How Well Do Random Walks Parallelize?

Klim Efremenko and Omer Reingold*

Department of Computer Science and Applied Mathematics,
The Weizmann Institute of Science, Rehovot, 76100 Israel
klimefrem@gmail.com, omer.reingold@weizmann.ac.il

Abstract. A random walk on a graph is a process that explores the graph in a random way: at each step the walk is at a vertex of the graph, and at each step it moves to a uniformly selected neighbor of this vertex. Random walks are extremely useful in computer science and in other fields. A very natural problem that was recently raised by Alon, Avin, Koucky, Kozma, Lotker, and Tuttle (though it was implicit in several previous papers) is to analyze the behavior of k independent walks in comparison with the behavior of a single walk. In particular, Alon et al. showed that in various settings (e.g., for expander graphs), k random walks cover the graph (i.e., visit all its nodes), $\Omega(k)$-times faster (in expectation) than a single walk. In other words, in such cases k random walks efficiently "parallelize" a single random walk. Alon et al. also demonstrated that, depending on the specific setting, this "speedup" can vary from logarithmic to exponential in k.

In this paper we initiate a more systematic study of multiple random walks. We give lower and upper bounds both on the cover time *and on the hitting time* (the time it takes to hit one specific node) of multiple random walks. Our study revolves over three alternatives for the starting vertices of the random walks: the worst starting vertices (those who maximize the hitting/cover time), the best starting vertices, and starting vertices selected from the stationary distribution. Among our results, we show that the speedup when starting the walks at the worst vertices cannot be too large - the hitting time cannot improve by more than an $O(k)$ factor and the cover time cannot improve by more than $\min\{k \log n, k^2\}$ (where n is the number of vertices). These results should be contrasted with the fact that there was no previously known upper-bound on the speedup and that the speedup can even be *exponential* in k for random starting vertices. Some of these results were independently obtained by Elsässer and Sauerwald (ICALP 2009). We further show that for k that is not too large (as a function of various parameters of the graph), the speedup in cover time is $O(k)$ *even for walks that start from the best vertices* (those that minimize the cover time). As a rather surprising corollary of our theorems, we obtain a new bound which relates the cover time C and the mixing time mix of a graph. Specifically, we show that $C = O(m\sqrt{\mathrm{mix}}\log^2 n)$ (where m is the number of edges).

Keywords: Markov Chains, Random Walks.

* Research supported by grant 1300/05 from the Israel Science Foundation.

I. Dinur et al. (Eds.): APPROX and RANDOM 2009, LNCS 5687, pp. 476–489, 2009.

1 Introduction

A random walk on a graph is a process of exploring the graph in a random way. A simple random walk starts at some node of a graph and at each step moves to a random neighbor. Random walks are fundamental in computer science. They are the basis of MCMC (Markov-Chain Monte-Carlo) algorithms, and have additional important applications such as randomness-efficient sampling (via random walks on expanders) [AKS87], and space-efficient graph connectivity algorithms [AKL+79]. Random walks became a common notion in many fields, such as computational physics, computational biology, economics, electrical engineering, social networks, and machine learning.

Assume that we have some network (e.g. a communication or a social network), and some node u sends a message. Assume that at each step this message is sent to a random neighbor of the last recipient. The message will travel through the network as a random walk on a graph. The expected time until the message will arrive to some other node v is called the hitting time $h(u, v)$. The expected time until the message will visit all the nodes is called the cover time C_u. The hitting time and the cover time of a random walk are thoroughly studied parameters (see surveys [AF99, LWP, Lov96]).

In this paper we consider the following natural question: What happens if we take multiple random walks instead of a single walk? Assume that instead of one copy, k copies of the same message were sent. How long would it take for one of these copies to reach some node v? How long would it take until each node receives at least one of the k copies? What are the speedups in the hitting and cover times of multiple walks compared with a single walk?

Multiple random walks were studied in a series of papers [BKRU89, Fei97, BF93] on time-space tradeoffs for solving undirected s-t connectivity. These papers considered upper bounds for the cover time of multiple random walks, each paper giving a different answer for different distributions of the starting vertices of the random walks. In randomized parallel algorithms, multiple random walks are a very natural way of exploring a graph since they can be easily distributed between different processes. For example, multiple random walks were used in [HZ96, KNP99] for designing efficient parallel algorithms for finding the connected components of an undirected graph.

Multiple random walks were suggested as a topic of independent interest by Alon, Avin, Koucky, Kozma, Lotker, and Tuttle [AAK+07]. Alon et al. [AAK+07] studied lower bounds on the relation between the cover time of a simple random walk and of multiple random walks when the walks start from the same node. The paper proves that if the number of random walks k is small enough (i.e., asymptotically less than $\frac{C}{h_{\max}}$, where C and $h_{\max}$ are the maximal cover time and hitting time respectively) then the relation between the cover time of a single random walk and of multiple random walks is at least $k - o(k)$. In such a case, we can argue that multiple random walks "parallelize" a single walk efficiently (as they don't increase the total amount of work by much). [AAK+07] also showed that there are graphs with logarithmic speedup (e.g., the cycle), and there are graphs with an exponential speedup for specific starting point (e.g., the so called barbell graph; we will shortly discuss a related example). [AAK+07] leaves open the question of upper bounds for the speedup.

