$. For each graph we execute the $RWC(d)$ for $d=1,2,3$ and in each case we average over many runs.

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 $d \geq 2$ and a graph G of size n is defined to be

$$I_d(n) = \frac{C_G^{\mathrm{RWC}(1)}}{C_G^{\mathrm{RWC}(d)}}.$$

What is the asymptotic behavior of $I_d(n)$ for the different graphs? From Lemma 1 we know that for the complete graph K_n, $I_d(n) = \frac{d}{d-1}$ as $n \to \infty$. Note that if $I_d(n) = \omega(1)$(i.e., an order larger than a constant) then $C_G^{\mathrm{RWC}(d)}$ is of a lower order than $C_G^{\mathrm{RWC}(1)}$. We will be mostly interested in $I_2(n)$, the improvement of random walk with power of choice 2, compare to the simple random walk. In the complete graph this improvement ratio is 2, could it be larger for other graphs? Similarly we define the improvement ratio for the partial cover (e.g. 50%) and ask the same questions.
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?
4. *Load balancing*: Load balancing is of crucial interest 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

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

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 at 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 one visit at cover time) and average over all runs. This yields the expected number of visits to the ith most visited node.
5. *Speed of mixing time*: At each step t we take the probability to be at node v as $\frac{c_t(v)}{t}$ and find the variation distance between this distribution at time t and the stationary distribution. What is the effect 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

5.1. Cover time distribution and load balancing

As mentioned 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 for these graphs, we present data from 10,000 random walk runs on an instance of $G(9000.0981)$ (i.e., $r = 2r_{\mathrm{con}}$).

Fig. 1 compares the histograms of cover times for the simple random walk (SRW) with the cover times for the random walk with choices 2 and 3 (i.e., RWC(2), RWC(3)). Table 1 presents statistical data of these distributions. Note that the x-axis in all three figure is set to 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

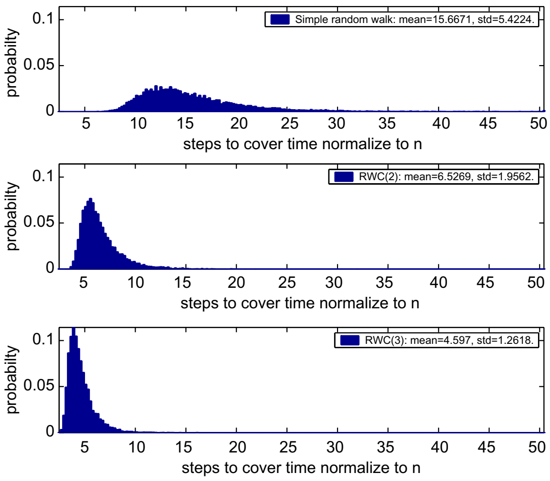


Fig. 1. The distribution of the cover time on $G(900, 0.0981)$ as an histogram from 10,000 runs.

Table 1
Statistical data for the cover time distribution of Gð900; 0:0981Þ (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

simple walk. The strong effect of choice on the distribution is clear from the figure and the table. It seems that 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 wants to avoid very long random-walk-based queries even if this happens only occasionally. In a 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, similar results could be obtained using a TTL timer, but still one wants to avoid such restarts as much as possible.

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 ith most visited node at cover time. The first node on the left is the most visited node and right-most 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 and, again, we can clearly see the reduction in cover time as a result of choice. With choice, the most visited node has many less visits (i.e., load) during cover, and moreover, the visits are distributed more evenly among all nodes. Intuitively it looks like allowing choice pushes "down" the bin with the highest load, causing the load to be distributed more uniformly.

5.2. Cover time as a function of the radius r

The results for the following two subsections were obtained in the following way: For each graph size n we generated six sets of random points in the unit square. For each such instantiation we generate graphs with radius r varying from $r = 1.5r_{\text{con}}$ to $r = 6r_{\text{con}}$ (on the same set of points). For each graph and choice of 1, 2 or 3 we run 100 walks. We average across all runs for the same n, r and choice, so each data point is the average of 600 walks. Fig. 3 gives example of random geometric graphs of size $n = 900$ and different radii.

Fig. 4a and b demonstrate the influence of the radius r on the cover and partial cover time. As r

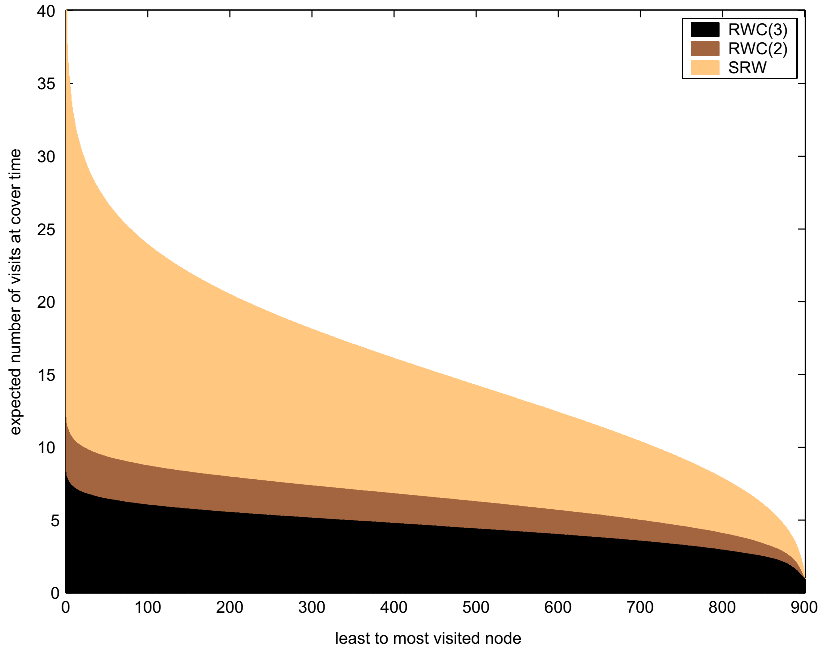


Fig. 2. The load balance at cover time for random geometric graph Gð900; 0:0981Þ.

