Improved Approximation of the Minimum Cover Time

Eden Chlamtac Uriel Feige*

February 4, 2005

Abstract

Feige and Rabinovich, in [FR], gave a deterministic $O(\log^4 n)$ approximation for the time it takes a random walk to cover a given graph starting at a given vertex. This approximation algorithm was shown to work for arbitrary reversible Markov Chains. We build on the results of [FR], and show that the original algorithm gives a $O(\log^2 n)$ approximation as it is, and that it can be modified to give a $O\left(\log n(\log\log n)^2\right)$ approximation. Moreover, we show that given any $c(n)$-approximation algorithm for the maximum cover time (maximized over all initial vertices) of a reversible Markov chain, we can give a corresponding algorithm for the general cover time (of a random walk or reversible Markov chain) with approximation ratio $O(c(n) \cdot \log n)$.

1 Introduction

1.1 Random walks and Markov chains

A *random walk* on an undirected graph $G = (V, E)$ is the following process: we start at some vertex $v_0 \in V$, then choose one of its neighbors uniformly at random, then move to that neighbor, then apply the same step at the new vertex (moving to a random neighbor), and repeat ad infinitum.

A more general notion of this process is a *Markov chain*. A Markov chain is any sequence of random variables $\{X_t\}_{t=0}^{\infty}$ over some (finite) set of states S which, with respect to some stochastic *transition matrix* $P = (p_{ij})$, satisfies the *Markov property*. That is, for any arbitrary sequence of states $\{x_j\}_{s=0}^{t-1}$, we have

$$\Pr\left(X_{t+1} = j \mid X_t = i,\ X_s = x_s(0 \leq s < t)\right) = \Pr(X_{t+1} = j \mid X_t = i) = p_{ij}$$

*Department of Computer Science and Applied Mathematics, the Weizmann Institute of Science, Rehovot, Israel.

In a random walk, we have $S = V$, and $p_{ij} = 1/\deg(v_i)$.

In an irreducible Markov chain (one in which every state is connected to every other state by a path of positive probability) we have a unique *stationary distribution* on the set of states S, which is denoted by $\pi(\cdot)$. The stationarity property means that for every $i \in S$ we have $\pi(i) = \sum_{j \in S} p_{ji} \cdot \pi(j)$. Note that we will only concern ourselves with irreducible Markov chains.

A *reversible* Markov chain is a Markov chain with transition matrix $P = (p_{ij})$ for which we have $\pi(i) \cdot p_{ij} = \pi(j) \cdot p_{ji}$ for all $i, j \in S$. In fact, this is only a slightly more elaborate notion than a random walk. We can think of a reversible Markov chain as a random walk on a weighted undirected graph $G = (V, E, w)$, where every edge $e \in E$ (including self-loops) has some nonnegative weight $w(e)$. The transition matrix $P = (p_{ij})$ here is defined by

$$p_{ij} \stackrel{\text{def}}{=} \frac{w(v_i, v_j)}{\sum_{u \in V} w(v_i, u)}$$

1.2 Notation for Markov chains

Let $\{X_t\}_t$ be a Markov chain over a state set S. For any $i \in S$ we use the following notation from [AF]:

$$T_i \stackrel{\text{def}}{=} \min\{t \geq 0 \mid X_t = i\}$$

$$T_i^+ \stackrel{\text{def}}{=} \min\{t \geq 1 \mid X_t = i\}$$

We note that $T_i = T_i^+$ unless $X_0 = i$, in which case $T_i = 0$ and T_i^+ is the first return time to state i. Next, we define the *hitting time*, and related notions.

- $H(i,j) \stackrel{\text{def}}{=} \mathbb{E}[T_j \mid X_0 = i]$ is the *hitting time* from state i to state j (the expected time for a Markov chain starting at i to reach j).
- $D(i,j) \stackrel{\text{def}}{=} H(i,j) - H(j,i)$ is the *difference time* between i and j.

Finally, we have the *cover time* of a graph, or a set of states. For any $S' \subseteq S$ (where S is the state set of a Markov chain), and any $i \in S$ we define the *cover time* of S' (starting at i) as

$$C_i(S') \stackrel{\text{def}}{=} \mathbb{E}[\max_{j \in S'} T_j \mid X_0 = i]$$

which is the expected time for a Markov chain with initial state i to cover all states in S'. When discussing random walks, we will also use the notation $C_v(G)$ to indicate the cover time of all the vertices of G, which we will then simply call the "cover time of G". We extend our cover time notation to include

- $C_{max}(S') \stackrel{\text{def}}{=} \max_{i \in S'} C_i(S')$ – the *maximum* cover time of S'.

