Deterministic approximation of the cover time [†]

Uriel Feige [‡] Yuri Rabinovich [§]

Abstract

The cover time is the expected time it takes a random walk to cover all vertices of a graph. Despite the fact that it can be approximated with arbitrary precision by a simple polynomial time Monte-Carlo algorithm which simulates the random walk, it is not known whether the cover time of a graph can be computed in deterministic polynomial time. In the present paper we establish a deterministic polynomial time algorithm that for any graph and any starting vertex approximates the cover time within polylogarithmic factors. More generally, our algorithm approximates the cover time for arbitrary reversible Markov chains. The new aspect of our algorithm is that the starting vertex of the random walk may be arbitrary and is given as part of the input, whereas previous deterministic approximation algorithms for the cover time assume that the walk starts at the worst possible vertex. In passing, we show that the starting vertex can make a difference of up to a multiplicative factor of $\Theta(n^{3/2}/\sqrt{\log n})$ *in the cover time of an* n*-vertex graph.*

1 Introduction

An n state *Markov chain* is a discrete-time process defined by an $n \times n$ stochastic matrix $P = \{p_{ij}\}$, called the *transition matrix.* (In a stochastic matrix, all the entries are nonnegative, and for every row $1 \le i \le n$, $\sum_{j=1}^{n} p_{ij} = 1$.) Entry p_{ij} specifies the probability that the Markov chain at state i moves at the next step to state j. For our purposes, it is convenient to represent the Markov chain by an edge-weighted directed graph, whose vertices represent the states of the chain, and the weights of the edges $\{(v_i, v_j)\}$ are given by the transition probabilities $\{p_{ij}\}$.

A *reversible* (equivalently, *time-reversible*) Markov chain can be viewed as a random walk on a connected undirected graph with nonnegative weights on the edges. Its transition matrix is obtained by considering a nonnegative symmetric matrix W specifying the edge weights w_{ij}, and scaling each of its rows to sum up to 1. This gives $p_{ij} = w_{ij}/\sum_k w_{ik}$.

We use the term *random walk* do denote reversible Markov chains in which the weights of all edges in the underlying graph are equal (to 1), and there are no parallel edges and self loops. The associated transition matrix P is derived from the adjacency matrix A of the graph by scaling each row of A so that the entries in the row sum up to 1. For such a random walk, the next vertex to be visited is chosen uniformly at random from the set of neighbors of the current vertex. The emphasis in the current paper

[†]A preliminary version of this paper has appeared in the Proceedings of the Fourth Israel Symposium on Theory of Computing and Systems, 1996, Jerusalem, pp. 208–218, IEEE Computer Society Press, Los Alamitos, California.

[‡]Department of Computer Science and Applied Mathematics, the Weizmann Institute, Rehovot, Israel. Incumbent of the Joseph and Celia Reskin Career Development Chair. e-mail: `feige@wisdom.weizmann.ac.il`.

[§]Department of Computer Science, University of Haifa, Haifa, Israel. e-mail: `yuri@cs.haifa.ac.il`.

is on random walks, though most of the results extend to general reversible Markov chains, and some of the results extend to general Markov chains.

For a Markov chain M with a starting point s, let $E_s[M]$ denote the *cover time* of the chain, i.e., the expected number of steps it takes for a chain that starts at s to visit all the vertices. When M is a random walk on the graph G, we shall write simply $E_s[G]$. The cover time is a natural parameter of random walks, and it arises in a number of applications (see [2] for example).

Other important quantities related to random walks are the *hitting time* $H[u,v]$ from vertex u to vertex v, defined as the expected number of steps it takes the chain to move from u to v, the *commute time* $C[u,v] = H[u,v] + H[v,u]$, and the *difference time* $D[u,v] = H[u,v] - H[v,u]$. (Observe that in general $H[u,v] \neq H[v,u]$. For example, if u is the endpoint of an n-vertex path and v is its neighbor, then $H[u,v] = 1$ whereas $H[v,u] = 2n-3$.) The hitting times (and consequently the commute and the difference times) can be computed in polynomial time as follows. The hitting times $H[u_i,v]$, $i = 1,\ldots,n$, satisfy the following relations. $H[v,v] = 0$, and for $u_i \neq v$,

$$H[u_i,v] = 1 + \sum_j p_{ij}\, H[u_j,v]\,.$$

Treating the hitting times $H[u_i,v]$ as variables $\{x_{u_j,v}\}$, this gives a system of n equations in n variables, which can be solved in polynomial time. The unique solution of this system is the hitting times $H[u_i,v]$. (Note: if the graph is not connected then the above procedure should be applied only to the connected component containing v.)

The cover times are apparently much harder to compute, and in particular, despite a considerable amount of work on the subject, it is not known whether they can be computed, or even approximately computed, in deterministic polynomial time. However, they are computable in exponential time: arguing as before, it suffices to solve a system of linear equations in exponentially many variables, where the variable $x_{u_i,S}$ specifies the expected time it takes to cover the set of vertices $V(G) \setminus S$, starting at vertex u_i. The boundary conditions are $x_{u_i,V(G)} = x_{u_i,V(G)\setminus\{u_i\}} = 0$, while for $S \neq \{u_i\}, V(G)$,

$$x_{u_i,S} = 1 + \sum_j p_{ij} x_{u_j,S\cup\{u_i\}}\,.$$

The following simple (expected) polynomial time Monte Carlo algorithm approximates the cover time of a random walk with arbitrary precision. Using a source of randomness, simulate the Markov chain until it covers the entire graph, counting the steps to find out how long it takes. Repeat this experiment (with fresh randomness) several times (where the number of repetitions will be discussed shortly), and output the average of the observed cover times. The expected running time of each repetition is proportional to the actual cover time, and the accuracy of the approximation depends on the variance of the cover time. Since for random walks both the cover time and its variance are bounded by a polynomial in $n = |V(G)|$ (see, e.g. [2]), it suffices for the number of repetitions to be polynomial in (n/ϵ) to guarantee a probability of $1-\epsilon$ that the output of the algorithm is within ϵ of the true cover time. One negative feature of this algorithm (from a purely theoretical point of view) is that it does not reveal much about the relations between the global structure of the graph and its cover time.

The running time of the above randomized algorithm might not be polynomial for general reversible Markov chains, where the cover time can be exponential in the total length of description of the edge-weights. (For example, consider an undirected graph on three vertices, where edge $(1,2)$ has weight 1 and edge $(2,3)$ has integer weight t. Using binary encoding, the description length of such a graph is $O(\log t)$, whereas the cover time is exponentially larger: $\Theta(t)$.) For non-reversible Markov chains (not treated in this paper) the cover time can be exponential in the number of states even when all edge-weights are equal. (Consider for example an n state Markov chain where from every state $0 \leq i \leq n-1$ one moves to state $i+1$ (modulo n) with probability $1/2$ and to state 0 with probability $1/2$. The cover time is exponential in n.)

In view of the above mentioned randomized algorithm, it is natural to ask whether there exists a deterministic polynomial time algorithm that approximates the cover time of Markov chains in general, and of random walks in particular. The question was addressed previously mostly in the form of estimating the cover time for Markov chains (or random walks) that start at a state (or vertex) from which the cover time is largest. (See next section for the bibliography and the discussion why the choice of the starting vertex makes a difference.) In this paper we treat the starting vertex as part of the input, and provide polylogarithmic deterministic approximation of the cover time from an arbitrary starting vertex. We also indicate how our results extend to time-reversible Markov chains.

1.1 Related work

Related previous works usually provide upper or lower bounds on the cover time that hold for any size-n graph G belonging to some subclass of graphs, and any starting vertex. That is, the bounds typically depend only on the number of vertices of the graph, or on some property of the subclass from which G has been driven. These general bounds do not directly lead to any reasonable approximation of the cover time of an input graph with a specified starting vertex.

We survey some of the known results.