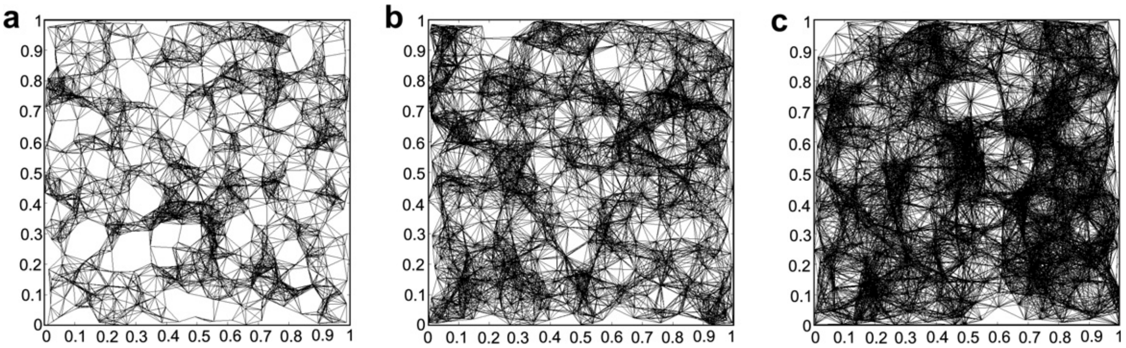


Fig. 3. An example of Gð900; rÞ(a) r ¼ 1:5r$_{\text{con}}$ (b) r ¼ 2r$_{\text{con}}$ (c) r ¼ 2:5r$_{\text{con}}$.

increases, the random geometric graph becomes more connected, the expected degree increases and one should expect that the cover time will decrease. As mentioned earlier it was recently shown that for r P p 8pr$_{\text{con}}$, the cover of random geometric graph is on the same order as the complete graph, so we expect that further increasing r will not reduce the cover time as much. On the other hand when r approaches r_{con} the graph becomes less connected and cover time should increase. This behavior is nicely shown in Fig. 4a. Note that when adding choice we observe a similar-type behavior, although it seems as the radius from which the improvement becomes negligible is much smaller. In Fig. 4b, that presents an example of partial cover time (i.e., visiting 80% of the nodes), we observe again similar behavior. Moreover, note that partial cover takes much fewer steps than full cover; in particular covering 80% of the nodes with random walk with choice of 3, takes approximately n steps (for large enough r).

Fig. 4 highlights some of the most interesting aspects of random walk with choice vs. the simple random walk. First, for all radii, random walk with choice achieves faster cover and partial cover time. Second, and more interestingly, in wireless networks this results in the following outcome: for a given cover time of the simple random walk, random with choice will achieve the same cover time with a smaller radius, which implies lower energy consumption. Since the transmission power is proportional to the distance squared, reducing the radius by half, for example, will results in using four times less energy. In the case of random geometric graphs, one can therefore use random walk with choice to not only reduce the total number of steps for cover or partial cover, but also to reduce the energy consumption by reducing the transmission radius.

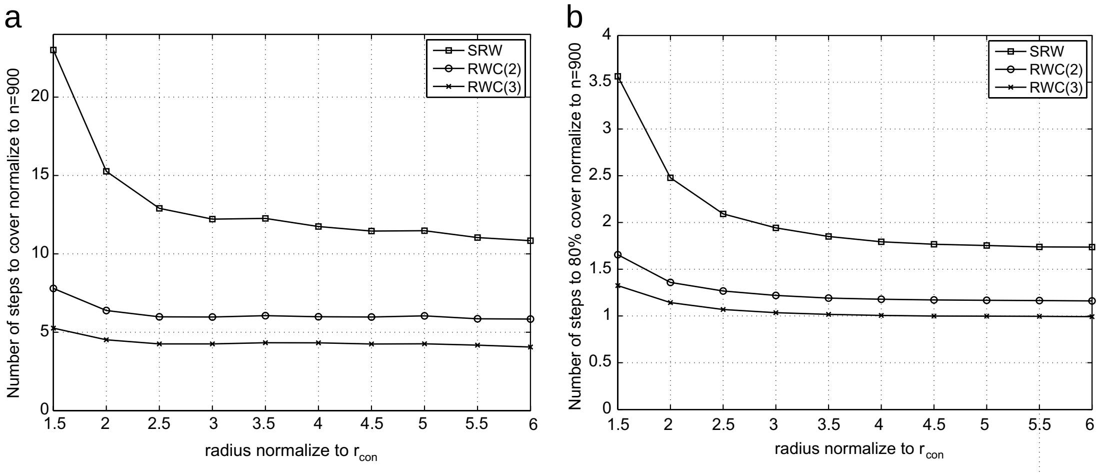


Fig. 4. The cover time of Gð900; rÞfor increasing radius and choice: (a) full cover time; (b) time to cover 80% of the nodes.