- $C_{min}(S') \stackrel{\text{def}}{=} \min_{i \in S'} C_i(S')$ – the *minimum* cover time of S'.

Note that in the literature on random walks the term "cover time" is usually used to mean the *maximum* cover time. Here we will use the term to mean the cover time of a random walk (or reversible Markov chain) with respect to a particular (sometimes unspecified) starting vertex. When we want to make the distinction clear, we will refer to this notion as the *general* cover time.

1.3 Background

From a computational complexity standpoint, hitting times are amongst a family of easily computable Markov chain parameters. The computation of these parameters is characterized by solving some set of linear equations related to the transition matrix P, which we represent, in our computational model, using rational numbers (i.e. pairs of binary integers). This operation can be performed in polynomial time in the length of the input (using Gaussian elimination, for instance).

For example, suppose for some fixed state $s \in S$, we wish to compute all the hitting times of the form $H(i,s)$. Then the variables $\{x_i\}_{i \in S}$, $x_i = H(i,s)$ are characterized by the linear equations $x_s = 0$, and $x_i = 1 + \sum_{j \in S} p_{ij} \cdot x_j$ for all $i \neq s$. This is a harmonic system of linear equations with fixed boundary conditions, and therefore has a unique solution (see, for example, [DS] for an elegant discussion). From this we can immediately compute the commute and difference times, and in a similar fashion we can also compute the first hitting time of a set of states, as well as various other parameters. We also note that Tetali [Tet1] showed how to compute all hitting times $\{H(i,j)\}_{i,j \in S}$ using a single matrix inversion (as opposed to solving n systems of linear equations, as discussed here).

While hitting times in random walks and general Markov chains pose an easy computational problem, computation of cover times has remained more elusive. Specifically, to date there is no known deterministic algorithm which approximates the maximum cover time, or specific cover times, to within a constant factor. This is somewhat peculiar in light of the existence of a very simple randomized algorithm which approximates the cover time of random walks to within any desired degree of accuracy; simply simulate the chain several times, measuring the cover time of each simulation, and output the average. Note that this approach does not work for arbitrary reversible Markov chains, where the cover time may be exponential in the number of states.

Note that there is a method to compute cover times which is analogous to the computation of hitting times. The drawback is that it is not efficient (i.e. the computation time is exponential in the number of states). The key is to construct a Markov chain (of exponential size) with hitting times corresponding to cover times of the original Markov chain.

For example, if we want to compute the cover time of the Markov chain (S, P) starting at state $s \in S$, then the state set of the new Markov chain will be $\{(i, S') \mid i, s \in S',\ S' \subseteq S\}$, and the transition probabilities $p\left((i, S'), (j, S' \cup \{j\})\right) = p_{ij}$. This Markov chain imitates the original Markov chain, while keeping track of all states covered so far. The cover time of the original Markov chain will simply be the hitting time (in the new Markov chain) from state $(s, \{s\})$ to the set of states $\{(i, S) \mid i \in S\}$.

It was long known that the maximum cover time is at most a $O(\ln n)$ factor greater than the maximum hitting time (note that it cannot be less than the maximum hitting time). Matthews [Mat] showed that the maximum cover time is at most a $\ln n$ factor greater than the maximum hitting time (as well as giving a corresponding lower bound) using the following elegant argument. Let H be the maximum hitting time of a Markov chain over state space $S = \{1, \ldots, n\}$, and suppose we wish to bound the cover time starting at state 1. Let $\sigma(\cdot)$ be a permutation on the states $\{2, \ldots, n\}$ chosen uniformly at random and independently of the Markov chain, and extend it to include $\sigma(1) = 1$. Now, consider the first time we have covered all the states $\sigma(1), \ldots, \sigma(k)$. What is the probability that up to this time we have left $\sigma(k+1)$ uncovered? Fixing any single progression of the Markov chain, this is simply the probability that $\sigma(k+1)$ is the last state discovered in the set $\{\sigma(2), \ldots, \sigma(k+1)\}$. This probability is $\frac{1}{k}$ since σ is chosen uniformly at random and independently of the Markov chain. This is still the case when we take expectation over the choice of transitions. Therefore the expected time to cover $\sigma(v_2), \ldots, \sigma(v_{k+1})$ after we already covered $\sigma(v_2), \ldots, \sigma(v_k)$ is bounded by $\frac{1}{k} \cdot H$. By linearity of expectation, the cover time is bounded by $\sum_1^{n-1} \frac{1}{k} \cdot H < \ln n \cdot H$. This argument holds for arbitrary (not necessarily reversible) irreducible Markov chains.