The goal of this paper is to systematically study multiple random walks. In addition to the cover time of multiple random walks we will also discuss the hitting time, proving both lower and upper bounds on the speedup. We will extend the discussion to the case where not all the walks start from the same node.

Before getting into the details of our results, let us consider an example which illustrates how multiple random walks behave differently according to the choice of their starting vertices. Consider a graph G which is composed of two cliques of size n connected by a single edge (see Figure 1).

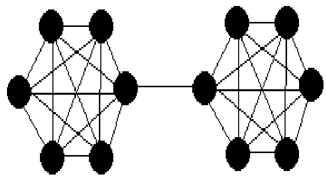

Fig. 1. Two cliques graph - how the speedup changes according to the starting vertices

While the cover time of a single random walk will not depend on the starting vertex and is $\Theta(n^2)$, the cover time of multiple random walks will be very different for different starting vertices of the random walks. When the walks start from the worst vertices (all walks start from the same clique) the cover time is $\Theta(\frac{n^2}{k})$. Even for $k = 2$, when the random walks start from the best vertices (one walk starts at one clique and the other from another clique) the cover time is $\Theta(n \log n)$. When the starting vertices of k random walks are drawn independently from the stationary distribution, then the probability that all starting vertices will fall into the same clique is 2^{-k}. Therefore, for $k \leq \log n - \log \log n$, the cover time in this case is $\Theta(2^{-k} n^2)$. When considering the hitting times, we get the same behavior for the worst starting vertices and for randomly-chosen starting vertices. The case of the best starting vertices is uninteresting when discussing the hitting time as the hitting time in such a case is zero (even for a single walk).

As we can see from the aforementioned example, both the cover and the hitting times heavily depend on the starting vertices. Therefore, we study these three scenarios separately: (1) The case when the random walks start from the nodes which maximize the cover/hitting time (worst starting vertices). (2) The case when the random walks start from the nodes which minimize the cover time (best starting vertices). (3) The case when the starting vertices are drawn independently according to the stationary distribution (random starting vertices).

Our Contribution

In this paper we systematically study multiple random walks and their speedup both in terms of the cover time and in terms of the hitting time. We give various lower and upper bounds for different ways of choosing the starting vertices. Our main bounds on the speedup of multiple random walks are summarized in Table 1.

Table 1. Summary of main bounds on the speedup. Notation: n - number of vertices; k - the number of walks; C - maximal cover time; $h_{\max}$ - maximal hitting time; mix - mixing time.

	Worst case	Average Case	Best Case
Hitting time Upper bounds	$O(k)$ *for any* k, Theorem 4	$k + o(k)$ *for* $k \log n = o(\frac{h_{\max}}{\text{mix}})$ Theorem 20	Not applicable
Hitting time Lower bounds	$\Omega(k)$ *for* $k \log n = O(\frac{h_{\max}}{\text{mix}})$ Theorem 8	k *for any* k Theorems 6	Not applicable
Cover time Upper bounds	$O(\min\{k^2, k \log n\})$ Theorems 12 & 13	$k + o(k)$ *for* $k \log k = o(\frac{C}{\text{mix}})$ Theorem 19	$k + o(k)$ *for* $k = o(\frac{C}{h_{\max}})$ Theorem 15
Cover time Lower Bounds	$(\frac{k}{\log n})(1 - o(1))$ for $k \log n = o(\frac{h_{\max}}{\text{mix}})$ Theorem 14	$\Longrightarrow$	$\Longrightarrow$
	$k - o(k)$ for $k = o(\frac{C}{h_{\max}})$ Theorem 5 in [AAK+07]		

Upper bounds on the speedup. [AAK+07] left open the question of upper bounding the speedup of multiple random walks. In this work we show that the answer depends on how the starting vertices are selected. In Theorem 4 we show that for the worst starting vertices, the speedup on *hitting time* is at most $O(k)$. In Section 4, we use this theorem to show that the speedup on the *cover time* is at most $O(\min(k^2, k \log n))$. As we can see from the example above, the speedup for the best or even for random starting vertices may be very large (e.g., exponential in k). Still, we are able to show in Section 4 that even in these cases, if the number of walks is small enough then the speedup will be at most $k + o(k)$. In Theorem 15 (see also Corollary 17) we show that for $k \ll \frac{C}{h_{\max}}$ the speedup for the best starting vertices is at most $k + o(k)$. This result is interesting for graphs with a large gap between the cover time and the hitting time. For random starting vertices, Theorem 19 (see also Corollary 21) shows that if $k \log k \ll \frac{C}{\text{mix}}$, then the speedup is at most $k + o(k)$. The mixing time, mix, of a graph is the number of steps a random walk has to make until its position is distributed almost according to the stationary distribution.

Lower bounds on the speedup. In Theorem 6 we show that the speedup for the hitting times is at least k when all the starting vertices are drawn from the stationary distribution. This theorem also allows us to prove lower bounds for the case of worst starting vertices for graphs with small mixing time. Using this theorem we prove in Theorem 8 that when the number of walks is less than $\tilde{O}(\frac{h_{\max}}{\text{mix}})$ the speedup for the hitting times is at least $\Omega(k)$. We get similar results for the cover time (Theorem 14). Namely, we show that the speedup for the cover time is at least $(\frac{k}{\log n})(1 + o(1))$, when k is less than $\tilde{o}(\frac{h_{\max}}{\text{mix}})$. This result improves the lower bound of $\Omega(\frac{k}{\log n \cdot \text{mix}})$ from [AAK+07].