Aleliunas *et al.* [2] were the first to give an n^3 upper bound on the cover time for random walks. Their approach is sketched below. From graph G one derives a complete digraph G', where every pair of vertices (u,v) is connected by a (directed) edge of weight $H[u,v]$. Observe that the expected time to cover vertices v and w when starting at u can neither be larger than $H[u,v]+H[v,w]$, nor larger than $H[u,w]+H[w,v]$. Let $HPATH_s(G')$ be the weight of the minimum-weight Hamiltonian path in G' starting at s (i.e., path starting at s and traversing every vertex in G' exactly once). Then $E_s[G] \leq HPATH_s(G')$. In order to bound $HPATH_s(G')$ from above, one considers a Hamiltonian path in G' of a particular structure. Consider an arbitrary spanning tree T in G rooted at s. This induces a Hamiltonian path in G' by traversing the tree in depth first search order. Because of the triangle inequality for hitting times, the weight of this Hamiltonian path cannot be more than the sum of hitting times along all edges of the tree, where for each edge $(u,v) \in T$ we take both $H[u,v]$ (going "up" the tree) and $H[v,u]$ ("backtracking" in the depth first search). Recall that $H[u,v]+H[v,u]$ is the commute time $C[u,v]$. Aleliunas *et al.* [2] show that the commute time $C[u,v]$ between any two adjacent vertices $v,u \in G$ is at most $2m$, where m is the number of edges in G. As T contains $(n-1)$ edges, the sum of the commute times along its edges is at most $2m(n-1) < n^3$, implying $E_s[G] < n^3$.

Let us denote by $MST[G']$ the minimum over all spanning trees of G of the sum of commute times along edges of the spanning tree. In subsequent work, tighter bounds were given on $MST[G']$, implying similar upper bounds on the cover time. Kahn *et al.* [16] showed that for any connected regular graph G, $MST[G'] = O(n^2)$. Coppersmith *et al.* [7] generalized this result to any connected graph, showing that $MST[G'] = \Theta(n^2 d_{ave}(\frac{1}{d})_{ave})$, where d_{ave} is the average degree in G, and $(\frac{1}{d})_{ave}$ is the average of the inverse of the degrees. Feige [10] showed that for every graph $MST[G'] \leq 4n^3/27$ up to low order terms, implying a similar upper bound on the cover time. This upper bound matches (up to low order terms) the cover time for the *lollipop* graph – a clique with $2n/3$ vertices connected to a path of length $n/3$.

All the above upper bounds on the cover time that use $MST(G')$ are not sensitive to the starting vertex s. An alternative way of using $MST(G')$ that is sensitive to the starting vertex (as we wish to do here) was analyzed in Feige [12]. In particular, it was shown that for any graph, $\min_s[E_s[G]] \leq 2n^3/27$, up to low order terms. (This bound is best possible, up to low order terms, as can be seen by considering the *grape* graph – a clique with $2n/3$ vertices connected to two paths of length $n/6$.)

A different approach, suitable also for general Markov chains, was taken by Matthews in [19]. He shows that $E_s[G] \leq \max_{u,v} H[u,v] \operatorname{Ln} n$, where we use $\operatorname{Ln} n$ to denote the $(n-1)$ harmonic number $\sum_{i=1}^{n-1} 1/i$, which is roughly equal to $\ln n$. Clearly, Matthews' bound is off by a factor of at most $\ln n$ if the Markov chain starts at such vertex u. The result implies also an easy $O(\log n)$-approximation of $E_s^+[G]$, where $E_s^+[G]$ is the expected number of steps it takes the chain starting at s to cover G and then return to s. Since $E_s^+[G]$ is larger than both $H[u,s]$ and $H[s,v]$ for any u, v, the triangle inequality for hitting times implies that $\max_{u,v} H[u,v]/2 < E_s^+[G] \leq \max_{u,v} H[u,v](1 + \operatorname{Ln} n)$, and thus $\max_{u,v} H[u,v]$ provides the desired approximation.

Matthews [19] establishes also a lower bound: $E_s[G] \geq \min_{u,v} H[u,v] \operatorname{Ln} n$. While sometimes this bound is far from optimal (e.g., it may happen that $\min_{u,v} H[u,v] = 1$), the method by which it was obtained often proves useful. Refinements of this method were employed by Zuckerman [25] in order to show that the cover time of the $\sqrt{n} \times \sqrt{n}$ mesh is $\Theta(n(\log n)^2)$ (this result was extended by Jonasson and Schramm [21] to show a similar lower bound for arbitrary bounded degree planar graphs), and by Feige [11] who has established a general lower bound $E_s[G] \geq n \ln n$ (up to low order terms) for arbitrary graphs. Recently, Kahn *et al.* [15] used extensions of the Matthews bound in order to design a polynomial time deterministic algorithm that approximates $\max_s E_s[G]$ (the cover time from the worst possible vertex) within a factor of $O((\log \log n)^2)$.

Much information on a Markov chain can be obtained also by considering the eigenvalues and eigenvectors of its transition matrix. Broder and Karlin [4] use this approach to obtain tight bounds on the cover time. In particular, they have shown a $\Theta(n \log n)$ bound for the cover time of regular expander graphs.

For more information on reversible Markov chains and on random walks, see [1].

1.2 Our results

Our main result is a deterministic polynomial time algorithm that given a graph and a starting vertex produces an upper bound on the cover time of random walks. We prove that the ratio between the computed upper bound and the actual cover time is at most polylogarithmic in n. Our algorithm applies also to more general reversible Markov chains, with the same performance guarantee.

The most computationally intensive part of the algorithm is computing hitting times between all pairs of vertices. As shown by Tetali [22], this can be accomplished using a single matrix inversion (and thus in time $O(n^\omega)$, where $2 \leq \omega \leq 3$ is the exponent of matrix multiplication). On these hitting times the algorithm performs $O(n^2)$ simple operations such as additions and comparisons to produce a decomposition of the input graph. Based on the decomposition, the algorithm computes in a straightforward manner an upper bound on the cover time.

Before presenting the main result, we discuss in Section 2 a number of possible approaches for approximating the cover time, and also some results and counterexamples in these directions. Among these results we show (in Section 2.5) that the cover time for arbitrary Markov chains can be approximated within arbitrary small relative error in time $n^{O(\sqrt{n})}$.

2 Possible approaches

2.1 Derandomization

As noted earlier, there is a simple polynomial time randomized algorithm for approximating the cover time for random walks. One may attempt to derandomize this algorithm. Under complexity theoretic assumptions (the existence of pseudorandom generators for BPP) it follows that the cover time can be approximated with arbitrary precision in time $O(2^{n^\epsilon})$ for any $\epsilon > 0$ (see [3, 24, 14]).

Making an observation that the simulation of a random walk requires only logarithmic space, one may attempt to approximate the cover time using an efficient pseudorandom generator for $LOGSPACE$ computations, such as Nisan's generator [20]. However, we do not know whether the known pseudorandom generators for space-bounded computations apply in this case. Keeping track of the vertices the walk visited so far appears to require linear space.

Finding an efficient pseudorandom generator such that the expected cover time for walks that use the output of this generator provably approximates the true cover time remains an open question.

2.2 The starting vertex makes a big difference

As noted earlier, for any graph G, $\max_u[E_u[G]]$ is $\mathrm{Ln}\, n$-approximated by $\max_{u,v}[H[u,v]]$. Hoping that the choice of the starting point s does not have much effect on the value of $E_s[G]$, one could get a good approximation of $E_s[G]$ by always outputting $\max_{u,v}[H[u,v]]$. This hope is, however, illusory, and the resulting approximations are sometimes very poor.

Consider the following example. Let the graph be a clique on $n - \sqrt{n \log n}$ vertices, connected to a path of length $\sqrt{n \log n}$. Let t be a point in the clique, and let s be the far endpoint of the path. For this graph both $H[t,s]$ and $E_t[G]$ are $\Theta(n^{5/2}\sqrt{\log n})$. However, $E_s[G]$ is only $\Theta(n \log n)$. (The expected number of steps to reach the clique from s is $(\sqrt{n \log n})^2 = n \log n$. Thereafter, we use the fact that the cover time for an n vertex clique is roughly $n \ln n$. The fact that in our graph there is a short path connected to the clique does not significantly change the cover time for the clique, because it can be shown that for a walk that starts in the clique, the expected fraction of time steps spent on the path is roughly equal to the ratio between the number of edges along the path and the total number of edges in the graph. This ratio is negligible in our case, and hence the path hardly slows down the cover time for the clique.) This establish a gap of $\Omega(n^{3/2}/\sqrt{\log n})$ between $\max_u[E_u[G]]$ and $\min_u[E_u[G]]$. This example gives the worst possible gap, as shown by the following theorem:

Theorem 1 *For any connected graph G,*

$$\max_v[E_v[G]] \;=\; O\left(\frac{n^{3/2}}{\sqrt{\log n}}\right) \min_u[E_u[G]] \;.$$

Before proving Theorem 1, let us review some known connections between random walks and the electrical resistance [9]. A graph can be viewed as an electrical network where edges represent resistances of 1 ohm. The *effective resistance* between two vertices, $R[u,v]$, is the inverse value of the current between u and v when a voltage of 1 volt is maintained between these two points by an external source. The following identity (see [5] for a proof) relates the commute time and the effective resistance:

Lemma 2 *For any connected graph with m edges, and any two vertices,* $C[u,v] = 2m\, R[u,v]$.