Following this result, the first deterministic algorithm approximating general cover times (with respect to specific initial vertices, and for arbitrary reversible Markov chains) was given by Feige and Rabinovich [FR], which gave a $O(\log^4 n)$ approximation, and which will be the primary focus here. Finally, Kahn et al. [KKLV] gave a $O\left((\log \log n)^2\right)$ approximation for the maximum cover time, which also works for reversible Markov chains.

2 The Feige-Rabinovich Algorithm

2.1 Previous results, current improvements

We shall focus here on the deterministic algorithm given by Feige and Rabinovich [FR], which approximates the expected time it takes a random walk to cover a given graph, starting at a given vertex.

The strategy used in [FR] was to order and partition the vertices of the graph into a sequence of disjoint subsets (or *intervals*), so that the choice of starting vertex will not cause the cover time of any particular interval to vary by more than some factor $O(c(n))$, and such that the time for the walk to progress from one interval to the next will also be relatively small.

For each interval, the maximum cover time is approximated, for example, using the Matthews bound. (In fact, the $c(n)$ factor mentioned above is simply the approximation ratio of the algorithm used for these local approximations.) The Feige-Rabinovich algorithm outputs the sum of these local bounds, plus the total expected time to progress from each interval to the next.

It is elementary to see why this gives an upper bound. However, the lower bound requires more in-depth analysis. [FR] used a series of sifting steps, in which they filtered out many of the intervals in the partition. The remaining intervals had comparable cover times, as well as other uniformities regarding the likely course of a random walk prior to reaching them. This facilitated the final step of the analysis, in which the remaining intervals were used in a sense as "milestones" in analyzing the behavior of a random walk as if it were progressing along the path of ordered vertices.

Two improvements are given here. The first is in the analysis of the Feige-Rabinovich algorithm. The original analysis showed a $O\left(\log^4 n\right)$ approximation ratio. Some of these $\log n$ factors were lost due to the "sifting" steps in the analysis. Here, we follow the same general lines, but replacing the case analysis in [FR] (which relied on the uniformity of remaining intervals) with a greedy algorithm which takes into account the differences between the intervals. This eliminates the need for sifting, improving the known approximation ratio of the original algorithm to $O\left(\log^2 n\right)$.

The second improvement pertains to the algorithm itself. When [FR] was first published, the best approximation available for the maximum cover time was still the Matthews bound, which gives a $\ln n$ approximation. Since then, Kahn et al. [KKLV] have shown a $O((\log\log n)^2)$ approximation, using what they call the *augmented Matthews bound*. Here we show that given any algorithm for approximating the maximum cover time, with a $c(n)$ approximation ratio, we can modify the Feige-Rabinovich algorithm to use the new algorithm as a subroutine for the local bounds, yielding a $O(c(n) \cdot \log n)$ approximation. In particular, substituting the augmented Matthews bound yields an approximation ratio of $O\left(\log n(\log\log n)^2\right)$.

We stress that our assumption here is the existence of an algorithm which approximates the maximum cover time of arbitrary reversible Markov chains, even if we only wish to approximate general cover times of a simple random walk. Though currently known (deterministic) methods for approximating the maximum cover time apply to simple random walks as well as arbitrary reversible Markov chains, it is not at all self-evident that this would be the case for any algorithm which approximates the maximum cover time for simple random walks.

The algorithm will be presented here in its more general form (i.e. using an unspecified approximation algorithm for the maximum cover time as a black box). The results of the improved analysis as pertaining to the original Feige-Rabinovich algorithm will be a special case of the more general results that follow.