A new relation between the cover time and the mixing time. Finally, our study of multiple random walks gives a rather surprising implication on the study of a single random walk. Our results, together with the results of [BKRU89] about multiple random walks, imply a new relation between the cover time and the mixing time of a graph. Specifically, we prove that $C = O(m\sqrt{\text{mix}}\log^2 n)$. The best previous result we are aware of is due to Broder and Karlin [BK88]. In [BK88] it was proven that $C = O(\frac{m \log n}{1-\lambda(G)})$, where $\lambda(G)$ is the second eigenvalue of the normalized adjacency matrix. A known relation between $\lambda(G)$ and mix is that $\Omega(\frac{1}{1-\lambda(G)}) \leq \text{mix} \leq O(\frac{\log n}{1-\lambda(G)})$ (cf. [Sin92], Proposition 1). Therefore a corollary of [BK88] is that $C = O(\text{mix} m \log n)$. Our result improves this bound whenever $\text{mix} = \omega(\log^2 n)$.

Our new relation also has an application in electrical engineering. View a graph G as an electrical network with unit resistors as edges. Let R_{st} be the effective resistance between nodes s and t. Then it was shown in [CRRS89] that for any two nodes s and t it holds that $mR_{st} \leq C$. Therefore, together with our result it implies that $R_{st} = O(\sqrt{\text{mix}}\log^2 n)$. The best previous upper bound on the electrical resistance in terms of the mixing time was also obtained by Chandra et al. [CRRS89] and was $R_{st} \leq \frac{2}{1-\lambda(G)} = O(\text{mix})$.

Related Work. Independently of our work, Elsässer and Sauerwald [ES09] recently studied multiple random walks. Their most related results are upper bounds and lower bounds on the speed-up of cover time for worst case starting points. In fact, [ES09] gives an upper bound of $O(k \log n)$ on the speed-up of any graph (similarly to our Theorem 12) and a lower bound of $\Omega(\frac{k}{\log n})$ under some conditions on mixing time (similarly to our Theorem 14). Under some mild conditions, they are also able to prove an upper bound of $O(k)$. Another recent work on multiple random walks is due to [CCR09]. This work studies multiple random walks in random graphs, and among other result show that for random d-regular graph the speed-up is $O(k)$.

2 Notation

We will use standart definitions of the hitting time, the cover time and the mixing time. We briefly review the notation that will be used throughout the paper: The mixing time of a graph G is denoted mix. Let $\varsigma(u, v)$ be the time it takes for a random walk that starts at u to reach v i.e. $\varsigma(u, v) = \min\{t \mid X_u(t) = v\}$. Note that $\varsigma(u, v)$ is a random variable. Let the hitting time $h(u, v) = \mathbf{E}(\varsigma(u, v))$ be the expected time for the random walk to traverse from u to v. Let $h_{\max} = \max_{u,v \in V} h(u, v)$ and $h_{\min} = \min_{u,v \in V} h(u, v)$ be the maximal and minimal hitting times. Similarly let τ_u be the time for the simple random walk to visit all the nodes of the graph. Let $C_u = \mathbf{E}(\tau_u)$ be the cover time for a simple walk starting at u. The cover time $C = \max_u(C_u)$ is the maximal (over the starting vertex u) expected time it takes for a single walk to cover the graph. It will be convenient for us to define the following parameter of a graph: $H(G) = \frac{C}{h_{\max}}$.

The following theorem provides fundamental bounds on the cover time in terms of the hitting time (for more details see [LWP] Chapter 11 or [Mat88]):

Theorem 1 (cf. [Mat88]). *For every graph G with n vertices*

$$h_{min} \cdot \log n \leq C \leq h_{max} \cdot \log n.$$

Note that there also exists a trivial bound of $h_{max} \leq C$. It will be convenient for us to define the following parameter of a graph: $H(G) = \frac{C}{h_{max}}$. Note that $1 \leq H(G) \leq \log n$. Also note that there exist graphs where $H(G) = O(1)$ (for example the cycle), and there exist graphs with $H(G) = \Omega(\log n)$ (for example the complete graph).

For k parallel independent random walks we have the following notation: $\varsigma(\{u_1, u_2, \dots u_k\}, v) = \min_{i=1}^{k} \varsigma(u_i, v)$ is the random variable corresponding to the hitting time of k random walks, where some of the u_i's may be equal. Let $h(\{u_1, u_2, \dots u_k\}, v) = \mathbf{E}(\varsigma(\{u_1, u_2, \dots u_k\}, v))$ be the hitting time of k random walks starting at vertices u_i. If all the walks start at the same vertex u we will write it as $h^k(u, v)$. Let $h^k_{\max} = \max_{u_i, v} h(\{u_1, u_2, \dots u_k\}, v)$ be the maximal hitting time of k random walks. Similarly, for the cover time we define $\tau_{u_1, u_2, \dots u_k} = \min\{t \mid \bigcup_{i=1}^{k} \{X_{u_i}(1), X_{u_i}(2), \dots X_{u_i}(t)\} = V\}$ and define $C_{u_1, u_2, \dots u_k} = \mathbf{E}\tau_{u_1, u_2, \dots u_k}$ to be the expected cover time. Let $C^k = \max_{u_1, u_2, \dots u_k} C_{u_1, u_2, \dots u_k}$.

