Deterministic approximation of the cover time
(Preliminary version)

Uriel Feige [†] Yuri Rabinovich [‡]

Abstract

The cover time is the expected time it takes a simple random walk to cover all vertices of a graph. It arises in numerous questions related to the behaviour of random walks on graphs. 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 $PTIME$. 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.

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*. 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 a digraph, whose vertices represent the states of the chain, with the weights of the edges $\{(v_i, v_j)\}$ given by the transition probabilities $\{p_{ij}\}$.

A simple *random walk* on an undirected graph is a special case of a Markov chain. 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 neighbours of the current vertex.

A *reversible* (equivalently, *time-reversible*) Markov chain can be viewed as a random walk on an edge-weighted undirected graph. Its transition matrix is obtained by considering a nonnegative symmetric matrix W (specifying the edge weights), and scaling each of its rows to sum up to 1.

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 simple random walk on the graph G, we shall write simply $E_s[G]$. The cover is one of the most important parameters of the chain M. It naturally arises in various problems related to the behaviour of random walks, as well as in a number of application.

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* $D[u, v] = H[u, v] - H[v, u]$. The hitting times (and consequently the commute and the difference times) can be computed in polynomial time. In order to do this, one uses the fact that the hitting times $H[u_i, v]$, $i = 1, .., n$, satisfy the following n linear equations: For $u_i = v$, $H[v, v] = 0$, and for $u_i \neq v$, $H[u_i, v] = 1 + \sum_{v_j} p_{ij} H[u_j, v]$. The resulting system of equations always has a unique solution (namely, $H[u_i, v]$-s), and therefore by solving it one obtains the values of the hitting times.

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 $PTIME$. For one thing, they are computable in $EXPTIME$: 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. As

[†]Department of Applied Math and Computer Science, The Weizmann Institute, Rehovot, Israel. Incumbent of the Joseph and Celia Reskin Career Development Chair. Supported by a Yigal Alon fellowship. e-mail: feige@wisdom.weizmann.ac.il.

[‡]Department of Computer Science, Cornell University, Ithaca, NY, USA. e-mail: yuri@cs.cornell.edu.

before, $x_{u_i,V(G)} = x_{u_i,V(G)\setminus\{u_i\}} = 0$, and $x_{u_i,S} = 1 + \sum_{u_j} p_{ij} x_{u_j,S\cup\{u_i\}}$.

A simple (expected) polynomial time Monte Carlo algorithm can approximate the cover time 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 several times (the actual number of repetitions is depending on the desired precision), and output the average of the observed cover times. The expected running time of this algorithm is proportional to the actual cover time, while 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]), the above algorithm does achieve the goal. One negative feature of this algorithm (from a purely theoretical point of view) is that it does not give much information about the relations between the global structure of the graph and its cover time.

All this does not apply to arbitrary reversible Markov chains, where in the case when the ratio between the maximum and the minimum (nonzero) edge weights is super-polynomial in the number of states, the cover time can be super-polynomial in the length of description. For non-reversible Markov chains the situation is even worse, and the cover time can be exponential even if the edge weights are restricted to values 0 and $\frac{1}{2}$.

It is natural to ask whether there exists a polynomial time algorithm that approximates the cover time of Markov chains in general, and of random walks in particular. In this paper we focus on the latter question (in fact, our results extend to arbitrary reversible Markov chains).

It is our hope that research in this direction will provide new insight on the behaviour of random walks. The question also belongs to the more general context of relating the performance of randomized algorithms to that of the deterministic ones (in this case, for the task of approximating the expected value of a random variable – the cover time).

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 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 can be described as follows. From graph G one can derive a digraph G', where every pair of vertices (u, v) is connected by a (directed) edge of weight $H[u, v]$. Let $HPATH_s(G')$ be the weight of the minimum weight Hamiltonian path in G' starting at s. Then $E_s[G] \leq HPATH_s(G')$. In order to bound $HPATH_s(G')$, one replaces the digraph G' by a complete graph G'' with edge weights $C[u, v] = H[u, v] + H[v, u]$, which satisfy the triangle inequality. Finally, bounding $HPATH_s[G'']$ is achieved using the standard techniques for approximating metric TSP-s (Traveling Salesman Problem), e.g., the depth-first search order traversal of the minimum weight spanning tree MST(G''). Aleliunas *et al.* [2] then 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. Let T be an arbitrary spanning tree of G'' which is at the same time a spanning tree of G. By the above claim, it has weight $\leq 2m(n-1) \leq n^3$. But the weight of MST(G'') is at most that of T, implying $E_s[G] \leq n^3$.