2.2 Some preliminaries

Recall the notion of difference time between vertices in a random walk $D(u,v) = H(u,v) - H(v,u)$. Another important notion is the commute time, defined as $\kappa(u,v) \stackrel{\text{def}}{=} H(u,v) + H(v,u)$. Tetali's hitting time formula [Tet3] (in terms of electrical resistance) gives

$$H(u,v) = \frac{1}{2}\kappa(u,v) + \frac{1}{2}(\kappa(\pi,v) - \kappa(\pi,u))$$

where $\kappa(\pi,u) \stackrel{\text{def}}{=} \sum_i \pi(i)\kappa(i,u)$. Equivalently, this may be written as $D(u,v) = \kappa(\pi,v) - \kappa(\pi,u)$. From this equality the following two results from [CTW], [TW] immediately follow. For any three vertices u,v,w we have $D(u,w) = D(u,v) + D(v,w)$. Furthermore, the vertices of a graph can be sorted (evidently, by decreasing order of $\kappa(\pi,u)$, breaking ties arbitrarily) so that for any $u < v$ we have $D(u,v) \le 0$ (i.e. $H(u,v) \le H(v,u)$). We call this the *difference order*, and henceforth we will denote the vertices of an n-vertex graph by $\{1,2,\ldots,n\}$, according the difference order.

Next, we note that to approximate the cover time $C_v(G)$ for any vertex v, it suffices to approximate $C_1(G)$. This is because $H(v,1) + C_1(G)$ is a good approximation for $C_v(G)$, as follows from the properties of the difference order:

$$C_v(G) \le H(v,1) + C_1(G) \le H(v,1) + H(1,v) + C_v(G) \le 2H(v,1) + C_v(G) \le 3C_v(G)$$

Hence the Feige-Rabinovich algorithm concentrates on approximating $C_1(G)$. Using the same argument, we see that in fact $C_1(G)$ is at most twice the minimum cover time, so the Feige-Rabinovich algorithm may be seen as an approximation algorithm for the minimum cover time.

Finally, we note that all this holds for arbitrary reversible Markov chains (in fact, the identity $D(u,w) = D(u,v) + D(v,w)$ is an equivalent condition to reversibility in Markov chains [Tet2]). The original Feige-Rabinovich algorithm was shown in [FR] to work for arbitrary reversible Markov chains. The approximation analysis is given for the case of random walks (on simple, unweighted graphs), though arguing as in [FR], we can show that the algorithm in its general form works equally well for arbitrary reversible Markov chains. The details are provided in appendix A.

2.3 The algorithm

Suppose we have some deterministic algorithm which approximates the maximum cover time of a reversible Markov chain up to some factor $c(n)$. For any set of vertices $S \subseteq V$, let us denote the lower and upper bounds which the algorithm returns by $C_*(S)$ and $C^*(S)$, respectively. Also, we know that $C^*(S) \le c(n) \cdot C_*(S)$. For example, the original Feige-Rabinovich algorithm used the Matthews bound, which gives $C_*(S) = \max_{u,v \in S} H(u,v)$ and $C^*(S) = C_*(S) \cdot \mathrm{Ln}(|S|)$ (where $\mathrm{Ln}(k) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{k-1} \frac{1}{i} \approx \ln(k)$).

The algorithm partitions the vertices of G, arranged from left to right by the difference order, into consecutive intervals $I_1, \ldots, I_s$. For each interval I, we denote

$$D(I) \stackrel{\mathrm{def}}{=} \max_{i,j \in I} D(i,j) = D(\mathrm{right}(I), \mathrm{left}(I))$$

$$H(I) \stackrel{\mathrm{def}}{=} \max_{i,j \in I} H(i,j)$$

The algorithm, which computes an upper bound on $C_1(G)$, is as follows:

- Arrange the vertices by the difference order.
- Create the partition $I_1, \ldots, I_s$ as follows:
  - Scan vertices from left to right.
  - Increase current interval, I, as long as $D(I) \le \frac{1}{2} C_*(I)$.
  - Once current interval can no longer be extended (according to above rule), move on to next interval starting at the next uncovered vertex.
- For $i = 1, \ldots, s-1$ define

  $$w_i \stackrel{\mathrm{def}}{=} C^*(I_i) + \max_{v \in I_i} H(v, \mathrm{left}(I_{i+1}))$$