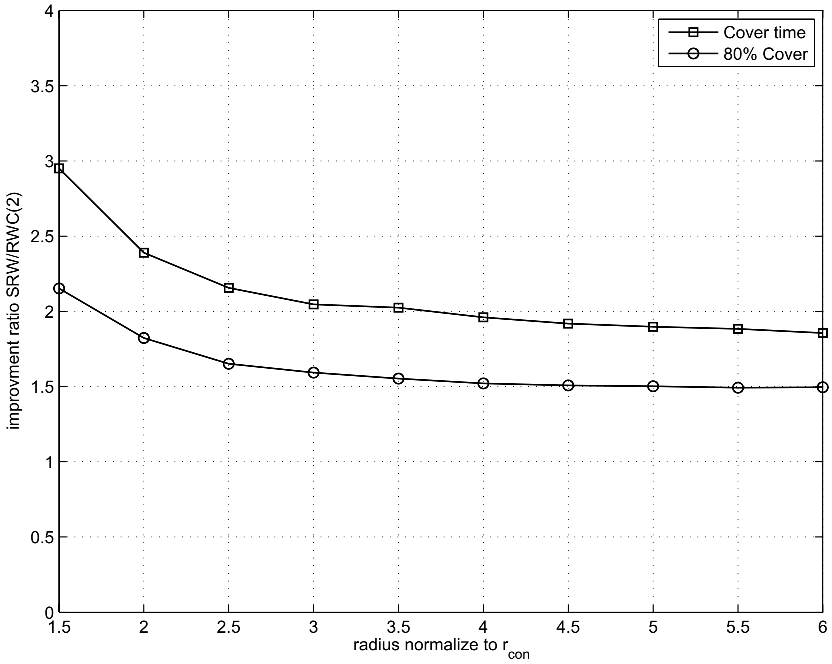


Fig. 5. The improvement ratio (SRW/RWC(2)) of Gð900; rÞfor increasing radius.

Fig. 5 presents the improvement ratio (i.e., SRW/RWC(2)) as a function of r. The intuition is the following: the better the cover time is, the smaller the improvement ratio will be, or in other words the less the random walk with choice can help. Recall from Fig. 4 that the cover time is decreasing (i.e., becoming better) when r is increasing, so Fig. 5 indeed confirms our intuition by showing that the improvement ratio decreases with r.

5.3. Different graph sizes

Fig. 6 repeats the results of Fig. 4a for two more sizes: $n = 1600$ and $n = 2500$. The results are similar and consistent: Cover time decreases with r, random walk with choice reduces the number of steps to cover for all radii and random walk with choice allows for the use of a smaller transmission radius. We also observe that cover time increases as the

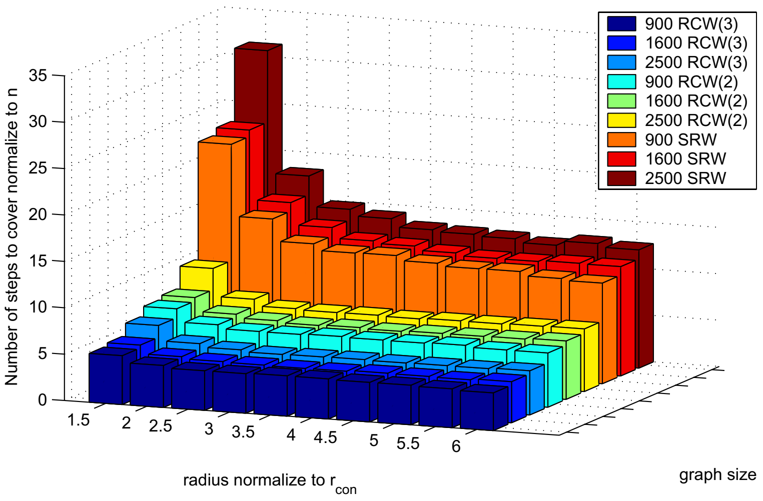


Fig. 6. Cover Time as a function of radius, size and choice.

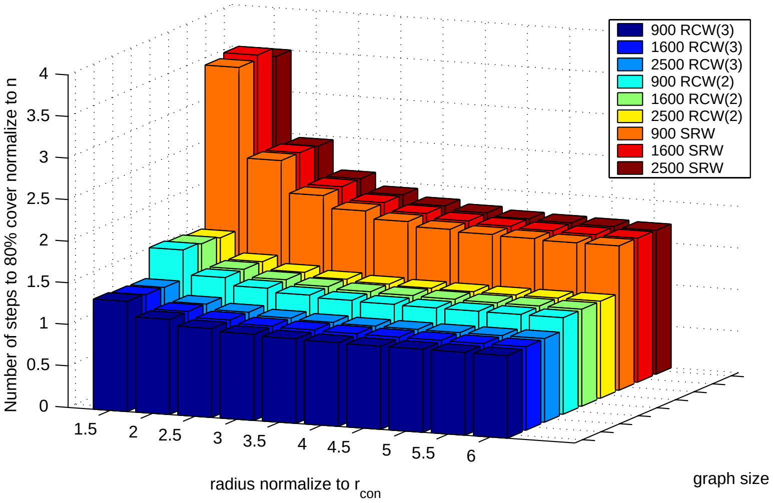


Fig. 7. Eighty percent of cover time as a function of radius, size and choice.