The proof of Theorem 1 (see [LWP] Chapter 11) easily extends to multiple walks implying the following theorem:

Theorem 2. *For every (strongly connected) graph G with n vertices, and for every k*

$$\frac{C_k}{h^k_{\max}} \leq \log n.$$

3 Hitting Time of Multiple Random Walks

In this section we study the behavior of the hitting time of k random walks. The first question we will consider is: what are the starting vertices of multiple random walks which maximize the hitting time? Later, we will give a lower bound on the maximal hitting time of multiple random walks. We will prove that $\frac{h_{\max}}{h^k_{\max}} = O(k)$. Then we will consider the case where the walks' starting vertices are chosen independently according to the stationary distribution. Note that in this setting the ratio between hitting times is not upper bounded by $O(k)$; in fact it may even be exponential in k. We will prove that in this setting the ratio between the hitting time of the single walk and the hitting time of k walks is at least k. Next we will use this theorem in order to prove that for graphs with small mixing time the ratio $\frac{h_{\max}}{h^k_{\max}} = \Omega(k)$. Finally we consider the evaluation of hitting times.

3.1 Worst to Start in a Single Vertex

Let us prove that the maximal hitting time is achieved when all the walks start from the same node.

Theorem 3. *For every graph $G = (V, E)$, for every $v \in V$ it holds that*

$$\max_{u_1, u_2, \dots u_k} h(\{u_1, u_2, \dots u_k\}, v) = \max_{u} h^k(u, v).$$

The proof of the theorem (which employs a generalization of Hölder's Inequality) is deferred to the full version.

3.2 Upper Bound on the Speedup of the Hitting Time of Multiple Random Walks

We will now prove that the ratio between the hitting time of a single random walk and the hitting time of k random walks is at most $O(k)$.

Theorem 4. *For any graph G it holds that $h_{\max} \leq 4kh^k_{\max}$.*

Loosely, the theorem is proved by deducing a bound of $\frac{1}{2k}$ on the probability that a single walk will hit the target vertex in $2h^k_{\max}$ steps. The formal proof is deferred to the full version. By a slightly more complicated argument we can replace the constant 4 in Theorem 4 by $\mathbf{e} + o(1)$. However it seems plausible that the right constant is 1.

Open Problem 5. *Prove or disprove that for any graph G it holds that $h_{\max} \leq kh^k_{\max}$.*

3.3 Lower Bounds on the Speedup of the Hitting Time of Multiple Random Walks

In this section, we consider the case where the starting vertices of the random walks are selected according to the stationary distribution. Theorem 4 shows that for worst-case starting vertices the ratio between the hitting times of a single walk and multiple walks is at most $O(k)$. But as we will soon show, when the starting vertices of all walks are drawn independently from the stationary distribution then, loosely speaking, this ratio becomes at least k. Note that in some graphs the ratio of hitting times, when the starting vertices are selected according to the stationary distribution, may even become exponential in k. Indeed, such an example is given in Figure 1 and is discussed in the introduction (the discussion there is for the cover time but the analysis for the hitting time is very similar)

The next theorem gives a lower bound on the ratio between hitting times for random starting vertices.

Theorem 6. *Let $G(V, E)$ be a (connected) undirected graph. Let X be a random walk on G. Let $u, u_1, \ldots u_k \in V$ be independently chosen according to the stationary distribution of G. Then:*

$$\boldsymbol{E}_u(h(u, v)) \geq k\left(\boldsymbol{E}_{u_i} h(\{u_1, u_2, \ldots u_k\}, v) - 1\right).$$

Remark 7. *In this theorem we assume continues model of random walk.*

As we will later see (in Corollary 22), when $k \log k = o(h(u, v)/\text{mix})$ then the speedup is at most $k + o(k)$ in the scenario of random starting vertices. Thus when $k \log k = o(h(u, v)/\text{mix})$ the speedup is k up to lower order terms.

The proof of the theorem is deferred to the full version.

Lower bound on the speedup for worst starting vertices. The lower bound on the speedup for walks starting at the stationary distribution translates into a lower bound that also applies to walks starting at the worst vertices: First let the walks converge to the stationary distribution and then apply the previous lower bound. The bounds that we obtain are especially meaningful when the mixing time of the graph is sufficiently smaller than the hitting time.

Theorem 8. *Let $G(V, E)$ be a (connected) undirected graph. Then*

$$h^k_{\max} \leq \frac{h_{\max}}{k} + O(\mathrm{mix}(\log n + \log k)).$$

As a corollary we get:

Corollary 9. *Let $G(V, E)$ be a (connected) undirected graph such that $k\mathrm{mix}(\log n + \log k) = o(h_{\max})$. Then:*

$$\frac{h_{\max}}{h^k_{\max}} \geq k(1 - o(1)).$$

3.4 Calculating the Hitting Time of Multiple Random Walks

We would like to address a question which is somewhat orthogonal to the main part of this paper. Namely, we would like to discuss how the hitting time of multiple walks can be calculated. Let us observe that multiple random walks on graph G can be presented as a single random walk on another graph G^k.