Another identity characterizes the effective resistance in terms of the number of returns to the origin (see [23] for a proof):

Lemma 3 *Let d_u denote the degree of vertex u. For any connected graph and any two vertices, $R[u,v]d_u$ is exactly the expected number of visits at u (including the start) in a random walk starting at u and stopping upon hitting v.*

As a consequence of the two lemmas, we obtain the following result:

Lemma 4 *For any connected graph and any two vertices, $H[u,v] \geq R[u,v]^2$.*

Proof: Let $dist[u,w]$ be the length of the shortest path $P \subseteq G$ between vertices u and w. By monotonicity of the effective resistance, the resistance between u and w in G is at most that in P, i.e., $R[u,w] \leq dist[u,w]$. By the triangle inequality for the effective resistance, $R[w,v] \geq R[u,v] - R[u,w] \geq R[u,v] - dist[u,w]$.

Let $l = dist[u,v]$ denote the distance between u and v, and let $R = R[u,v]$. Consider a random walk starting at u and stopping upon hitting v. Let $u_0 = u$, and let u_i, $i = 1, \ldots, l-1$, be, respectively, a random variable whose value is the first vertex visited by the walk among those that are at distance exactly i from u and whose degree is at least 2. Observe that the walk must encounter such vertices before hitting v. By Lemma 3, the expected number of visits to u_i by our random walk is exactly $R[u_i,v]d_i$, where d_i is the degree of u_i. By the previous argument, this number is at least $(R[u,v] - dist[u,u_i])d_i$. By linearity of expectation, the length of the random walk is at least the sum of the number of visits to $u_0, \ldots, u_{l-1}$, which in turn is at least $\sum_{i=0}^{l-1}(R-i)^+ d_i$, where $(R-i)^+ = \max[(R-i), 0]$. Since all the vertices u_i for $i \geq 1$ have degree ≥ 2, the sum is at least $R + 2((R-1)^+ + \ldots + (R-l+1)^+) \geq R^2$, where we have used here the fact that $l \geq R$. $\square$

We are now ready to prove Theorem 1.

Proof: (of Theorem 1): Let G be an arbitrary connected graph, and $v, u \in V(G)$. Then

$$E_v[G] \leq H[v,u] + E_u[G] \; . \tag{1}$$

By Lemma 4, it holds $H[u,v] \geq R[u,v]^2$. From Lemma 2, one has $H[v,u] < 2m\, R[u,v] < n^2 R[u,v]$. Therefore,

$$H[v,u] \; < \; n^2 H[u,v]/R[u,v] \; . \tag{2}$$

Consider now two cases:

- $R[u,v] \geq \sqrt{n \log n}$.
  Combining (1), (2) and the fact that $E_u[G] \geq H[u,v]$, we conclude in this case that

  $$E_v[G] \leq (\sqrt{n^3/\log n} + 1)E_u[G] \; .$$

- $R[u,v] \leq \sqrt{n \log n}$.
  Combining (1) and Lemma 2, we get

  $$E_v[G] \leq \sqrt{n^5 \log n} \; + \; E_u[G] \; .$$

  But by [11], $E_u[G] = \Omega(n \log n)$, and the statement follows.

$\square$

2.3 Using a low weight Hamiltonian path

As explained in Section 1.1, $HPATH_s[G']$ is an upper bound to $E_s[G]$. This bound is however far from optimal. For example, the cover times for the clique and for the star are $O(n \log n)$, but the shortest Hamiltonian path for either graph has weight $\Omega(n^2)$. Nevertheless, approaches for bounding the cover time based on low weight Hamiltonian paths are informative, and our main approximation algorithm (in Section 3) uses the Hamiltonian path discussed below.

An efficiently computable and particularly informative Hamiltonian path in G' is based on the difference time, $D[u,v] = H[u,v] - H[v,u]$. An important property of the difference time is its additivity:

Lemma 5 *For any reversible Markov chain, and any three vertices u, v, w, $D[u,w] = D[u,v] + D[v,w]$.*

The proof of the lemma appears in [1, 8]. Here we only comment that it is equivalent to the following statement: For any three vertices u, v, w, $H[u,v] + H[v,w] + H[w,u] \;=\; H[u,w] + H[w,v] + H[v,u]$.

As pointed out in [8], Lemma 5 implies that the vertices of G can be arranged in a linear order $v_1, \ldots, v_n$, where $D[v_i, v_j] \leq 0$ whenever $i < j$. Indeed, consider a tournament on the vertices of G where the edge (v_i, v_j) is directed from v_i to v_j whenever $D[v_i, v_j] \leq 0$. By the additivity property of the difference time, the tournament is transitive, and therefore induces a linear order on $V(G)$. We call this the *difference order*. The usefulness of this order in problems related to the cover time of graphs was already demonstrated in [11], where it was used for proving lower bounds on $E_v[G]$.

Define

$$DPATH_{v_i}[G] \;=\; H[v_i, v_1] \;+\; \left(\sum_{j=1}^{i-2} H[v_j, v_{j+1}] \right) \;+\; H[v_{i-1}, v_{i+1}] \;+\; \sum_{j=i+1}^{n-1} H[v_j, v_{j+1}] \;.$$

Proposition 6 *For reversible Markov chains and $DPATH_{v_i}[G]$ as defined above,*

$$\frac{1}{n-1} DPATH_{v_i}[G] \;\leq\; E_{v_i}[G] \;\leq\; DPATH_{v_i}[G] \;.$$

Proof: The upper bound is trivial. To prove the lower bound, we show that each of the $n-1$ terms appearing in the definition of $DPATH_{v_i}[G]$, is at most $E_{v_i}[G]$. Clearly, $H[v_i, v_1] \leq E_{v_i}[G]$. For other terms, whenever a term $H[u,v]$ is included in $DPATH_{v_i}[G]$, it holds (due to the difference ordering) $H[v,u] \geq H[u,v]$. But $E_s[G] \geq \min[H[u,v], H[v,u]]$, since the Markov chain must visit one of the two vertices before the other, and $E_s[G] \geq H[u,v]$ follows. □

The analysis is essentially tight, as for the path the cover time is $O(n^2)$, but $DPATH_{v_i}[\text{path}] = \Omega(n^3)$. The ordering imposed by the difference time on the n-Path $\langle u_1, u_2, \ldots, u_n \rangle$ is easily seen to be $\{u_1, u_n\}, \{u_2, u_{n-1}\}$, *etc.*

2.4 Maximum hitting times, and the key graph

Clearly, $E_s[G] \geq \max_t[H[s,t]]$. It is natural to ask whether $\max_t[H[s,t]]$ provides a good approximation for $E_s[G]$. The answer is negative. In what follows, we describe an example in which the gap between the two numbers is as big as $\Omega\left(n/\log n\right)$. Before doing so, we observe that this gap can never exceed $n-1$: since $E_s[G] \;=\; E_s[\bigcup_{v \in V(G)} \{\text{covering } v\}]$,

$$E_s[G] \;\leq\; \sum_{v \in V(G)} E_s[\text{covering } v] \;=\; \sum_{v \in V(G)} H[s,v] \;\leq\; (n-1) \max_v H[s,v] \;.$$

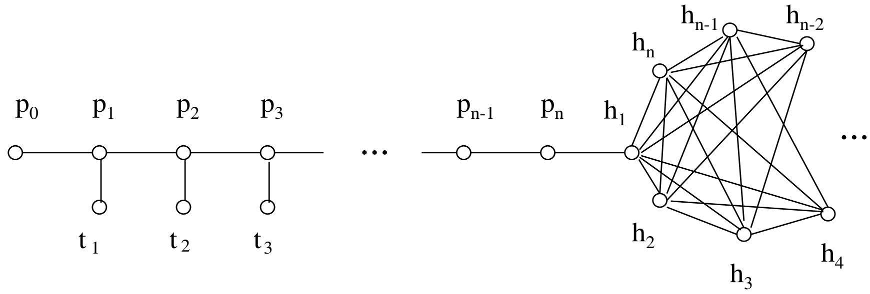

Figure 1: The "key" graph.