graph size increases, although the number of steps is normalized to n, the cover time is at least $n\log n$ for all r. Similarly Fig. 7 repeat the results of Fig. 4b with same conclusions regarding the random walk with choice. Note, however, that since we measure partial cover in this case, and since the number of steps is normalized to n, there is no increase in time to partial cover when the graph size increases. This is a consequence of the results which state that in random geometric graphs the partial cover time is $\Theta(n)$ for large enough r. This is nicely illustrated in the figure; in particular, it seems as though random walk with choice 3 achieves 80% cover within n steps.

We next turn to 2-D mesh grids (with wrap-around into a torus to avoid edge effects). Because this is a deterministic graph model, we are able to undertake a more systematic investigation, and carefully check the asymptotic behavior of choice across a larger number of different graph sizes.

6. Grids

6.1. Expected cover and partial cover time

Fig. 8 presents the expected cover time progress up to full cover for meshes of size $n = 400, 900$. The results are based on 10,000 runs. In all cases

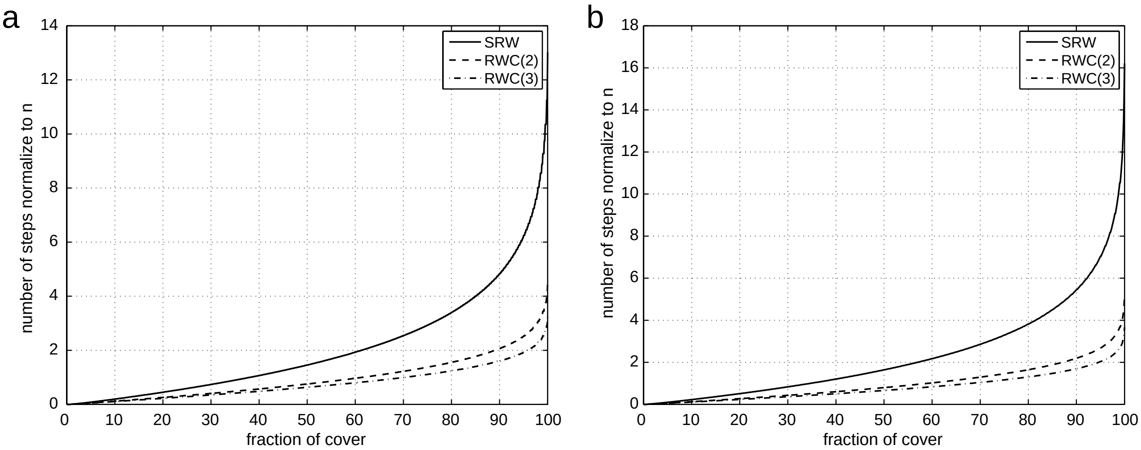


Fig. 8. The expected cover time progress on a grid (normalized by grid size) for the simple random walk, and choice of 2 and 3: (a) 400 nodes; (b) 900 nodes.

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 compared 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

From this point on all the results are based on the average of 1000 runs per graph and choice. Fig. 9 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 10,000. 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 $C^{\mathrm{RWC}(2)}_{G^2_n}$, 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 $d = 3$ does not seem 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. 9 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 is what the figure shows. For the choice of 2, it is harder to conclude what is the order, but it does not seem to be optimal partial cover time. Recall that optimal partial cover time is linear which should result in a constant line since we normalized by n. The partial cover of the random walk with choices $d = 2$, 3 is not a constant and therefore does not appear optimal. Some interpolation suggests a behavior of $O(n \log \log n)$ but this is highly speculative.

Fig. 10 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. 11 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 envelope. The statistical data is summarized in Table 2.

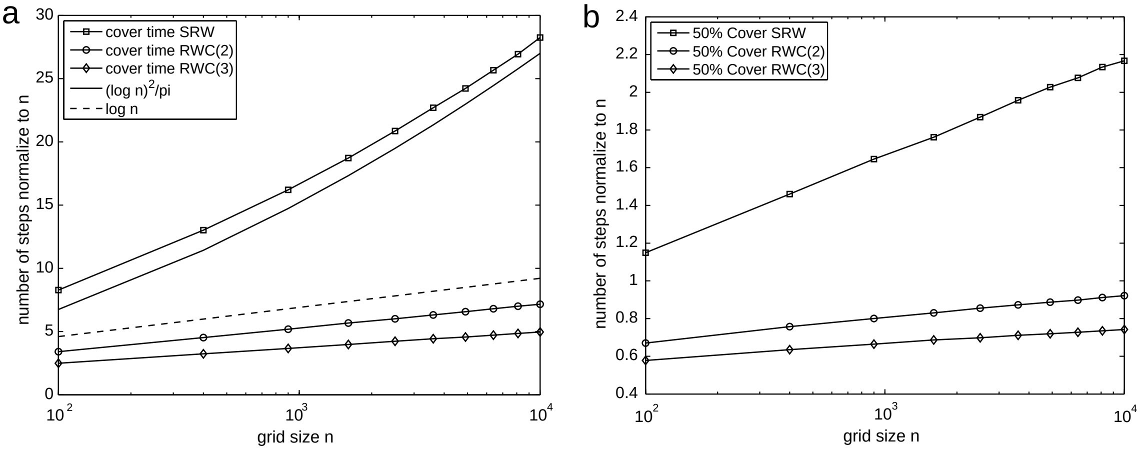


Fig. 9. The cover time and 50% cover time for different grid sizes: (a) cover time; (b) 50% cover time.