Definition 10. *Let $G = (V, E)$ be some graph. Then the graph $G^k = (V', E')$ is defined as follows: The vertices of G^k are k-tuples of vertices of G i.e.*

$$V' = \underbrace{V \oplus V \ldots \oplus V}_{k \text{ times}} = V^k.$$

For every k edges of G, (u_i, v_i) for $i = 1, \ldots, k$ we have an edge between $u' = (u_1, u_2, \ldots u_k)$ and $v' = (v_1, v_2 \ldots v_k)$ in G^k.

One can view k random walks on G as a single random walk on G^k where the first coordinate of G^k corresponds to the first random walk, the second coordinate corresponds to the second random walk, and so on.

Let $A \subset V^k$ be the set of all nodes of G^k which contain the node $v \in V$. Assume that we have k random walks beginning at $u_1, u_2, \ldots u_k$. Then the time it will take to hit v is equal to the time for a single random walk on G^k beginning at node $(u_1, u_2, \ldots u_k)$ to hit the set A. Thus instead of analyzing multiple random walks we can study a single random walk on G^k. There is a polynomial time algorithm for calculating hitting times of a single random walk (cf. [Lov96]). This gives us an algorithm, which is polynomial in n^k, for calculating $h(\{u_1, u_2, \ldots u_k\}, v)$. A natural question is whether there exist more efficient algorithms.

Open Problem 11. *Find a more efficient algorithm for calculating $h(\{u_1, u_2, \ldots u_k\}, v)$.*

4 Cover Time of Multiple Random Walks

Let us turn our attention from the hitting time to the cover time. As in the case of the hitting time, the cover time heavily depends on the starting vertices of the random walks. The graph given by Figure 1 and discussed in the introduction gives an example where the speedup in cover time of k random walks is linear in k for worst-case starting vertices, it is exponential in k for random starting vertices, and even for $k = 2$ it is $\Omega(n/\log n)$ for the best starting vertices.

Theorem 1 gives a relation between hitting times and cover times. Thus, our results on hitting times from the previous section also give us results on the cover times. In Subsection 4.1 we will give these results and will analyze the speedup, $\frac{C}{C_k}$, for worst starting vertices. We show that it is bounded by $\min\{k^2, k\log n\}$ for any k. We will also show that for k such that $k \log n \mathrm{mix} = O(h_{\max})$ the speedup is $\Omega(\frac{k}{\log n})$.

We will show in Subsection 4.2 that when k random walks begin from the best starting vertices for $k = o(H(G))$ the speedup is roughly k and is therefore essentially equal to the speedup for the worst case. In Subsection 4.3 we will show that when the starting vertices are drawn from the stationary distribution for k such that $\mathrm{mix} k \log(k) = o(C)$, the speedup is at most k.

4.1 The Worst Starting Vertices

As a simple corollary of Theorem 4 we obtain the following relation:

Theorem 12. *The speedup* $\frac{C}{C^k}$ *is at most* $4kH(G) \leq 4k \log n$

Proof. Recall that $C^k \geq h^k_{\max}$ so $\frac{C}{C^k} \leq \frac{C}{h^k_{\max}} = \frac{h_{\max}}{h^k_{\max}} H(G)$. From Theorem 4 it follows that $\frac{h_{\max}}{h^k_{\max}} H(G) \leq 4kH(G)$. And finally from Theorem 1 we have that $4kH(G) \leq 4k \log n$.

From this theorem it follows that for $k = \Omega(H(G))$ the speedup is $O(k^2)$. Theorem 15 implies that if $k < 0.01H(G)$ then the speedup $\frac{C}{C^k}$ is at most $2k$. Therefore, we can conclude a bound for every k:

Theorem 13. *For every (strongly connected) graph G and every k, it holds that* $\frac{C}{C^k} = O(k^2)$.

From Theorem 8 we can also deduce a lower bound on the speedup for rapidly-mixing graphs:

Theorem 14. *Let* $G(V, E)$ *be an undirected graph and let* k *be such that* $k(\log n)\mathrm{mix} = o(h_{\max})$ *then*

$$\frac{C}{C_k} \geq \frac{k}{\log n}(1 - o(1)).$$

Proof. From Theorem 2 it follows that $\frac{C}{C_k} \geq \frac{h_{\max}}{h^k_{\max} \log n}$. Since $k \log n \mathrm{mix} = o(h_{\max})$, Theorem 8 implies that $\frac{h_{\max}}{h^k_{\max}} = k(1 - o(1))$. Thus: $\frac{C}{C_k} \geq \frac{k}{\log n}(1 + o(1))$.