In subsequent work, the existence of spanning trees of weight less than n^3 in G'' was investigated. Kahn *et al.* [13] showed that for any connected regular graph G, $MST[G''] = O(n^2)$. Coppersmith *et al.* [6] 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 [9] showed that $MST[G''] \leq 4n^3/27$ 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 $n/3$.

A different approach, suitable also for general Markov chains, was taken by Matthews in [16]. He shows that $E_s[G] \leq \max_{u,v} H[u, v] \ln n$. Clearly, this 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. By a simple argument which uses the triangle inequality for hitting times, $\max_{u,v} H[u, v]/2 < E_s[G^+] \leq \max_{u,v} H[u, v] \ln n$, and thus $\max_{u,v} H[u, v]$ provides the desired approximation.

Matthews [16] establishes also a lower bound:

$E_s[G] \geq \min_{u,v} H[u,v] \ln n$. 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 [21] in order to show that the cover time of the $\sqrt{n} \times \sqrt{n}$ mesh is $\Theta(n(\log n)^2)$, and also by Feige [10], who has established a general lower bound $E_s[G] \geq n \ln n$ (up to low order terms) for arbitrary graphs.

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.

1.2 Our results

Our main result is a deterministic polynomial time algorithm for upper bounding 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 [18], this can be accomplished using a single matrix inversion (and thus in $O(n^{3-\delta})$ time). 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.

We also show (in Section 2.6) 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

Before presenting the main result, we would like to discuss in this section a number of possible approaches for approximating the cover time, and also some results and counterexamples in this direction.

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, 20, 12]).

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 [17]. However, we do not know whether the known pseudorandom generators for space-bounded computations apply in this case. Also, keeping track of the vertices the walk visited so far appears to require a 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 $\ln n$-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 $\max_{u,v}[H[u,v]]$. The hope is, however, illusory, and the resulting approximations can be very poor.

Consider the following example. Let the graph be a clique on about 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 $\Theta(n \log n)$, establishing 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 [8]. 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 [19] 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, $H[u,v] > R[u,v]^2/2$.*

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. Consider a random walk starting at u and stopping upon hitting v. Let u_i, $i = 0,..,l-1$, be, respectively, the first vertex at distance i from u visited by the chain. Observe that $u_0 = u$, and that the chain 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 $(R[u,v] - dist[u,u_i])d_i \leq (l-i)d_i$. By linearity of expectation, the length of the random walk is at least the sum of the number of visits to $u_0,..,u_{l-1}$, which in turn is at least $\sum_{i=0}^{l-1}(l-i)d_i \geq \binom{l+1}{2}$. But $l \geq R[u,v]$, and we conclude that $H[u,v]$ is at least $R[u,v]^2/2$. □

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

$$E_v[G] \leq H[v,u] + E_u[G] \ . \qquad (1)$$

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

$$H[v,u] < 2n^2 H[u,v]/R[u,v] \ . \qquad (2)$$

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
  $$E_v[G] \leq (2\sqrt{n^3/\log n} + 1)E_u[G] \ .$$
- $R[u,v] \leq \sqrt{n \log n}$.
  Combining (1) and Lemma 2, we get
  $$E_v[G] \leq \sqrt{n^5 \log n} \ + \ E_u[G] \ .$$
  But by [10], $E_u[G] = \Omega(n \log n)$, and the statement follows.

□

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 $\Omega(n^2)$.

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, 7]. Here we would like but to 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 [7], 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 [10], where it was used for proving lower bounds on $E_v[G]$.