  and

  $$w_s \stackrel{\mathrm{def}}{=} C^*(I_s)$$

- Compute and output $\sum_{i=1}^{s} w_i$

To see why this is an upper bound, define the following series of random variables. For all $i < s$ let Λ_i be the time it takes a random walk starting at $\mathrm{left}(I_i)$ to cover interval I_i and then walk until $\mathrm{left}(I_{i+1})$ is reached. Let Λ_s be the time it takes a random walk starting at $\mathrm{left}(I_s)$ to cover interval I_s. We can look at $\{\Lambda_i\}_i$ as measuring mutually exclusive portions of a single random walk starting at vertex 1 which ultimately covers all vertices. Even though a random walk which covers G need not cover the intervals in this order, $\{\Lambda_i\}_i$ are always well defined, and the sum $\sum_{i=1}^{s} \Lambda_i$ is always greater than or equal to the cover time. Note that $\mathbb{E}[\Lambda_i] \leq w_i$ for all $i = 1, \ldots, s$ (by definition of w_i). Hence, by linearity of expectation, we have

$$C_1(G) \leq \mathbb{E}\left[\sum_{i=1}^{s} \Lambda_i\right] \leq \sum_{i=1}^{s} \mathbb{E}[\Lambda_i] \leq \sum_{i=1}^{s} w_i$$

Note a slight discrepancy between our assumptions regarding approximation of the maximum cover time, and the actual use thereof in this algorithm. We assume we have an algorithm which approximates the maximum cover time of a reversible Markov chain, but in practice, we use it to approximate the time it takes to cover only a *subset* of the vertices. It is not self-evident that any algorithm which approximates the maximum cover time can do so for a subset of vertices. However, with some additional work, it can be adapted to perform this task as well. Since the details are not crucial to understanding the analysis of the Feige-Rabinovich algorithm, we defer the discussion to appendix B.

For now, observe that since we know hitting and distance times to be computable in polynomial time, and assuming $C_*(\cdot)$ and $C^*(\cdot)$ are also computable in polynomial time, since we only perform $O(n)$ such operations, the Feige-Rabinovich algorithm is polynomial.

2.4 The lower bound

Here we prove our main claim regarding the general Feige-Rabinovich algorithm.

Theorem 2.1. *Given an approximation algorithm for the maximum cover time of an arbitrary reversible Markov chain with approximation ratio $c(n)$, we can give a deterministic approximation algorithm for the general cover time – starting at a specific vertex – of a random walk (or reversible Markov chain) with approximation ratio $O(c(n) \cdot \log(n))$.*

Substituting the augmented Matthews bound of [KKLV], we have our main concrete result.

Corollary 2.2. *Given an n-state reversible Markov chain with starting state s, we can deterministically approximate the expected cover time of this chain to within a $O(\log n(\log\log n)^2)$ factor.*

Let us denote the output of the algorithm by $FR(G)$. Since we've already seen the upper bound $C_1(G) \leq FR(G)$, let us proceed to proving the following lower bound:

$$C_1(G) \geq \Omega\left(\frac{1}{c(n) \cdot \log(n)}\right) FR(G)$$

We extend every I_i, for $i = 1\dots, s-1$, to include the vertex immediately following it $(\text{left}(I_{i+1}))$, and call the new interval J_i. We define $J_s = I_s$, and for the sake of uniformity of notation, we'll denote by J_0 the degenerate interval $\{1\}$. We will also denote $w(J_i) = w_i$, and call it the *weight* of J_i.

In the analysis that follows, we make the natural assumption that the maximum cover time approximation, $C^*(\cdot)$, is monotonic with respect to set inclusion. Specifically, we assume for every i, $C^*(I_i) \le C^*(J_i)$. However, there is no way to guarantee this, therefore, when this property is not guaranteed, we can change the algorithm to use $(w(J_i) =)w_i \stackrel{\text{def}}{=} \min\{C^*(I_i), C^*(J_i)\} + \max_{v\in I_i} H(v, \text{left}(I_{i+1}))$ for $i = 1,\dots,s-1$. This does not detract from the upper bound, and is sufficient to make the lower bound analysis rigorous without additional assumptions.

Note the following important observations, which also provide some intuition regarding the choice of I_i.

Claim 2.3. *For every $i = 1,\dots,s-1$, we have $D(J_i) \ge w_i/(4c(n))$.*

Proof. By definition of I_i, we have

$$
\begin{aligned}
D(J_i) > \frac{1}{2}C_*(J_i) \ge \frac{C^*(J_i)}{2c(n)}& \\
\ge \frac{C_{max}(J_i)}{2c(n)} &\ge \frac{H(v,\text{left}(I_{i+1}))}{2c(n)}
\end{aligned}
$$

for all $v \in I_i$.