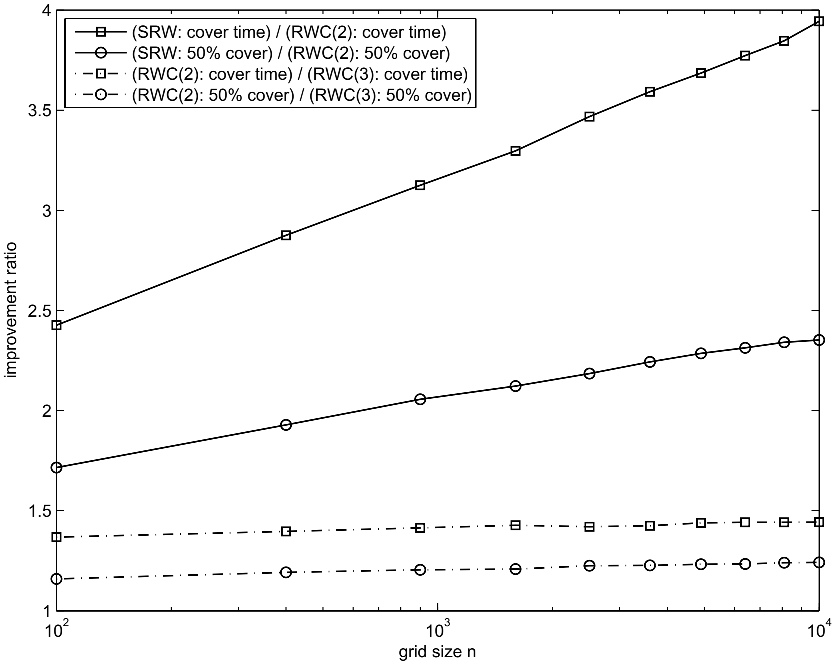


Fig. 10. The improvement ratio of cover time for the choices of 2 and 3 for different grid sizes.

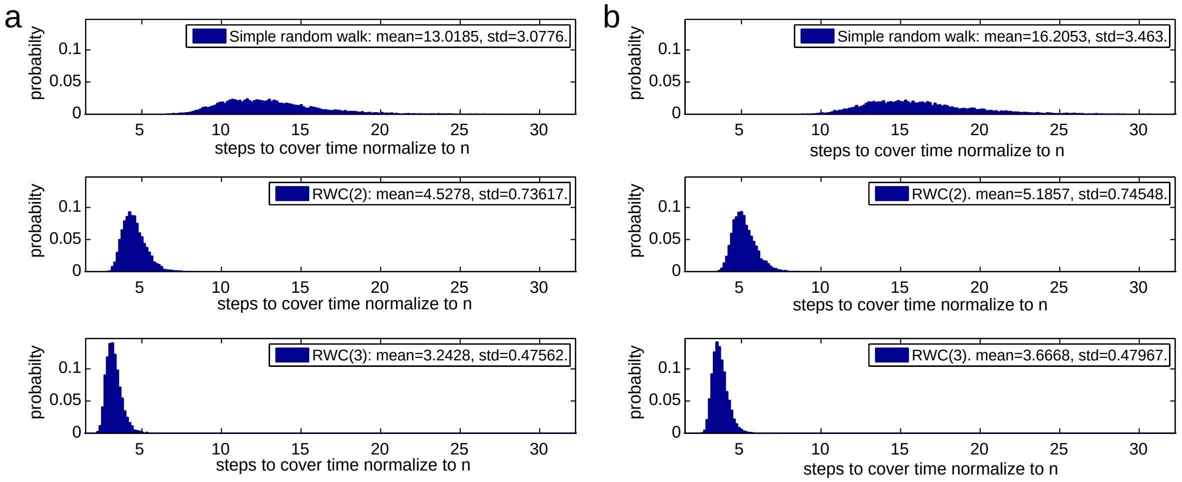


Fig. 11. The distribution of the cover time on a grid as an histogram from 10,000 runs for the simple random walk, and choice of 2 and 3: (a) 400 nodes; (b) 900 nodes.

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

Walk type	$n = 400$			$n = 900$		
	Mean	Std	99%	Mean	Std	99%
SRW	13.018	3.077	22.625	16.205	3.463	26.460
RWC(2)	4.527	0.736	6.780	5.185	0.745	7.404
RWC(3)	3.242	0.475	4.692	3.666	0.476	5.082

6.3. Load balancing

Reproducing the load balance figure of the random geometric graph, Fig. 12 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 compared to the simple random walk. Does something similar happen on grids at cover time?