For the lower bound, consider the following graph, which we call the *key*. It consists of a path on n vertices $p_0, \dots p_{n-1}$, where to each p_i, $i = 1, \dots, n/2$ there is an attached "tooth" t_i, and an n-vertex clique $h_1, \dots, h_n$, where h_1 is connected to p_{n-1} (see Fig. 1).

Consider a random walk starting at p_0. First we would like to show that $H[p_0, v] = O(n^2)$ for any vertex v. For $v = p_i$ the statement follows at once, since $H[p_0, p_i] < H[p_0, h_1] = O(n^2)$, as the teeth cause but a constant factor slowdown, and without the teeth the hitting time from p_0 to h_1 would be exactly n^2. To establish the statement for $v = t_i$ and for $v = h_i$ it would suffice to show that $H[p_i, t_i] = O(n^2)$ and $H[h_1, h_i] = O(n^2)$ respectively, and then apply the triangle inequality for the hitting times.

Using the formula of Lemma 2 we get $H[p_i, t_i] = 2m - 1 = O(n^2)$ where m is the number of edges in the graph. To estimate $H[h_1, h_i]$, (mentally) attach to h_i a "tail" similar to that attached to h_1 in the key graph. By symmetry, the original $H[h_1, h_i]$ is equal to half of the commute time $H'[h_1, h_i] + H'[h_i, h_1]$ in the new graph, which is, by Lemma 2, $2m' \cdot R'[h_1, h_i] = 2m' \cdot 2/n = O(n)$.

In order to estimate $E_{p_0}[\text{key}]$, define q as the probability that h_1 is hit before all teeth are covered. Since for any $i \leq n/2$ it holds, by Lemma 2, that $H[h_1, t_i] = 2m(n - i + 1) - H[t_i, h_1] = \Omega(n^3 - n^2) = \Omega(n^3)$, it follows that $E_{p_0}[\text{key}] = \Omega(qn^3)$.

Claim 7 *The probability q that in the key graph described above the clique is reached before all teeth are covered satisfies $q = \Omega(1/\log n)$.*

Proof: Let X_i be a random variable taking value 0 if the random walk starting at p_0 hits t_i before h_1, and 1 otherwise. Then $X = \sum_{i=0}^{n/2} X_i$ is the random variable counting the number of uncovered "teeth" upon hitting h_1. Recall the definition of the k-th moment of X, $m_k(X) = E[X^k]$. We shall prove below that $m_1 = \Theta(1)$ and $m_2 = \Theta(\log n)$. Since $m_1^2 \leq qm_2$, as we shall see in a second, this implies the desired bound of $q = \Omega(1/\log n)$. Indeed, by the Cauchy-Schwarz inequality for inner products, $\langle \mathbf{a}, \mathbf{b} \rangle^2 \leq \langle \mathbf{a}, \mathbf{a} \rangle \cdot \langle \mathbf{b}, \mathbf{b} \rangle$; taking $\mathbf{a} = (\sqrt{\Pr[X = 1]}, \dots, \sqrt{\Pr[X = n/2]})$ and $\mathbf{b} = (\sqrt{\Pr[X = 1]} \cdot 1, \dots, \sqrt{\Pr[X = n/2]} \cdot n/2)$ we obtain

$$\left(\sum_{i=1}^{\infty} \Pr[X = i] \cdot i \right)^2 \leq \left(\sum_{i=1}^{\infty} \Pr[X = i] \right) \cdot \left(\sum_{i=1}^{\infty} \Pr[X = i] \cdot i^2 \right) .$$

This last inequality is exactly $m_1^2 \leq qm_2$, as claimed.

We now turn to computing m_1 and m_2. Recall that the standard method for computing the probabilities $r(v)$ that a random walk starting at v will hit a specified vertex v_0 before hitting a specified set S, is to solve the system of "harmonicity" equations satisfied by $r(v)$-s:

$$r(v_0) = 1; \qquad r(u) = 0 \text{ for all } u \in S; \qquad r(u) = \sum_v p_{uv} r(v) \text{ for the rest of the vertices.} \tag{3}$$

(See [9] for a detailed discussion). Using this method with $v = p_0$, $v_0 = h_1$ and $S = t_i$, one obtains at once

$$E[X_i] = \frac{1}{n-i+1} .$$

The calculation is greatly simplified by removing all teeth but t_i, removing the vertices $p_0, \ldots, p_{i-1}, h_2, \ldots, h_n$, and setting $v = p_i$. Since $r(p_0) = r(p_i)$ in the original graph, computing $r(p_i)$ in the simplified graph will yield the same result.

A similar analysis shows that for $i < j$, $E[X_i X_j]$, which can be interpreted as the probability that the random walk starting from p_0 hits h_1 before hitting $\{t_i, t_j\}$, is

$$E[X_i X_j] = \Theta\left(\frac{1}{n\,|i-j|}\right) . \tag{4}$$

Indeed, we first simplify the problem by removing all teeth but t_i and t_j, as well as the vertices $p_0, \ldots, p_{i-1}, h_2, \ldots, h_n$, and looking at the random walk starting from p_i. The harmonicity conditions 3 imply that:

1. $r(t_i) = r(t_j) = 0$; $r(h_1) = 1$.
2. $r(p_j) = r(p_i)(j-i+1)$
   (since r is linear along the path from t_i to p_j, $r(t_i) = 0$, and p_i is the second vertex on the path.)
3. $r(p_j) + (n-j)\delta = r(h_1) = 1$
   (by linearity of r along the path from p_j to h_1, where $\delta = r(p_{j+1}) - r(p_j)$ is the step of the arithmetical progression.)
4. $r(p_j) = \frac{1}{3}(r(p_j) - r(p_i)) + \frac{1}{3}(r(p_j) + \delta)$
   (by harmonicity condition for p_j, and keeping in mind that $r(t_j) = 0$).

Using item 2 to express $r(p_j)$ in terms of $r(p_i)$ in item 4, we obtain that $\delta = r(p_i)(j-i+2)$. Substituting these values in item 3 we obtain $r(p_i)(j-i+1) + (n-j)r(p_i)(j-i+2) = 1$, implying $r(p_i) = \Theta(1/n(j-i))$. Since $r(p_i) = E[X_i X_j]$, he have verified Equation (4).

Hence

$$m_1(X) = E[X] = \sum_{i=0}^{n/2} E[X_i] = \sum_{i=0}^{n/2} \frac{1}{n-i} = \Theta(1) ,$$

and

$$m_2(X) = E[X^2] = \sum_{i=0}^{n/2} E[X_i^2] + \sum_{i \neq j} E[X_i X_j] = \sum_{i=0}^{n/2} E[X_i] + \Theta\left(\sum_{i \neq j} \frac{1}{n\,|i-j|}\right) = \Theta(\log n) .$$

□

By the Claim 7, $E_{p_0}[\text{key}] = \Omega(n^3/\log n)$, while $\max_v H[p_0, v] = O(n^2)$. This establishes the claimed gap of $\Omega(n/\log n)$ between the maximum hitting time and the cover time from the same vertex.

2.5 Approximate inclusion exclusion

In this section we first derive an exact formula expressing the cover time $E_s[G]$ in terms of hitting times $H[s,K]$ (i.e., the expected time it takes to hit a set $K \subseteq V(G)$ starting from s), then show how $E_s[G]$ can be approximated using much less such terms. The derivation is based on the inclusion-exclusion principle. We remark that formulas similar to that of Proposition 8 were previously used in similar contexts. See [13] for example.

Proposition 8 $E_s[G] = \sum_{K\subseteq V(G)} (-1)^{|K|+1} H[s,K]$.