Now, if $C^*(J_i) \ge C^*(I_i)$, then we are done (since $D(J_i) \ge C^*(J_i)/(2c(n))$). Otherwise we could make the proof rigorous by using the alternative definition of w_i discussed above. □

Claim 2.4. *For every $i = 1,\dots,s$, we have $C_{min}(I_i) \ge C_{max}(I_i)/4$.*

Proof. Recall our notation $H(I) \stackrel{\text{def}}{=} \max_{i,j\in I} H(i,j)$. Obviously $H(I) \le C_{max}(I)$. Consider two cases.

Case 1: $H(I) \le \frac{3}{4}C_{max}(I)$

Let $w, v \in I$ be such that $C_w(I) = C_{min}(I)$ and $C_v(I) = C_{max}(I)$. Then we have

$$C_{max}(I) \le H(v,w) + C_{min}(I) \le C_{min}(I) + H(I)$$

Hence we have

$$C_{min}(I) \ge C_{max}(I) - H(I) \ge \frac{1}{4}C_{max}(I)$$

Case 2: $H(I) > \frac{3}{4}C_{max}(I)$

Let $i, j \in I$ be such that $H(I) = H(j, i)$ (w.l.o.g. $j > i$). By definition of I we have

$$H(I) > \frac{3}{4}C_{max}(I) \geq \frac{3}{4}C_*(I) \geq \frac{3}{2}D(I) \geq \frac{3}{2}D(j, i)$$

This gives $H(i, j) > \frac{1}{3}H(j, i) \left(\geq \frac{1}{4}C_{max}(I)\right)$. But this is enough, since $C_{min}(I) \geq H(i, j)$ for any $i < j \in I$ (indeed, to cover all of I we must at some point walk from i to j, or vice-versa, and so $C_{min}(I) \geq \min\{H(i, j), H(j, i)\} = H(i, j)$).

□

Claim 2.5. *For every $i = 1, \ldots, s$, and for all $u \in J_i$, we have $C_u(J_i) \geq w_i/(5c(n))$.*

Proof. As we mentioned, for any $v, w \in J_i$, $v < w$, we have $C_u(J_i) \geq \min\{H(v, w), H(w, v)\} = H(v, w)$. In particular, $C_u(J_i) \geq \max_{v \in I_i} H(v, \mathrm{left}(I_{i+1}))$. The claim follows directly from this fact together with the following:

$$C_u(J_i) \geq C_{min}(J_i) \geq C_{min}(I_i) \geq \frac{1}{4}C_{max}(I_i) \geq \frac{1}{4}C_*(I_i) \geq \frac{C^*(I_i)}{4c(n)}$$

□

For $i = 0, \ldots, s$ we define the random variable C_i as the time it takes a random walk starting at 1 to cover all the vertices $1, \ldots, \mathrm{right}(J_i)$. In the upper bound analysis we considered a walk that covered the intervals in order from left to right. This analysis is tight if once any given interval is reached, all the intervals to its left have been covered with high probability. In such a case we could use linearity of expectation, summing over $\mathbb{E}[C_i - C_{i-1}] \geq \Omega\left(C_{min}(J_i)\right)$, together with Claim 2.5, to get the corresponding lower bound.

However, the random walk does not necessarily cover the intervals one at a time. It may be, for a given interval J_i, that when J_i is first reached, some vertex to the left of J_i is still uncovered. Let u be the leftmost such vertex, and w the first vertex reached in J_i. Then before time C_i, the walk must double back and continue until u is reached, which takes at least $H(w, u) \geq D(w, u) \geq D(\mathrm{left}(J_i), u)$ steps. Of course, u is not a fixed vertex, but if we can guarantee that there is some interval I_j ($j < i$) such that $u \in I_j$ with high probability, then we have $\mathbb{E}[C_i - C_{j-1}] \geq \Omega\left(D(\mathrm{left}(J_i), \mathrm{right}(J_j))\right)$. Using the linearity of difference times, and Claim 2.3, we can regain the weight of all intervals between J_{j+1} and J_i, discarding at most that of J_j. Then it would only be a matter of choosing intervals J_i cleverly so that we do not discard intervals with large total weight.