4.2 The Best Starting Vertices

As we discussed earlier, multiple random walks can be dramatically more efficient than a single random walk if their starting vertices are the best nodes (rather than the worst nodes). In fact, we have seen an example where taking two walks instead of one reduces the cover time by a factor of $\Omega(n/\log n)$. In this section we show that in graphs where the cover time is significantly larger than the hitting time, a few random walks cannot give such a dramatic speedup in the cover time, *even when starting at the best nodes*: If $k = o(H(G))$ (recall that $H(G) = \frac{C}{h_{max}}$), then the speedup $\frac{C}{C_{u_1,u_2,\ldots u_k}}$ (where $u_1, u_2, \ldots u_k$ are best possible) is not much bigger than k. Note that in the case where $k = o(H(G))$ it has been shown in [AAK+07] that the speedup $\frac{C}{C_{u_1,u_2,\ldots u_k}}$ is at least $k - o(k)$, even if $u_1, u_2, \ldots u_k$'s are worst possible. Combining the two results we get that the speedup is roughly k regardless of where the k walks start.

We want to show that the cover time of a single random walk is not much larger than k times the cover time of k random walks. For that we will let the single walk simulate k random walks (starting from vertices u_i) as follows: The single walk runs until it hits u_1, then it simulates the first random walk. Then it runs until it hits u_2 and simulates the second random walk and so on until hitting u_k and simulating the k'th random walk. The expected time to hit any vertex from any other vertex is bounded by $h_{\max}$. Thus intuitively the above argument should imply the following bound: $C \leq kC_{u_1,u_2,\ldots u_k} + kh_{\max}$. Unfortunately, we do not know how to formally prove such a strong bound. The difficulty is that the above argument only shows how a single walk can simulate k walks for t steps, where t is fixed ahead of time. However, what we really need is for the single walk to simulate k walks until the walks cover the graph. In other words, t is not fixed ahead of time but rather a random variable which depends on the k walks. Nevertheless, we are still able to prove the following bound which is weaker by at most a constant factor:

Theorem 15. *For every graph G and for* ***any*** *k nodes $u_1, u_2, \ldots u_k$ in G, it holds that:*

$$C \leq kC_{u_1,u_2,\ldots u_k} + O(kh_{\max}) + O\left(\sqrt{kC_{u_1,u_2,\ldots u_k} h_{\max}}\right).$$

The proof will appear in the full version.

In [AAK+07] the following theorem was proved:

Theorem 16 (Theorem 5 from [AAK+07]). *Let G be a strongly connected graph and $k = o(H(G))$ then $\frac{C}{C_k} \geq k - o(k)$.*

In the case where $k = o(H(G))$ then $O(kh_{\max}) + O(\sqrt{C_{u_1,u_2,\ldots u_k} kh_{\max}}) = o(C)$ and therefore $C \leq kC_{u_1,u_2,\ldots u_k} + o(C)$. As a corollary we get:

Corollary 17. *Let G be a strongly connected graph and $k = o(H(G))$ then for any starting vertices $u_1, u_2, \ldots u_k$ it holds that: $\frac{C}{C_{u_1,u_2,\ldots u_k}} = k \pm o(k)$*

It seems plausible that the speedup is at most k for *any* starting vertices, also when k is significantly larger than $H(G)$. When $k \geq \mathbf{e}^{H(G)}$ we can give an example where $kC_{u_1,u_2,\ldots u_k} << C$. Consider a graph G which is composed of a clique of size n and t vertices where each vertex is connected by one edge to some node of a clique. We will assume that $n >> t$. The maximal hitting time for this graph is $O(n^2)$. The cover time of this graph is $O(n^2 \log t)$ and $H(G) = \log t$. If $k = t$ then when k multiple random walks start from the t vertices which are not in the clique, then $C_{u_1,u_2,\ldots u_k} = \frac{n \log n}{k} + O(1)$. Therefore, a natural open problem is the following:

Open Problem 18. *Prove or disprove that for some constant $\alpha > 0$, for any graph G, if $k \leq \mathbf{e}^{\alpha H(G)}$ then $C \leq O(k)C_{u_1,u_2,\ldots u_k}$.*

4.3 Random Starting Vertices

Finally we consider the cover time of k walks that start from vertices drawn from the stationary distribution. In this case, Theorem 6 loosely states that the ratio between the *hitting times* is at least k. Now let us show an upper bound on the ratio between the *cover time* of a single random walk and multiple random walks.

The intuition for the bound is quite similar to the intuition behind the proof of Theorem 15 (nevertheless, the proofs are quite a bit different). We will simulate k random walks by a single walk. The single random walk will first run $\ln(k)\mathrm{mix}$ steps, getting to a vertex that is distributed almost according to the stationary distribution. The walk then simulates the first of the k random walks. Next, the walk takes $\ln(k)\mathrm{mix}$ steps again and simulates the second random walk and so on until simulating the kth random walk. Since the start vertex of the k simulated walks are *jointly* distributed almost as if they were independently sampled from the stationary distribution it seems that we should obtain the following upper bound: $C \leq k\mathbf{E}_{u_i} C_{u_1,u_2,\ldots u_k} + k\ln(k)\mathrm{mix}$, where $u_1, u_2, \ldots u_k$ are independently drawn from the stationary distribution. But as before we can not make this intuition formal, mainly because we do not know ahead of time how long the k random walks will take until they cover the graph. We will instead prove the following bound which again may be weaker by at most a constant factor:

Theorem 19. *Let $G = (V, E)$ be any (strongly connected) graph. Let $u_1, u_2, \ldots u_k$ be drawn from the stationary distribution of G. Then:*

$$C \leq k\boldsymbol{E}_{u_i} C_{u_1,u_2,\ldots u_k} + O(k\ln(k)\mathrm{mix}) + O\left(k\sqrt{\boldsymbol{E}C_{u_1,u_2,\ldots u_k}\mathrm{mix}}\right).$$

Under some restrictions, the mixing time cannot be much larger than the maximal hitting time and often will be much smaller. In such cases, Theorem 19 may be more informative than Theorem 15 in the sense that it implies a bound of roughly k on the speedup as long as $k = \tilde{O}(\frac{C}{\mathrm{mix}})$ (rather than $k = O(\frac{C}{h_{\max}})$ as implied by Theorem 15). On the other hand, the starting vertices in Theorem 19 are according to the stationary distribution rather than arbitrary starting vertices as in Theorem 15.

The proof of Theorem 19 will appear in the full version. We note that the proof also works if we consider the hitting times (rather than the cover times), implying the following theorem:

Theorem 20. *Let $G = (V, E)$ be any (strongly connected) graph. Let u, v be any nodes of the graph and let $u_1, u_2, \ldots u_k$ be drawn from the stationary distribution of G. Then:*

$$h(u,v) \leq k\boldsymbol{E}_{u_i} h(\{u_1,u_2,\ldots u_k\},v) + O(k\ln(k)\mathrm{mix}) + O\left(k\sqrt{\boldsymbol{E}_{u_i} h(\{u_1,u_2,\ldots u_k\},v)\mathrm{mix}}\right).$$

As a corollary of Theorems 19 it follows that if $k\log k\mathrm{mix}$ is negligible relative to the cover time then the speedup of the cover time is at most k

Corollary 21. *Let $G = (V, E)$ be any (strongly connected) graph. Let $u_1, u_2, \ldots u_k$ be drawn from the stationary distribution of G. Then if $k\log(k) = o(C/\mathrm{mix})$ then*

$$\frac{C}{\boldsymbol{E}_{u_i} C_{u_1,u_2,\ldots u_k}} \leq k + o(k).$$

Similarly, from Theorem 20 we obtain the following corollary:

Corollary 22. *Let $G = (V, E)$ be any (strongly connected) graph. Let $u_1, u_2, \ldots u_k$ be drawn from the stationary distribution of G and u, v any nodes. Then if $k \log(k) = o(h(u, v)/\mathrm{mix})$ then*

$$\frac{h(u, v)}{\boldsymbol{E}_{u_i} h(\{u_1, u_2, \ldots u_k\}, v)} \leq k + o(k).$$

5 A New Relation between Cover and Mixing Time

In this section we will show how we can use the results proven above in order to prove a new upper bound on the cover time in terms of mixing time. In order to do this we will need the following bound from [BKRU89].

Theorem 23 (cf. [BKRU89] Theorem 1). *Let G be a connected undirected graph with n vertices and m edges. Let $u_1, u_2, \ldots u_k$ be drawn from the stationary distribution of G. Then:*

$$\boldsymbol{E}_{u_i}(C_{u_1, u_2, \ldots u_k}) \leq O(\frac{m^2 \log^3 n}{k^2}).$$

As a rather intriguing corollary of Theorem 23 and Theorem 19 we get the following bound on the cover time.

Theorem 24. *Let G be a connected undirected graph with n vertices and m edges. Then:*

$$C \leq O(m\sqrt{\mathrm{mix}} \log^2 n).$$

Proof. From Theorem 19 it follows that:

$$C(G) \leq k\mathbf{E}_{u_i} C_{u_1, u_2, \ldots u_k}(G) + O(k \ln(k)\mathrm{mix}) + O\left(k\sqrt{\mathbf{E}C_{u_1, u_2, \ldots u_k}\mathrm{mix}}\right).$$

Thus from Theorem 23 we get the following bound on $C(G)$:

$$C(G) \leq O(\frac{m^2 \log^3 n}{k}) + O(k \ln(k)\mathrm{mix}) + O(m \log^{1.5} n\sqrt{\mathrm{mix}}).$$

As long as k is at most polynomial in n it follows that $\log k = O(\log n)$. Thus:

$$C(G) \leq O(\frac{m^2 \log^3 n}{k}) + O(k \ln(n)\mathrm{mix}) + o(m \log^2 n\sqrt{\mathrm{mix}}).$$

Setting $k = \frac{m \log n}{\sqrt{\mathrm{mix}}}$ implies the theorem.

6 Future Research

This paper systematically studies the behavior of multiple random walks. While we have given various upper and lower bounds for the speedup of multiple random walks, there is still much more that we do not know on this topic, with a few examples being Open Problems 5, 11 and 18. In this section, we will discuss a few additional directions for further research.