Proof: Let $F_s[v|i]$ denote the event that a Markov chain starting at s fails to visit the vertex v by step i, and let $F_s[G|i] = \cup_v F_s[v|i]$ be the event that the Markov chain fails to cover the graph by step i. Observe that $E_s[G] = \sum_{i\geq 0} Pr\left[F_s[G|i]\right]$. Now, by the inclusion-exclusion formula,

$$Pr\left[F_s[G|i]\right] = Pr\left[\cup_v F_s[v|i]\right] = \sum_{K\subseteq V(G)} (-1)^{|K|+1} Pr\left[\cap_{v\in K} F_s[v|i]\right] .$$

Thus,

$$E_s[G] = \sum_{i\geq 0}\sum_{K\subseteq V(G)} (-1)^{|K|+1} Pr\left[\cap_{v\in K} F_s[v|i]\right] = \sum_{K\subseteq V(G)} (-1)^{|K|+1} \left(\sum_{i\geq 0} Pr\left[\cap_{v\in K} F_s[v|i]\right]\right) .$$

Since $\sum_{i\geq 0} Pr\ \left[\cap_{v\in K} F_s[v|i]\right] = \sum_{i\geq 0} Pr\ \left[F_s[K|i]\right]$ is exactly $H[s,K]$, the claim follows. □

Under reasonable assumptions on the transition probabilities (e.g., there are binary fractions with n-bit representations), any $H[s,K]$ can be computed in time polynomial in n. However, since there are 2^n such terms, Proposition 8 alone does not yield an efficient algorithm for computing the cover time.

Studies [18, 17] concerning how well the probability of an union of n events can be approximated by probabilities of intersections up to size k, show that when $k \simeq \sqrt{n}$, the relative error becomes very small. The probability of the union is then approximated by a weighted sum of the probabilities of the intersection, where the weights $\alpha^{k,n}$ (explicitly given in [18]) depend only on k, n, and the size of the actual intersection involved. Arguing exactly as in the proof of Proposition 8, we conclude that $E_s[G]$ can be arbitrarily well approximated by the sum

$$\sum_{K,|K|\leq k} \alpha^{k,n}_{|K|} H[s,K] ,$$

where $k \simeq \sqrt{n}$. The coefficients $\alpha^{k,n}_{|S|}$ are effectively computable. Since there are $n^{O(\sqrt{n})}$ subsets of size $O(\sqrt{n})$, we conclude that

Theorem 9 *For every $\rho > 1$ and every starting state, there is a $n^{O(\sqrt{n})}$-time deterministic algorithm that approximates the cover time of an n state Markov chain within a ratio of ρ.*

Can this approach be used to get a polynomial approximation algorithm? Arguing as before, we conclude that for a subset K of k vertices on an n vertex Markov chain, the expected time until S is covered can be approximated arbitrarily well in time $k^{O(\sqrt{k})}\mathrm{poly}(n)$. The time is polynomial in n when $k = O((\log n/\log\log n)^2)$. By partitioning the state space of a Markov chain into $O(n(\log\log n)^2/(\log n)^2)$ such subsets, we can compute the cover times (starting at s) for each such subset, and the cover time of the whole state space lies between the maximum of these cover times and the sum of these cover times.

Corollary 10 *For every Markov chain and every starting state, there is a polynomial time deterministic algorithm that approximates the cover time within a ratio of* $O(n(\log\log n)^2/(\log n)^2)$.

The main result of this paper gives a much better approximation ratio for the cover time of reversible Markov chains. We suspect that better approximation ratios are achievable also for nonreversible Markov chains.

3 The algorithm

Let G be the input graph of size n. Order the vertices of G by the difference order defined in Section 2.3. In what follows, for the sake of simplicity we shall call v_i simply i. Recall that by definition of the difference order, $i \leq j$ implies $D[i,j] \leq 0$, or equivalently $H[i,j] \leq H[j,i]$.

The first observation is that in order to obtain an approximation of *any* $E_i[G]$, it suffices to approximate $E_1[G]$.

Lemma 11 *Given an* α*-approximation of* $E_1[G]$*, i.e., a number* $\tilde{E}_1[G]$ *such that* $\alpha E_1[G] \geq \tilde{E}_1[G] \geq E_1[G]$*, a* $(2\alpha+1)$*-approximation of* $E_i[G]$ *can be obtained in deterministic* $PTIME$.

Proof: The following three inequalities hold in an obvious fashion:

$$\begin{aligned}
E_i[G] &\leq H[i,1] + E_1[G] \; ; \\
E_1[G] &\leq H[1,i] + E_i[G] \;\leq\; H[i,1] + E_i[G] \; ; \\
H[i,1] &\leq E_i[G] \; .
\end{aligned}$$

Hence, given an α-approximation $\tilde{E}_1[G]$ of $E_1[G]$, we conclude that

$$E_i[G] \;\leq\; \tilde{E}_1[G] + H[i,1] \;\leq\; \alpha E_1[G] + H[i,1] \;\leq\; \alpha E_i[G] + (\alpha+1)H[i,1] \leq (2\alpha+1)E_i[G] \; .$$

and thus $\tilde{E}_1[G] + H[i,1]$ is a $(2\alpha+1)$-approximation of $E_i[G]$.

This, combined with the fact that $H[i,1]$ is computable in $PTIME$, concludes the argument. □

In what follows we concentrate on approximating $E_1[G]$.

Let $\mathcal{P}$ be a partition of the vertices of G into consecutive (with respect to the difference order) intervals, all having the following property. For an interval $I \in \mathcal{P}$, let $D[I]$ denote the maximum difference time between two vertices in I; it is attained between the right and the left ends of I. Let also $M[I]$ be the maximum hitting time between two vertices $i, j \in I$ with $i < j$. The required property is:

For every interval $I \in \mathcal{P}$*, it holds* $D[I] \leq M[I]$*. Moreover, every* $I \in \mathcal{P}$ *(with the possible exception of the rightmost interval in* $\mathcal{P}$*) is maximal in the following sense: the interval* I' *obtained from* I *by adding the vertex immediately following it (to the right), does not satisfy this inequality anymore.*

Such a partition can be obtained by scanning the vertices according to the difference order, and closing an interval every time it becomes unextendible. Notice that some intervals may consist of a single point, in which case both $M[I]$ and $D[I]$ are 0. Since all the quantities involved can be computed effectively and deterministically, the partition is constructible in $PTIME$.

Matthews' Theorem (see [19]) claims that the cover time of a reversible Markov chain of size n is always bounded from above by $\operatorname{Ln} n$ times the largest hitting time. This implies in our case that for every $I \in \mathcal{P}$ and $v \in I$,

$$E_v[I] \;\leq\; \max_{u,w \in I} H[u,w] \operatorname{Ln}(|I|) \;\leq\; (M[I] + D[I]) \operatorname{Ln}(|I|) \;\leq\; 2 \operatorname{Ln}(|I|)\, M[I] \;.$$

Observe also (for future use) that $M[I]$ provides a lower bound on $E_v[I]$. Indeed, assume $M[I] = H[i,j]$ for some $i < j$ in I. The expected time for covering this pair of vertices alone is at least $\min\{H[i,j], H[j,i]\} = H[i,j] = M[I]$.

Let $\mathcal{P} = \langle I_1, I_2, \ldots, I_s \rangle$. By the above observation, the following quantity provides an upper bound on $E_1[G]$:

$$UB \;=\; \sum_{i=1}^{s} 2 \operatorname{Ln}(|I_i|)\, M[I_i] \;+\; \sum_{i=1}^{s-1} \max_{v \in I_i} \min_{u \in I_{i+1}} H[v,u] \;.$$

Our algorithm for approximating $E_1[G]$ is:

1. *Compute all the hitting times $H[i,j]$.*
2. *Arrange the vertices by the difference order.*
3. *Construct a partition $\mathcal{P}$ as described above.*
4. *Compute UB and output it.*

Theorem 12 *For random walks on size-n graphs,*

$$\frac{1}{polylog(n)} UB \;\leq\; E_1[G] \;\leq\; UB \;.$$

The proof of Theorem 12 appears in Section 4. Here we give an informal overview of the main ideas of our proof.

Recall that the cover time of a graph starting from the worst vertex can be approximated within a $\log n$ factor (e.g., via Matthews' bound), and that the problem that we are facing is to approximate the cover time starting at an arbitrary vertex. Our algorithm partitions the graph into parts that we call intervals, as they are indeed intervals in the difference order. An important property of each such part I is that its cover time hardly depends on the starting vertex within the part. As noted above, the cover time of the part is at least $M[I]$ regardless of the starting vertex, and the hitting time from the worst starting vertex to the best one is at most $M[I] + D[I] \leq 2M[I]$. Consequently,

$$\max_{u \in I} E_u[I] \leq 3 \min_{u \in I} E_u[I] \,.$$

Hence within an interval, it suffices to approximate the cover time from the worst possible starting vertex, to get an approximation for the cover time from any possible starting vertex. And, as we have seen, the Matthews bound implies that $M[I]$ is an $\operatorname{Ln} n$-approximation for the worst starting point.