Fig. 13 gives a positive answer to this question. It presents the improvement ratios $L_2(n)$ and $L_3(n)$. Our experiments show that the ratio

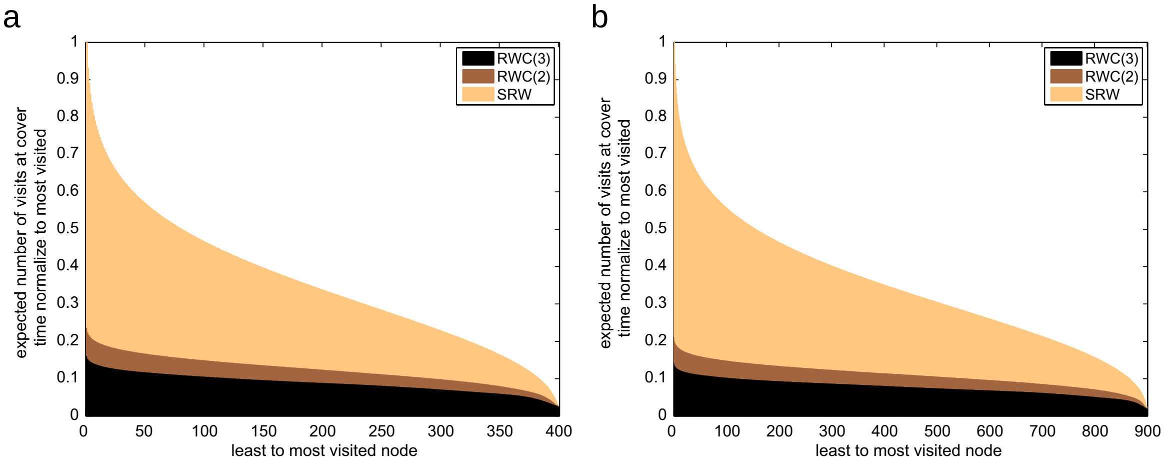


Fig. 12. The expected load balance as number of visits at cover time on a grid. Average over 10,000 runs for the simple random walk, and choice of 2 and 3: (a) 400 nodes; (b) 900 nodes.

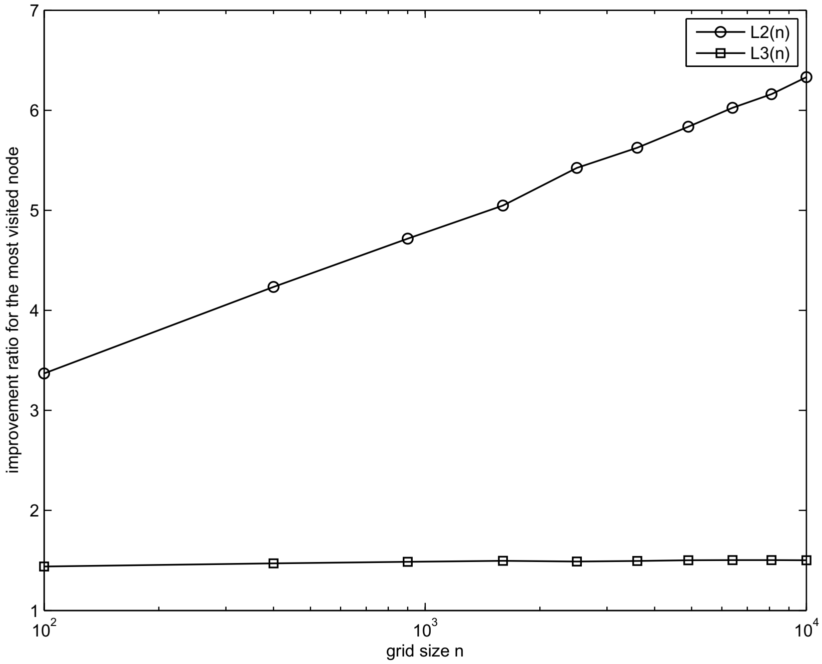


Fig. 13. The improvement ratios, L_2ðnÞand L_3ðnÞ, of the most visited node in SRW, RWC(2) and RWC(3) for different grid sizes.

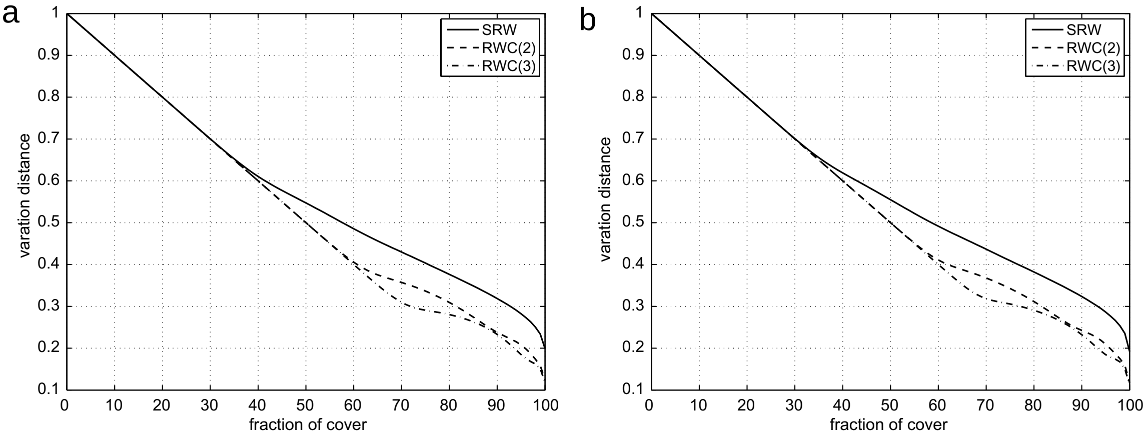


Fig. 14. The variation distance progress until cover time on a grid average over 10,000 runs for the simple random walk, and choice of 2 and 3: (a) 400 nodes; (b) 900 nodes.