The problem is that u does not necessarily fall in any one interval with large probability. We have to split up the vertices left of J_i into larger segments in order to guarantee that at least one of them contains u with high probability. We do this by setting a sequence of milestones at vertices with exponentially increasing difference distance from left(J_i). This gives not only a logarithmic number of segments, but also yields the useful property that the distances from left(J_i) to adjacent milestones are only a constant factor apart, so that we do not pay too heavy a penalty for "rounding" up to these larger segments. We now formalize this intuition.

For every (extended) interval J, we define a non-empty, decreasing (by the difference order) sequence $\{v_i^J\}_i$ as follows. First, for the sake of legibility, let us denote $d_J = \max\{0, \lfloor \log_2 D(\text{left}(J), 1) \rfloor\}$. Let $v_0^J \stackrel{\text{def}}{=} \text{left}(J)$, and if $d_J > 0$, then for $i = 1, \ldots, d_J$ define $v_i^J = \max\{v \mid D(\text{left}(J), v) \geq 2^i\}$. Finally, if $v_{d_J}^J > 1$, define $v_{d_J+1}^J \stackrel{\text{def}}{=} 1$. Note that indeed the sequence is decreasing, that is, the larger i is, the further away (to the left) v_i^J is from J (and the closer it is to vertex 1).

Now define for each interval J a random variable $\rho(J)$ as follows: Start a random walk at 1, and walk until J is hit, then let $\rho(J) \stackrel{\text{def}}{=} \min\{i \mid \text{vertices } 1, \ldots, v_i^J \text{ were all covered}\}$. That is, $v_{\rho(J)}^J$ is the rightmost vertex in the sequence $\{v_i^J\}_i$ which, along with all the vertices to its left, is covered by the time J is first reached. Note that $\rho(J)$ is well defined, since $1 = \min_i\{v_i^J\}$ is always covered at the very beginning of the walk. Let $r(J)$ be the most probable value of $\rho(J)$. Since $\rho(J)$ may assume at most $2 + \log_2 D(n, 1) = O(\log n)$ values, $\rho(J) = r(J)$ with probability $\Omega(1/\log n)$. (Note here that this only holds for random walks, where hitting times – and hence difference times – are bounded by $O(n^3)$. We'll deal with arbitrary reversible Markov chains in Appendix A.)

Also, for every interval $J = J_1, \ldots, J_s$ let $\lambda(J)$ be the index corresponding to the rightmost interval containing $v_{r(J)}^J$. That is, $J_{\lambda(J)}$ is the unique interval for which left$(J_{\lambda(J)}) \leq v_{r(J)}^J <$ right$(J_{\lambda(J)})$. For brevity, we'll write $\lambda(J_i) = \lambda(i)$. Note that $1 \leq \lambda(i) \leq i$. The following observations follow directly from the definitions:

- For any $j > 0$, and any $w > v_j^{J_i}$, we have
  - $D(\text{left}(J_i), w) < 2^j$
  - $D(\text{left}(J_i), v_{j-1}^{J_i}) \geq 2^{j-1}$
- If $\rho(J_i) = r(J_i)$ then at time T_{J_i} (when J_i is first hit), we have
  - $J_0, \ldots, J_{\lambda(i)-1}$ have been covered (i.e. $T_{J_i} \geq C_{\lambda(i)-1}$).
  - If $r(J_i) > 0$, some $u \leq v_{r(J_i)-1}^{J_i}$ has not been covered.
- $\Pr(\rho(J_i) = r(J_i)) = \Omega(1/\log n)$

We can now show a lower bound on $\mathbb{E}[C_i] - \mathbb{E}[C_{\lambda(i)-1}] = \mathbb{E}[C_i - C_{\lambda(i)-1}]$. Note that by definition of $\lambda(i)$, right$(J_{\lambda(i)-1}) \leq v_{r(J_i)}^{J_i}$. Hence, in the event that $\rho(J_i) = r(J_i)$, we know that all the vertices $1, \ldots,$ right$(J_{\lambda(i)-1})$ were covered by the time J_i is first reached. Also note that even though the first vertex $v \in J_i$ reached may depend on $\rho(J_i)$, the rest of the walk (after v is reached) is independent of $\rho(J_i)$ (except for the choice of initial vertex). Hence, by Claim 2.5, we have

$$\mathbb{E}[C_i - C_{\lambda(i)-1}] \geq \Omega\left(\frac{1}{\log n}\right) \cdot \min_{v \in J_i} C_v(J_i) = \Omega\left(\frac{1}{c(n) \cdot \log n}\right) \cdot w_i \tag{1}$$