(**Remark:** The reader may wonder at this point why we do not use for the same purpose the results of [15], which give a $O((\log\log n)^2)$, rather than $\operatorname{Ln} n$, approximation for the worst starting point. The main reason for not doing so is that the results of [15] were not available at the time we first obtained the results of the current paper. See also the discussion in Section 5.)

Our upper bound UB on the cover time of the whole graph reflects a scenario that the graph is covered sequentially, one interval after another, according to the difference order. Within an interval, the vertices may be covered in arbitrary order. Thus, UB is composed of two components: one is the sum of the cover times of each of the individual parts (i.e., intervals), and the other is the expected time it takes to move between the parts.

By the way of construction of UB it is clear that it constitutes an upper bound on $E_1[G]$: our scenario cannot be better than what actually happens. The hard part of the proof of Theorem 12 is to establish that UB is also a lower bound (up to a polylogarithmic factors). The reason for this difficulty is that our scenario can be (and is) far from true, and thus it may cause large overcount. The random walk need not cover the parts sequentially, and covering them in parallel might be much quicker. For example, the walk might start covering a new part I_{new} before completely covering a part I_{old} that precedes it in the difference order. The analysis pends on an observation that in this case the walk will need to return from I_{new} to I_{old}, and that the expected time to do so will provide a sufficient compensation for the "deviation". The reason is that, by befinition of I_{old}, for any I_{new} lying to the right of it (in the difference order), the return time from I_{new} to I_{old} is at least as large as the time to go from I_{old} to I_{new}, and is also comparable to the whole cover time of I_{old}.

Summing up the above intuitive argument, if the walk covers parts in parallel, it may not "pay" for some of the terms in UB, but then it will "pay" for expensive return times that compensate for the difference.

The above overview is informal and not completely accurate. In the next section we give a formal proof, and in the process lose some additional logarithmic factors in the approximation ratio. We believe that a more careful analysis might regain some of these factors.

4 Performance guarantee

We shall first analyze the performance of the algorithm for random walks on graphs with no parallel edges and no self loops. Then, in Section 4.1, we will argue that the same analysis with one small modification yields the same performance guarantee for arbitrary reversible Markov chains.

The upper bound being trivial, this entire section is dedicated to obtaining a lower bound; the goal is to prove that $E_1[G] \geq UB/\mathrm{polylog}(n)$.

Proof: Instead of working directly with UB, we shall work will the following expression:

$$UB^* \;=\; 2\,\mathrm{Ln}\,(|I_s|)\,M[I_s] \;+\; \sum_{i=1}^{s-1} \{2\,\mathrm{Ln}\,(|I_i|)\,M[I_i] \;+\; H[right(I_i), left(I_{i+1})]\} \;\;,$$

where $left(I)$ and $right(I)$ mean, respectively, the leftmost and the rightmost points in the interval I. Since for every $i = 1, \ldots, s-1$,

$$\max_{v \in I_i} \min_{u \in I_{i+1}} H[v,u] \;\leq\; M[I_i] + H[right(I_i), left(I_{i+1})]\,,$$

clearly $UB^* \geq c\,UB$ for some constant $c > \frac{1}{2}$.

Let us define w_i, $i = 1, 2, \ldots, s$, as the value of the i-th term in UB^*:

$$w_i \;=\; \{2\,\mathrm{Ln}\,(|I_i|)\,M[I_i] \;+\; H[right(I_i), left(I_{i+1})]\} \;\;.$$

For $i = s$ the second term disappears.

Next, let us extend every $I_i \in \mathcal{P}$, $i = 1, 2, \ldots, s-1$, by adding to it the vertex immediately following it (on the right). Call the new interval J_i. Also, let $J_s = I_s$. The intervals in $\{J_i\}$ are consecutive, consist of at least two points, and may overlap only at an endpoint per (adjacent) pair; thus the family is naturally ordered in accordance with the difference order.

Notice that for every $i = 1, 2, \ldots, s$,

$$\min_{v \in J_i} E_v[J_i] \;\geq\; \frac{1}{1 + 2\ln n} w_i \ .$$

Indeed, as we have pointed out previously, for any $v \in J_i$,

$$E_v[J_i] \;\geq\; M[J_i] \;\geq\; M[I_i],\ H[right(I_i), left(I_{i+1})] \ .$$

Similarly, for every $i = 1, 2, \ldots, s-1$, it holds

$$D[J_i] \;\geq\; \frac{1}{1 + 2\ln n} w_i \ ,$$

since by definition of I_i, $D[J_i] > M[I_i],\ H[right(I_i), left(I_{i+1})]$.

In what follows we shall refer to w_i as the *weight* of J_i, and write it as $w(J_i)$.

In the following three steps we shall construct a subfamily of $\{J_i\}$ with certain regularity properties.

Step 1: Classify the values w_i, $i = 1, \ldots, s$, according to the largest integer power p_i of 2 such that $w_i \geq 2^{p_i}$. Since $1 \leq w_i \leq n^3 \log n$, it follows that the class with the largest sum of weights will weigh at least $\Omega(\frac{1}{\log n}) \sum_i w_i = \Omega(\frac{1}{\log n}) UB^*$. Let $K \subset [1 \ldots s]$ be the subset of indices corresponding to this class. Define a new subfamily of intervals $\mathcal{J}^1 = \{J_i | i \in K\}$. By the fashion in which $\mathcal{J}^1$ was constructed, all the intervals in this subfamily have approximately (up to a factor of 2) the same weight. Let the minimum weight of an interval in $\mathcal{J}^1$ be W.

Step 2: Let $D = D[n, 1]$. For every $J \in \mathcal{J}^1$, define the sets define $S_h^J \subset V(G)$, $h \in \{0, 1, \ldots, \lceil \log_2 D \rceil + 1\}$ as follows. Let S_0^J be J itself, and let $S_1^J \subset V(G)$ be the set of all vertices v such that $0 < D[left(J), v] < 2$. For the remaining h, define $S_h^J \subset V(G)$ as a set of vertices v such that $2^{h-1} \leq D[left(J), v] < 2^h$. Note that for every J, the sets S_h^J form a partition of the vertices that are to the left of J in the difference order, and each such part forms a consecutive interval in the difference order. The larger h is, the further away S_h^J is from J (and the closer to vertex 1). (We note a technicality here: $D[left(J), v] > 0$ for every v to the left of J in the difference order, because otherwise the vertex $left(J)$ would not start a new interval in the partition of Section 3.)

Now define a random variable Δ_J so that Δ_J is the index d such that by the time J was first hit by a random walk starting at 1, all S_h^J with $h > d$ were covered, while S_d^J was not. Since Δ_J may assume only $2 + \log_2 D = O(\log n)$ values, the most probable value $d(J)$ of Δ_J occurs with probability $\Omega(\frac{1}{\log n})$.

Next, we classify the members of $\mathcal{J}^1$ according to their $d(J)$. Since there are at most $O(\log n)$ such values, the most numerous class must consist of at least $\Omega(\frac{1}{\log n}) |\mathcal{J}^1|$ members. Let $\mathcal{J}^2 \subset \mathcal{J}^1$ be the set of members of this class. Let d be the (integer) value corresponding to this class, and define $R = 2^d$.

An important observation is

$$\sum_{J \in \mathcal{J}^2} w(J) \;\geq\; |\mathcal{J}^2| \cdot W \;=\; \left(\frac{|\mathcal{J}^2|}{|\mathcal{J}^1|} \right) |\mathcal{J}^1| \cdot W \;\geq\; \left(\frac{|\mathcal{J}^2|}{2|\mathcal{J}^1|} \right) \sum_{J \in \mathcal{J}^1} w(J) \;\geq\; \Omega\left(\frac{1}{\log^2 n} \right) UB^* \ .$$

To summarize:
We have constructed a family $\mathcal{J}^2 \subseteq \{J_i\}$ of consecutive intervals, such that:

1. The weight of every $J \in \mathcal{J}^2$ is between W and $2W$;

2. For every $J \in \mathcal{J}^2$, $\min_{v \in J} E_v[J] \geq \Omega(\frac{1}{\log n})\, w(J)$;

3. For every $J \in \mathcal{J}^2$, $D[J] \geq \Omega(\frac{1}{\log n})\, w(J)$;

4. $\sum_{J \in \mathcal{J}^2} w(J) \geq \Omega(\frac{1}{\log^2 n})\, UB$;

5. The following property holds when $d > 1$ (respectively, $R \geq 4$). For every $J \in \mathcal{J}^2$, the random walk starting from v_1 will with probability $\Omega(\frac{1}{\log n})$ behave as follows. It will first cover all the vertices to the left of J whose (difference) distance from J is more than R, not cover at least one vertex (say x) whose (difference) distance from J satisfies $D[left(J), x] > R/2$, then first touch J, then at some time before the interval $[1 \ldots right(J)]$ is covered, will need to perform a digression back to x. The expected number of steps such a digression will take is at least $R/2$, as $\min_{u \in J} H[u, x] \geq D[u, x] \geq D[left(J), x]$.