between choice of 2 and the simple walk, $L_2(n)$, is unbounded and seems to be of the order of $O(\log n)$. On the other end, $L_3(n)$, the improvement between choice of 2 and 3 is a constant, similar 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 impact 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 that, by definition, balances the load (number of visits) at mixing time. Therefore, when the mixing time is smaller, the faster is $P^t(x,$Æ(i.e., the distributions of states at time t) converges to the uniform distribution; we should therefore expect a better load balancing at cover time. Fig. 14 plots the expected variation distance at step t until cover time (presented as fraction of cover) between $P^t(x,$Æand 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 being 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. 14 it seems that 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. Grids vs. hyper-cubes

In order to validate our results from Fig. 9 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 experiments on the hyper-cube, H_n, and compared the results with the grids.

Fig. 15 shows the cover time for hyper-cube and grid of different sizes. The cover time of the simple random walk on the grid, $C^{SRW}_{G^2_n}$, behaves as $O(\log^2 n)$ as we saw before. Regarding the hyper-cube, $C^{SRW}_{H_n}$, 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, $C^{RWCð2Þ}_{G^2_n}$, is less than the cover time of the simple random walk on the hyper-cube, $C^{SRW}_{H_n}$. When interpolating these two line as straight lines (on the log scale), $C^{RWCð2Þ}_{G^2_n}$ has a slightly smaller slope than $C^{SRW}_{H_n}$. This gives, yet more, 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 as the complete graph we expect that similar results will apply to it. Fig. 16 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

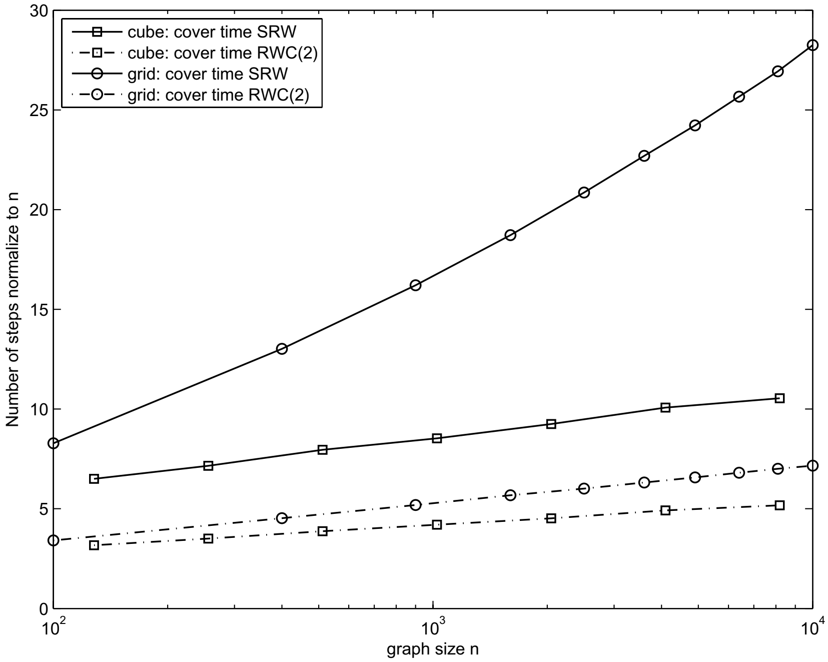


Fig. 15. The cover time of the different sizes cube and grid.

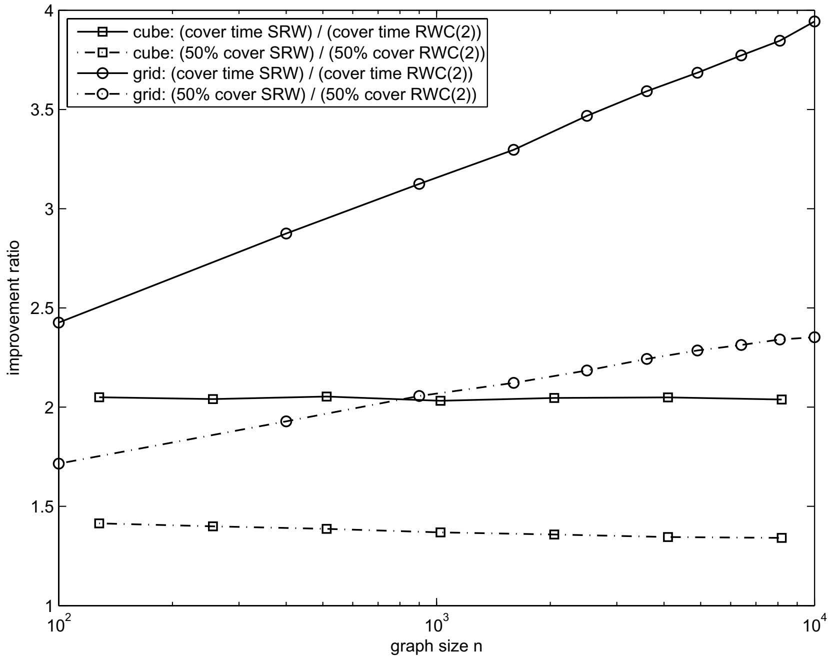


Fig. 16. The improvement ratio of cover time for the choice of 2 in the hyper-cube and grid for different sizes.

both the cover and partial cover times. On the other hand, the figure shows that for the hyper-cube, the improvement ratio for both cover and partial cover time is constant.