Define

$$HPATH_{v_i}[G] \ = \ H[v_i,v_1] \ + \ \sum_{j=1}^{i-2} H[v_j,v_{j+1}] \ +$$

$$+ \ H[v_{i-1},v_{i+1}] \ + \ \sum_{j=i+1}^{n-1} H[v_j,v_{j+1}] \ .$$

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

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

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 $HPATH_{v_i}[G]$, 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 $HPATH_{v_i}[G]$, 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 them 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 $HPATH_{v_i}[\text{path}] = \Omega(n^3)$. The ordering imposed by the difference time on the n-Path $\langle u_1, u_2, .., u_n \rangle$ is easily seen to be $\{u_1, u_n\}, \{u_2, u_{n-1}\}$, *etc.* Our main result can be viewed as a refinement of this approach.

2.4 Coupon collecting on trees

For random walks, one may order the vertices by performing a depth first search on $MST(G'')$ (see Section 1.1). An alternative way of using $MST(G'')$ was analyzed in Feige [11], where no particular order is enforced for covering the vertices of the MST. Using this approach Feige obtains upper bounds on the cover time which are sensitive to the starting point of the random walk (as we wish to do here). In particular, it is shown that for any graph, $\min_s[E_s[G]] \leq 2n^3/27$, up to low order terms.

In order for this approach to give good approximating algorithms for the cover time of general graphs, one should first be able to find good approximations of the cover time for trees. But even this is an open problem.

2.5 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 $E_s[G]$. Th answer is negative. In what follows, we describe an example in which the gap between the two numbers can be as big as $\Omega(n/\log n)$.

Consider the following graph, which we call the *key*. It consists of a path on n vertices $p_0, \ldots p_{n-1}$, where to each p_i, $i = 0, .., n/2$ there is an attached "tooth" t_i, and an n-vertex clique $h_1, \ldots, h_n$, where h_1 is connected to p_{n-1}.

Consider a random walk that starts at p_0. It is readily checked that $H[p_0, p_i] < H[p_0, h_1] = O(n^2)$, $H[p_i, t_i] = O(n^2)$, and $H(h_1, h_i] = O(n^2)$. Hence, by the triangle inequality for the hitting times, $H[p_0, v] = O(n^2)$ for any vertex v.

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 $H[h_1, t_i] = \Omega(n^3)$, we have $E_{p_0}[\text{key}] = \Omega(qn^3)$.

Claim 7 $q = \Omega(1/\log n)$.

Proof: (Sketch): 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. Let also $X = \sum_{i=0}^{n/2} X_i$ the random variable counting the number of uncovered "teeth" upon hitting h_1. Recall also the definition of the k-th moment of X, $m_k(X) = E[X^k]$. We claim that $m_0 = q$ (which is obvious), $m_1 = \Theta(1)$, $m_2 = \Theta(\log n)$.

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 the set S, is to solve the system of "harmonicity" equations satisfied by $r(v)$-s: $r(u) = 0$ for all $u \in S$, $r(v_0) = 1$, and $r(u) = \sum_v p_{uv} r(v)$ for all the rest (see [8] for a detailed discussion).

Using this method, one obtains

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

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

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

Therefore,

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

$$m_2(X) = E[X^2] \;=\; \sum_{i=0}^{n/2} E[X_i^2] + \sum_{i \neq j} E[X_i X_j] \;=$$

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

The statement now follows from the log-concavity of the sequence of moments, i.e., $m_1^2 \leq m_0 m_2$. □

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

On the other hand, this gap is never larger than $n-1$. Since $E_s[G] = E_s[\bigcup_{v\in V(G)}\{\text{covering } v\}]$,

$$E_s[G] \leq \sum_{v\in V(G)} E_s[\text{covering v}] =$$

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

2.6 Approximate inclusion exclusion

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. In order to approximate $E_s[G]$ in terms of the probabilities of $F_s[v|i]$-s, assume that we are given $Pr[\bigcap_{v\in S} F_s[v|i]]$ for any time step i and subset $S \subset V(G)$. The inclusion exclusion formula can then be used in order to compute the probability of the union $F_s[G|i]$ from the probabilities of all the intersections $\bigcap_{v\in S} F_s[v|i]$. However, since there are 2^n intersections, this alone does not yield an efficient algorithm.