We need one more step of sparsification.

Step 3: Perform the following procedure. While not all the intervals in $\mathcal{J}^2$ are eliminated, choose the rightmost surviving J, mark and remove it, then remove all the members of $\mathcal{J}^2$ whose minimal difference distance from J is $\leq R$.

After the procedure terminates, define the new subfamily $\mathcal{J}^3 = \langle J_1^*, J_2^*, \ldots, J_t^* \rangle$, as the set of all marked intervals in $\mathcal{J}^2$. In addition to the above listed properties of intervals in $\mathcal{J}^2$, $\mathcal{J}^3$ satisfies also

6. For every $k < t$, and every pair of vertices $v \in J_k^*$, $u \in J_{k+1}^*$, the (difference) distance $D[u, v]$ is $> R$.

In particular, note that the intervals forming $\mathcal{J}^3$ are disjoint.

How many intervals of $\mathcal{J}^2$ survived to be included in $\mathcal{J}^3$? Each round of the elimination procedure produces one member of $\mathcal{J}^3$, and eliminates all the remaining intervals in $\mathcal{J}^2$ which intersect a certain interval of the difference length R. If $d \in \{0, 1\}$ (respectively, $R \leq 2$) the number of eliminated intervals per round is at most two. This follows from the fact that for each interval J, $D[J] > M[J] \geq 1$. If $d \geq 2$, we use the fact that the "difference length" $D[J]$ of any interval in $J \in \mathcal{J}^2$ is at least $\frac{W}{1 + 2 \ln n}$, to conclude that one round of the procedure can remove at most $2 + (1 + 2 \ln n) R/W$ intervals. Hence the size of $\mathcal{J}^3$, being equal to the number of rounds performed, is at least $|\mathcal{J}^2|/3$ when $d \leq 1$, and

$$|\mathcal{J}^3| \geq \frac{|\mathcal{J}^2|}{2 + (1 + 2\, \mathrm{Ln}\, n) R/W}$$

when $d \geq 2$.

Let $C(k) = C_1([1 \ldots right(J_k^*)])$ be the random variable corresponding to the number of steps it takes a random walk starting at v_1 to cover all vertices in this interval. Define for convenience $J_0^* = \emptyset$, $C(0) = 0$. Since $E_1[G] \geq E[C(t)]$, it suffices to show for the lower bound that $E[C(t)]$ is large.

In the concluding argument we consider two different cases. Informally, the first of these cases corresponds to the case when the "scenario" presented in Section 3 of how the random walk progresses in a sense reflects its actual behavior; the second case corresponds to the case the random walk behaves essentially differently.

Case 1: $|\mathcal{J}^3| \geq |\mathcal{J}^2|/3$.

By Properties 5 and 6, for every $k = 1, 2, \ldots, t$, the random walk will with probability $\Omega(\frac{1}{\log n})$ reach J_k^* only after all the vertices up to $right(J_{k-1}^*)$ were already covered. This implies

$$E[C(k) - C(k-1)] \geq \Omega\left(\frac{1}{\log n}\right) \min_{v \in J_k^*} E_v[J_k^*] \, .$$

By Property 2, $\min_{v\in J_k^*} E_v[J_k^*] \;\geq\; \Omega(\frac{1}{\log n})W$. Thus

$$E[C(t)] \;\geq\; \sum_{k=1}^{t} E[C(k) \;-\; C(k-1)] \;\geq\; \Omega\left(\frac{1}{\log^2 n}\right) W\cdot|\mathcal{J}^3|\;.$$

Since by our assumptions for Case 1, the sizes of $\mathcal{J}^3$ and $\mathcal{J}^2$ differ by at most a constant factor, the Property 4 implies

$$E[C(t)] \;\geq\; \Omega\left(\frac{1}{\log^2 n}\right) W\cdot|\mathcal{J}^2| \;\geq\; \Omega\left(\frac{1}{\log^4 n}\right) UB\;.$$

Case 2: $|\mathcal{J}^3| < |\mathcal{J}^2|/3$.
Note that in this case necessarily $d \geq 2$, $R \geq 4$, and $(1+2\,\mathrm{Ln}\, n)R \;\geq\; W$.
By Properties 5 and 6, for every $k = 1,\ldots,t$, the random walk will with probability $\Omega(\frac{1}{\log n})$ perform an $R/2$-digression after covering $[1\ldots right(J_{k-1}^*)]$ and prior to covering $[1\ldots right(J_k^*)]$. Consequently,

$$E[C(k) \;-\; C(k-1)] \;\geq\; \Omega\left(\frac{1}{\log n}\right) R\;,$$

and therefore

$$E[C(t)] \;\geq\; \sum_{k=1}^{t} E[C(k) \;-\; C(k-1)] \;\geq\; \Omega\left(\frac{1}{\log n}\right) R\cdot|\mathcal{J}^3|\;.$$

By our assumptions for Case 2, $R\cdot|\mathcal{J}^3| \;\geq\; \Omega\left(\frac{1}{\log n}\right) W\cdot|\mathcal{J}^2|$. Combining this with Properties 1 and 4 we conclude that

$$E[C(t)] \;\geq\; \Omega\left(\frac{1}{\log^2 n}\right) W\cdot|\mathcal{J}^2| \;\geq\; \Omega\left(\frac{1}{\log^4 n}\right) UB\;.$$

□

4.1 Arbitrary reversible Markov chains

The algorithm presented in Section 3 can be applied without change to arbitrary reversible Markov chains, and not just to random walks on undirected graphs.

Theorem 13 *For arbitrary reversible Markov chain M with n states, the upper bound UB obtained by the algorithm of Section 3, satisfies*

$$\frac{1}{polylog(n)} UB \;\leq\; E_1[M] \;\leq\; UB\;.$$

The proof of this theorem is only slightly more elaborate than that of Theorem 12. The only point which should be altered is a more careful limitation of the range of values for the parameters p_i in Step 1, and for parameter h in Step 2. Whereas for random walks all hitting times are are between 1 and n^3, and hence there are at most $O(\log n)$ possible values for the above two parameters, this might not hold for arbitrary reversible Markov chains.

In order to restrict p_i to $O(\log n)$ possible values, we observe that we need only consider the $w_i = w_{max}$ who has the maximum value, and those w_i that are within a factor of at most n from w_{max}. The total weight of all other w_i is less than w_{max}, and they can be neglected.

In order to restrict h to $O(\log n)$ values, we classify all $h > \log(UB)$ in one "top" set, and all $h < \log(UB) - 2\log n$ in one "bottom" set. If the top set turns out to be more probable, the proof is greatly simplified: with probability $\Omega(1/\log n)$ there is a digression lasting UB units of time, hence $E_1[G] \geq \Omega(UB/\log n)$. If the bottom set turns out to be most probable, the analysis proceeds as in Case 1, as R is much smaller than W. If one of the remaining $O(\log n)$ sets is the most probable, the analysis is identical to the one used in the proof in Theorem 12.

4.2 Examples

Our algorithm approximates the cover time based on the matrix of all pairs hitting times. In some cases hitting times can be approximated fairly well using general principles, such as symmetry arguments, or the following well known proposition (following at once from Lemma 2):

Proposition 14 *Let (u,v) be a cut edge of a graph, and let $m_{u|v}$ denote the number of edges that remain in u's connected component if the edge (u,v) is cut. Then $H(u,v) = 2m_{u|v} + 1$.*

We use such estimates on the hitting times to estimate the output UB of our algorithm (and hence the cover time) in several toy examples.

Clique: Here, by symmetry, all hitting times are the same $(n-1)$, and hence all difference times are 0. All vertices will belong to the same interval, and the algorithm will output $UB = (n-1)\,\mathrm{Ln}\, n$, which is correct (see also remark that follows).