Our knowledge on the hitting time of multiple random walks is more complete than our knowledge on their cover time. Indeed, analyzing the hitting time seems easier than analyzing the cover time. Designing new tools for analyzing the cover time of multiple random walks is an important challenge. For example, we have proved that the maximal hitting time of multiple random walks is obtained when all the walks start from the same vertex (see Theorem 4), but we don't know if the same is also true for the cover times:

Open Problem 25. *Prove or disprove that for any graph G*

$$\max_{u_1,u_2,\ldots u_k} C^k_{u_1,u_2,\ldots u_k} = \max_u C^k_{u,u,\ldots u}.$$

We have proved that in the case of worst starting vertices the speedup of the hitting time is at most $4k$, and we raised the question of whether the correct constant is one (see Open Problem 5). It seems however, that for the cover time the speedup may be larger than k (though it is still possible that it is $O(k)$). Consider a walk on a "weighted" path $a-b-c$ with self loops such that the probability of staying in place is $1-\frac{1}{x}$. In other words, consider a Markov chain $X(t)$ with the following transition probabilities:

$$\begin{aligned} \Pr[X(t)=b|X(t-1)=a] &= \Pr[X(t)=b|X(t-1)=c] = \tfrac{1}{x} \\ \Pr[X(t)=c|X(t-1)=b] &= \Pr[X(t)=a|X(t-1)=b] = \tfrac{1}{2x} \end{aligned}$$

Calculating the cover times gives the following: The worst starting vertex of a single random walk is b and the cover time is $5x+o(x)$. The worst starting vertices of 2 random walks is when both walks start at a and the cover time in such a case is $2.25x+o(x)$. Thus, in this case the speedup for 2 walks is 2.222. It is an interesting question to find stronger examples (where the speedup is larger than k), and of course it would be interesting to find a matching upper bound on the speedup.

A technical issue that comes up in our analysis is that in order to understand the behavior of multiple random walks it may be helpful to understand the behavior of short random walks. For example, what kind of bound can be obtained on $\Pr[\varsigma(u,v) \geq h_{\max}/2]$ (for an undirected and connected graph).

Finally, it will be interesting to explore additional applications of multiple random walks, either in computer science or in other fields.

Acknowledgements

We thank Chen Avin, Uriel Feige, Gady Kozma, Zvi Lotker and Ofer Zeitouni for useful discussions. In particular, we are grateful to Gady for pointing out a flaw in one of our earlier arguments. We thank Rivka Efremenko for reviewing drafts of this paper.

References

[AAK+07] Alon, N., Avin, C., Koucky, M., Kozma, G., Lotker, Z., Tuttle, M.R.: Many Random Walks Are Faster Than One. ArXiv e-prints 705 (May 2007), http://arxiv.org/abs/0705.0467

[AF99] Aldous, D., Fill, J.: Reversible Markov Chains and Random Walks on Graphs (1999), http://www.stat.berkeley.edu/aldous/RWG/book.html

[AKL+79] Aleliunas, R., Karp, R.M., Lipton, R.J., Lovász, L., Rackoff, C.: Random walks, universal traversal sequences, and the complexity of maze problems. In: 20th Annual Symposium on Foundations of Computer Science (FOCS), San Juan, Puerto Rico, October 29-31, 1979, pp. 218–223. IEEE Computer Society Press, Los Alamitos (1979)

[AKS87] Ajtai, M., Komlós, J., Szemerédi, E.: Deterministic simulation in LOGSPACE. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC), New York City, May 25-27, 1987, pp. 132–140 (1987)

[BF93] Barnes, G., Feige, U.: Short random walks on graphs. In: STOC, pp. 728–737 (1993)

[BK88] Broder, A.Z., Karlin, A.R.: Bounds on the cover time. Symposium on Foundations of Computer Science, 479–487 (1988)

[BKRU89] Broder, A.Z., Karlin, A.R., Raghavan, P., Upfal, E.: Trading space for time in undirected s-t connectivity. In: STOC, pp. 543–549 (1989)

[CCR09] Frieze, A., Cooper, C., Radzik, T.: Multiple random walks in random regular graphs. In: ICALP (2009)

[CRRS89] Chandra, A.K., Raghavan, P., Ruzzo, W.L., Smolensky, R.: The electrical resistance of a graph captures its commute and cover times. In: STOC 1989: Proceedings of the twenty-first annual ACM symposium on Theory of computing, pp. 574–586. ACM, New York (1989)

[ES09] Elsässer, R., Sauerwald, T.: Tight bounds for the cover time of multiple random walks. In: ICALP (2009)

[Fei97] Feige, U.: A spectrum of time-space trade-offs for undirected s-t connectivity. J. Comput. Syst. Sci. 54(2), 305–316 (1997)

[HZ96] Halperin, S., Zwick, U.: An optimal randomised logarithmic time connectivity algorithm for the erew pram. J. Comput. Syst. Sci. 53(3), 395–416 (1996)

[KNP99] Karger, D.R., Nisan, N., Parnas, M.: Fast connected components algorithms for the EREW PRAM. SIAM J. Comput. 28(3), 1021–1034 (1999)

[Lov96] Lovász, L.: Random walks on graphs: A survey. Combinatorics, Paul Erdos is Eighty 2, 353–398 (1996)

[LWP] Levin, D.A., Wilmer, E., Peres, Y.: Markov Chains and Mixing Times. Oxford University Press, Oxford

[Mat88] Matthews, P.: Covering problems for markov chains. The Annals of Probability 16(3), 1215–1228 (1988)

[Sin92] Sinclair, A.: Improved bounds for mixing rates of markov chains and multicommodity flow. Combinatorics, Probability and Computing 1, 351–370 (1992)