Studies [15, 14] concerning how well the probability of a union 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 (explicitly given in [15]) depend only on k, n, and the size of the actual intersection involved. Hence, in our case, for a given set S, the same coefficient $\alpha^{k,n}_{|S|}$ is used for any i for the term $Pr[\bigcap_{v\in S} F_s[v|i]]$ appearing in the approximation of $F_s[G|i]$. Thus, to compute $E_s[G] = \sum_i F_s[G|i]$, we need to compute the term $\alpha^{k,n}_{|S|}\sum_i Pr[\bigcap_{v\in S} F_s[v|i]]$. Notice that $\sum_i Pr[\bigcap_{v\in S} F_s[v|i]]$ is equal to the expected time until a vertex in S is hit, and this can be computed in polynomial time. The coefficients $\alpha^{k,n}_{|S|}$ are also effectively computable. Since there are $n^{O(\sqrt{n})}$ subsets of size $O(\sqrt{n})$, this gives an $n^{O(\sqrt{n})}$ time algorithm for approximating the cover time.

If we restrict attention to polynomial time computations, computing the probabilities of polynomially many intersections give an $O(n(\log\log n)^2/(\log n)^2)$ approximation to the union (by approximating the probability of $O(n(\log\log n)^2/(\log n)^2)$ disjoint unions, each of size $(\log n)^2/(\log\log n)^2$). We do not know if there is an approach that can do better in the general inclusion exclusion scenario, but suspect that in our special case of probabilities derived from Markov chains, approximate inclusion exclusion can lead to much better approximations. This remains an area for future research.

3 The algorithm

Let G be the input graph of size n. We start by ordering the vertices of G by the difference order defined above (see 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 8 *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 3α-approximation of $E_i[G]$ can be obtained in deterministic PTIME.*

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

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

They imply that $E_i[G] \leq E_1[G] + H[i,1] \leq 3E_i[G]$. Hence, given an α-approximation of $E_1[G]$, we conclude that $\tilde{E}_1[G] + H[i,1]$ is a 3α-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 $E_1[G]$.

Next, 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 [16]) claims that the cover time of a reversible Markov chain of size n is always bounded from above by $\ln n$ times the largest hitting time. This implies in our case that for every $I \in \mathcal{P}$ and $v \in I$,

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

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 $\min\{H[i,j], H[j,i]\} = H[i,j]$.

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

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

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

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

Theorem 9 *For random walks on size-n graphs,*

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

By the way of construction of UB it is clear that it constitutes an upper bound on $E_1[G]$. Proving that it is also a lower bound (up to a polylogarithmic factor) is the content of the following section.

4 Performance guarantee

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

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

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

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

clearly $UB^* \geq cUB$ for some constant $c > \frac{1}{2}$. Let us define w_i, $i = 1, 2, .., s$, as the value of the i-th term in UB^*:

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

For $i = s$ the second term disappears.

Next, let us extend every $I_i \in \mathcal{P}$, $i = 1, 2, .., s-1$, 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 $i = 1, 2, .., s$,

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

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 $i = 1, 2, .., s-1$,

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

since $D[J_i] \geq D[I_i]$, and by definition of I_i, $D[J_i] > M[J_i]$.

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, $i = 1, .., s$, according to the largest integer power p_i of 2 such that $w_i \geq 2^{p_i}$. Since $s \leq n$, 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 $K \subset [1..s]$ 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 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 $J \in \mathcal{J}^1$, and every $h = 1,2,..,\lceil \log_2 D \rceil + 1$, define $S_h^J \subset V(G)$ as a set of vertices v such that $2^{h-1} \leq D[left(J),v] < 2^h$. Let S_0^J contain the rest of the vertices. Clearly, these sets constitute a partition of $V(G)$.

Now define a random variable Δ_J so that Δ_J is the minimal (integer) 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. If d is the (integer) value corresponding to this class, define $R = 2^d$.

An important observation is

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

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. Finally, 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, then first touch J, then at some time before the interval $[1..right(J)]$ is covered, perform an $R/2$-*digression*, i.e., reach a vertex x to the left of J such that $D[left(J),x] > R/2$. 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 it, and 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 $\mathcal{J}^3 = \langle J_1^*, J_2^*, .., J_t^* \rangle$, 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 removes all the remaining intervals in $\mathcal{J}^2$ which intersect a certain interval of the difference length R. Since the difference length $D[J]$ of any interval in $J \in \mathcal{J}^2$ is at least $\frac{W}{1+2\ln n}$, 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

$$|\mathcal{J}^3| \;\geq\; \frac{|\mathcal{J}^2|}{2 + (1 + 2\ln n)R/W} \;.$$

Let $C(k) = C_1([1..right(J_k^*)])$ 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 $J_0^* = \emptyset$, $C(0) = 0$.