Path: The order imposed by the difference time on the n-path $\langle u_1, u_2, \ldots, u_n\rangle$ is $\{u_1, u_n\}, \{u_2, u_{n-1}\}$, *etc.* All vertices will belong to the same interval, for the reason that $H[u_1, u_n] \geq \max_{u,v} D[u,v]$. The algorithm will output $UB = (n-1)^2\,\mathrm{Ln}\, n$, which is off by a factor of $\mathrm{Ln}\, n$. This can be tightened. Rather than naively applying the algorithm, observe that in order to cover a tree it suffices to cover all leaves. Our algorithm works without change if we want to estimate $E_v[S]$ where S is s subset of vertices of graph G. Since the path has only two leaves, the algorithm would in fact give the correct answer.

Remark: The attentive reader may wonder why $\mathrm{Ln}\, 2 = 1$. We remind that we use $\mathrm{Ln}\, n = \sum_{i=1}^{n-1} 1/i$. This is what Matthews' argument gives. □

Lollipop: A lollipop has a path (stick) of length $n/3$ connected to a clique (candy) of size $2n/3$. The difference order for the lollipop arranges the vertices starting at the end of the path, following the path, then all vertices of the clique in one group, except for the connection point to the stick, which is the last vertex. This will be partitioned into intervals, where each vertex of the path is a distinct interval, and all vertices of the clique are one interval. We obtain $UB \simeq (n/3)^2 + (2n/3)\ln(2n/3)$, which is essentially correct.

Lollipop+: If we add one more vertex v to the lollipop extending out of the clique, then in the difference order v enters just before the vertex $p_{n/3}$ of the path closest to the clique. This has the effect of having v and $p_{n/3}$ join the clique as one interval (since $H[v, p_{n/3}]$ is larger than any of the difference times that remain), and we obtain $UB \simeq (n/3)^2 + (2n/3)^2\ln(2n/3)$, which is off by a logarithmic factor.

Key: Recall that the key graph (Section 2.4) is composed of a path $p_0, p_1, \ldots, p_{n-1}$, respective teeth $t_1, \ldots, t_{n/2}$, and a clique $h_1, \ldots, h_n$, where h_1 is connected to p_{n-1}. The difference order on the key is

$$\{p_0, t_1\}, t_2, p_1, t_3, p_2, \ldots, t_{n/2}, p_{n/2-1}, p_{n/2}, \ldots, p_{n-1}, \{h_2, \ldots, h_n\}, h_1$$

When we partition this order into intervals, every tooth t_i is in the same interval with p_{i-1}, but not in the same interval as t_{i-2}. Since $H[t_i, p_{i-1}] = \Theta(n^2)$, we obtain that $UB = \Theta(n^3)$. This provides an alternative proof to our claim in Section 2.4 that $E_{p_0}[\mathrm{key}] = \Omega(n^3/\mathrm{polylog}(n))$.

5 Recent developments

The approximation algorithm of Section 3 for the cover time was revisited in the M.Sc. thesis of Eden Chlamtac (see [6]). The analysis of the algorithm was tightened, shaving off a factor of $O(\log^2 n)$ in the approximation ratio. Moreover, the algorithm was slightly modified so that one could plug in the improved $O((\log\log n)^2)$ approximation ratios for the cover time (from worst possible starting vertex) of [15], rather than the $\ln n$ approximation ratio of the Matthews bound. This results in a further $O(\log n/(\log\log n)^2)$ improvement in the approximation ratio, giving an approximation ratio of $O(\log n(\log\log n)^2)$. Anticipating possible further improvements beyond those of [15], the main result of [6] is the following.

Theorem 15 *Given a deterministic approximation algorithm A for the maximum cover time $\max_s[E_s[M]]$ of reversible Markov chains with approximation ratio $c(n)$, one can give a deterministic approximation algorithm B for the general cover time $E_s[M]$ of reversible Markov chains with approximation ratio $O(c(n)\log n)$.*

The algorithm B referred to in Theorem 15 is basically the same algorithm that is presented in Section 3. Technically, the only modification is that in B the partitioning of the vertices into consecutive intervals with respect to the difference order is done with the help of Algorithm A. The algorithm of Section 3 is an instantiation of Algorithm B to the case that Algorithm A is the Matthews bound.

Acknowledgements

We thank Ran Raz and Charles Rackoff for helpful discussions.

References

[1] D. J. Aldous, J. A. Fill, "Reversible Markov chains and random walks on graphs". *A draft of book in writing*, 1999.

[2] R. Aleliunas, R. Karp, R. Lipton, L. Lovász, and C. Rackoff. "Random walks, universal traversal sequences, and the complexity of maze problems". In *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, October 1979.

[3] M. Blum, S. Micali. "How to generate cryptographically strong sequences of pseudo random bits". *SIAM J. on Computing*, Vol. 13, 1984, 850–864.

[4] A. Broder and A. Karlin. "Bounds on the cover time". *Journal of Theoretical Probability*, 2(1):101–120, January 1989.

[5] A. Chandra, P. Raghavan, W. Ruzzo, R. Smolensky, and P. Tiwari. "The electrical resistance of a graph captures its commute and cover times". *Computational Complexity* 6 (1996/97), no. 4, 312–340.

[6] E. Chlamtac and U. Feige. "Improved approximation of the minimum cover time". *Technical Report MCS02-16*, The Weizmann Institute of Science, Faculty of Mathematical Sciences, December 2002.

[7] D. Coppersmith, U. Feige, and J. Shearer. "Random Walks on Regular and Irregular Graphs". *SIAM J. on Discrete Math.* 9 (1996), no. 2, 301–308.

[8] D. Coppersmith, P. Tetali, and P. Winkler. "Collisions among random walks on a graph". *SIAM J. on Discrete Math.,* 6(3):363–374, August 1993.

[9] P. G. Doyle and J. L. Snell. "Random Walks and Electrical Networks". *The Mathematical Association of America*, 1984.

[10] U. Feige. "A Tight Upper Bound on the Cover Time for Random Walks on Graphs". *Random Structures & Algorithms*, 6(1), 1995, 51–54.

[11] U. Feige. "A Tight Lower Bound on the Cover Time for Random Walks on Graphs". *Random Structures & Algorithms*, 6(4), 1995, 433–438.

[12] U. Feige. "Collecting Coupons on Trees, and the Analysis of Random Walks". *Comput. Complexity* 6 (1996/97), no. 4, 341–356.

[13] P. Flajolet, D. Gardy, L. Thimonier. "Birthday paradox, coupon collectors, caching algorithms and self-organizing search". *Discrete Applied Mathematics. Combinatorial Algorithms, Optimization and Computer Science* 39 (1992), no. 3, 207–229.

[14] J. Hastad, R. Impagliazzo, L. Levin, M. Luby. "A pseudorandom generator from any one-way function". *SIAM J. on Computing* 28 (1999), no. 4, 1364–1396.

[15] J. Kahn, J.H. Kim, L. Lovasz and V.H. Vu. "The cover time, the blanket time, and the Matthews bound". *Proc. of 41st IEEE Symp. on Foundations of Computer Science*, 2000, pp. 467–475.

[16] J. Kahn, N. Linial, N. Nisan, and M. Saks. "On the cover time of random walks on graphs". *Journal of Theoretical Probability*, 2(1):121–128, January 1989.

[17] J. Kahn, N. Linial, A. Samorodnitsky. "Inclusion-exclusion: exact and approximate". *Combinatorica* 16 (1996), no. 4, 465–477.

[18] N. Linial, N. Nisan. "Approximate inclusion-exclusion". *Combinatorica*, 10(1990), 349–365.

[19] P.C. Matthews. "Covering Problems for Brownian Motion on Spheres". *Ann. Probab., 16:189–199, 1988.*

[20] N. Nisan. "Pseudorandom generators for space bounded computation". *Proc. 22nd ACM Symposium on Theory of Computing*, 1990, 204–212.

[21] Jonasson, J., Schramm, O. "On the cover time of planar graphs". *Electron. Comm. Probab.* 5 (2000), 85–90.

[22] P. Tetali. "Design of online algorithms using hitting times". *SIAM J. on Computing* 28 (1999), no. 4, 1232–1246.

[23] P. Tetali. "Random walks and the effective resistance of networks". *Journal of Theoretical Probability*, 4:101–109, 1991.

[24] A.C. Yao. "Theory and applications of trapdoor functions". *Proc. of 23rd IEEE Symp. on Foundations of Computer Science*, 1982, 80–91.

[25] D. Zuckerman. "A Technique for Lower Bounding the Cover Time". *SIAM J. on Discrete Math.*, 5:81–87, 1992.