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 complete 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 $r \geq \sqrt{8\pi} r_{con}$. 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,

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

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

It is of great interest 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.

References

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

[2] D. Aldous, 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. Theor. Prob. 2 (1) (1989) 91–100.

[4] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, 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), IEEE, New York, 1979, pp. 218–223.

[5] C. Avin, 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, 2004, pp. 277–286.

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

[7] Y. Azar, A. Broder, A. Karlin, E. Upfal, E. Balanced allocations, in: Proc. 26th ACM Symp. Theory of Computing 1994, pp. 593–602.

[8] D. Braginsky, D. Estrin, Rumor routing algorithm for sensor networks, in: Proc. of the 1st ACM Int. workshop on Wireless sensor networks and applications 2002, ACM Press, pp. 22–31.

[9] A. Broder, A. Karlin, Bounds on the cover time, J. Theor. Prob. 2 (1989) 101–120.

[10] A.K. Chandra, P. Raghavan, W.L. Ruzzo, 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 1989, ACM Press, pp. 574–586.

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

[12] S. Dolev, E. Schiller, 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) IEEE Computer Society, 2002, p. 70.

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

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

[15] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks, INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, 7–11 March 2004, pp. 120–130.

[16] C.P. Gomes, B. Selman, N. Crato, H. Kautz, Heavy-tailed phenomena in satisfiability and constraint satisfaction problems, J. Autom. Reason. 24 (1–2) (2000) 67 (Feb).

[17] P. Gupta, P.R. Kumar, Critical power for asymptotic connectivity in wireless networks, in: Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, 1998, pp. 547–566.

[18] M. Jerrum, A. Sinclair, The Markov chain monte carlo method: an approach to approximate counting and integration, in: Dorit Hochbaum (Ed.), Approximations for NP-hard Problems, PWS Publishing, Boston, MA, 1997, pp. 482–520.

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

[20] J. Jonasson, O. Schramm, On the cover time of planar graphs, Electron. Commun. Prob. 5 (2000) 85–90.

[21] D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in: Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science 2003, pp. 482–491.

[22] L. Lovász, Random walks on graphs: A survey, in: Combinatorics, Paul Erdõs is eighty, vol. 2 (Keszthely, 1993), vol. 2 of Bolyai Soc. Math. Stud. János Bolyai Math. Soc., Budapest, 1996, pp. 353–397.

[23] P. Matthews, Covering problems for Brownian motion on spheres, Ann. Prob. 16 (1) (1988) 189–199.

[24] M. Mitzenmacher, A.M. Richa, R. Sitaraman, The power of two random choices: A survey of techniques and results. in: Handbook of Randomized Computing, vol. I. Kluwer Academic Press, 2001, pp. 255–312.

[25] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.

[26] R. Pemantle, Vertex-reinforced random walk, Prob. Theory Relat. Fields 92 (1) (1992) 117–136.

[27] J.S. Rosenthal, Convergence rates for Markov chains, SIAM Rev. 37 (3) (1995) 387–405.

[28] N. Sadagopan, B. Krishnamachari, A. Helmy, Active query forwarding in sensor networks (acquire), J. Ad Hoc Networks 3 (1) (2005) 91–113.

[29] S.D. Servetto, 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 ACM Press, 2002, pp. 12–21.

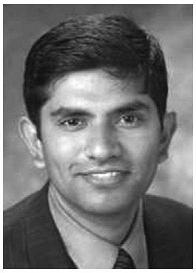
[30] A. Sinclair, Improved bounds for mixing rates of markov chains and multicommodity flow, Comb. Prob. Comput. 1 (1992) 351–370.

[31] S. Volkov, Vertex-reinforced random walk on arbitrary graphs, Ann. Prob. 29 (1) (2001) 66–91.

[32] I.A. Wagner, M. Lindenbaum, A.M. Bruckstein, Robotic exploration, brownian motion and electrical resistance, Lecture Notes Comput. Sci. 1518 (1998) 116–130.

[33] D. Zuckerman, A technique for lower bounding the cover time, in: Proc. of the twenty-second annual ACM symposium on Theory of computing, ACM Press, 1990, pp. 254–259.

Chen Avin received the B.Sc. degree in Communication Systems Engineering from Ben Gurion University, Israel, in 2000. He received the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA) in 2003 and 2006 respectively. He is a Lecturer in the Department of Communication Systems Engineering at the Ben Gurion University since October 2006. His current research interests are: Graphs and Networks Algorithms, Sensor Networks, Random Graphs, Complex Systems and Random Walks.



Bhaskar Krishnamachari received his B.E. in Electrical Engineering at The Cooper Union, New York, in 1998, and his M.S. and Ph.D. degrees from Cornell University in 1999 and 2002 respectively. He is currently the Philip and Cayley MacDonald Early Career Chair Assistant Professor in the Department of Electrical Engineering at the University of Southern California. His primary research interest is in the design and analysis of efficient mechanisms for wireless sensor networks.