In the concluding argument we shall proceed in a slightly differently fashion depending on whether $(1 + 2\ln n)R \geq W$ or not. Since $E_1[G] \geq E[C(t)]$, it suffices to show that $E[C(t)]$ is large.

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

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

and therefore

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

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

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

Case 2: $(1+2\ln n)R < W$ **or** $|\mathcal{J}^3| = 1$**.**
By Properties 5 and 6, for every $k = 1, 2, .., t$, 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

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

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

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

□

4.1 Arbitrary reversible Markov chains

A closer look on the algorithm presented in Section 3 reveals that it works in a more general setting:

Theorem 10 *For arbitrary reversible Markov chain with n states, the upper bound UB obtained in the same fashion as before, satisfies*

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

The proof of this theorem is only slightly more elaborate than that of Theorem 9. The main point that requires change is a more careful limitation of the range of values for the parameter h in Step 2. Whereas for random walks, having $h = 0, 1, 2, .., \lceil \log_2 D \rceil + 1$ allows for at most $O(\log n)$ values (since $D = O(n^3)$), this might not hold for arbitrary reversible Markov chains. 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 most probable, the analysis then proceeds as in Case 1. If the bottom set turns out to be most probable, the analysis then proceeds as in Case 2. Otherwise, proceed as before. Details are omitted from this preliminary version.

4.2 Examples

Our algorithm approximates the cover time based on the matrix of all pairs hitting times. Even if this matrix is given only approximately, in the sense that each entry deviates from its true value by a constant multiplicative factor, the algorithm and its analysis are robust enough to produce a polylogarithmic approximation of the cover time. In many cases hitting times can be approximated fairly well using general principles, such as symmetry arguments, and the following well known proposition:

Proposition 11 *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, and the robustness of our algorithm, to estimate its output UB (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 $UB = (n-1)\ln n$, which is correct (see also remark that follows).

Path: The order imposed by the difference time on the n-path $\langle u_1, u_2, .., u_n \rangle$ 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 $UB = (n-1)^2 \ln n$, which is off by a factor of $\ln n$.

However, rather than naively applying the algorithm, we 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 $\ln 2 = 1$. The reason for this strange equality is that throughout, in order to simplify notation, we use $\ln n$ really as an estimate for the Harmonic sum on $n-1$ terms (which is what Matthews' argument gives). This remark applies whenever we use the term $\ln n$. □

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 2.5) 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 2.5 that $E_{p_0}[\text{key}] = \Omega(n^3/\text{polylog}(n))$.

Acknowledgements

We thank Ran Raz and Charles Rackoff for helpful discussions.

References

[1] D. J. Aldous. "Reversible Markov chains and random walks on graphs". *Draft of first six chapters of book*, January 26, 1993.

[2] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. "Random walks, universal traversal sequences, and the complexity of maze problems". In *20th Annual Symposium on Foundations of Computer Science*, 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 Jour. 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". In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 574-586, Seattle, WA, May 1989.

[6] D. Coppersmith, U. Feige, and J. Shearer. "Random Walks on Regular and Irregular Graphs". *Technical report CS93-15*, the Weizmann Institute, Israel, 1993. To appear in *SIAM J. on Discrete Math.*

[7] 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.

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

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

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

[11] U. Feige. "Collecting Coupons on Trees, and the Analysis of Random Walks". *Technical report CS93-20*, the Weizmann Institute, Israel, 1993.

[12] R. Impagliazzo, L. Levin, M. Luby. "Pseudorandom generation from one-way functions". *21st STOC*, 1989, 12–24.

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

[14] J. Kahn, N. Linial, A. Samorodnitsky. "Inclusion-exclusion: exact and approximate". Draft, 1994.

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

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

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

[18] P. Tetali. "Design of online algorithms using hitting times". *Proc. 5th ACM-SIAM Symp. on Discrete Algorithms*, 1994, 402–411.

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

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

[21] D. Zuckerman. "A Technique for Lower Bounding the Cover Time". *SIAM J. Disc. Math., 5:81-87, 1992.*