Now, assume $r = r(J_i) > 0$ (hence $\lambda(i) < i$). Recall that if $\rho(J_i) = r$, then when J_i is reached for the first time (which is after time $C_{\lambda(i)-1}$), there is still some vertex $u \leq v^J_{r-1}$ uncovered. Since $u \in \{1 \ldots, \mathrm{right}(J_i)\}$, we will have to walk back to vertex u before time C_i. This regression will take expected time at least 2^{r-1}. For $r > 1$ this is because for any $v \in J_i$, $H(v,u) \geq D(v,u) \geq D(\mathrm{left}(J_i), v^{J_i}_{r-1}) \geq 2^{r-1}$. For $r = 1$, this is trivial ($2^{r-1} = 1$, and any non-degenerate walk takes at least one step). On the other hand, note that $v^{J_i}_r < \mathrm{right}(J_{\lambda(i)})$, and hence, $D\left(\mathrm{left}(J_i), \mathrm{right}(J_{\lambda(i)})\right) < 2^r$. Combining these facts, and Claim 2.3, we get

$$\begin{aligned}
\mathbb{E}[C_i - C_{\lambda(i)-1}] \geq \Pr\left(\rho(J_i) = r(J_i)\right) \cdot 2^{r-1} &= \Omega\left(\frac{1}{\log n}\right) \cdot 2^{r-1} \\
&> \Omega\left(\frac{1}{\log n}\right) \cdot \frac{1}{2} D\left(\mathrm{left}(J_i), \mathrm{right}(J_{\lambda(i)})\right) \\
&= \Omega\left(\frac{1}{\log n}\right) \sum_{j=\lambda(i)+1}^{i-1} D(J_j) \\
&= \Omega\left(\frac{1}{c(n) \cdot \log n}\right) \sum_{j=\lambda(i)+1}^{i-1} w_j
\end{aligned} \tag{2}$$

To summarize, for every $i = 1, \ldots, s$, either $r(J_i) = 0$ (and so $\lambda(i) = i$), in which case $\mathbb{E}[C_i] - \mathbb{E}[C_{i-1}] = \Omega\left(\frac{1}{c(n) \cdot \log n}\right) \cdot w_i$ (equation (1)), or $\lambda(i) < i$, in which case, incorporating equations (1) and (2), we get $\mathbb{E}[C_i] - \mathbb{E}[C_{\lambda(i)-1}] = \Omega\left(\frac{1}{c(n) \cdot \log n}\right) \sum_{j=\lambda(i)+1}^{i} w_j$. Hence, for any sequence $0 = i_0 < i_1 < \ldots < i_t \leq s$ such that $i_j < \lambda(i_{j+1})$ for all $0 \leq j < t$, we have

$$\begin{aligned}
C_1(G) = \mathbb{E}[C_s] \geq \mathbb{E}[C_{i_t}] &= \sum_{j=1}^{t} \left(\mathbb{E}[C_{i_j}] - \mathbb{E}[C_{i_{j-1}}]\right) \\
&\geq \sum_{j=1}^{t} \left(\mathbb{E}[C_{i_j}] - \mathbb{E}[C_{\lambda(i_j)-1}]\right) \\
&\geq \Omega\left(\frac{1}{c(n) \cdot \log n}\right) \left(\sum_{j:\lambda(i_j)=i_j} w_{i_j} + \sum_{j:\lambda(i_j)<i_j} \left(\sum_{k=\lambda(i_j)+1}^{i_j} w_k \right) \right)
\end{aligned}$$

If we can find such a sequence for which $\sum_{j:\lambda(i_j)=i_j} w_{i_j} + \sum_{j:\lambda(i_j)<i_j} \left(\sum_{k=\lambda(i_j)+1}^{i_j} w_k\right) = \Omega(1) \cdot \sum_{k=1}^{s} w_k$, then the proof of Theorem 2.1 would be done. Hence it remains to solve

a problem of a purely combinatorial nature. We are given a sequence of positive weights $w_1, \ldots, w_s$, and we want to find a maximum weight subsequence which obeys the following constraints. Certain elements can be included in the subsequence without any constraints. For other values of index i, we are given some value $\lambda(i) < i$ such that including w_i in the subsequence precludes the inclusion of $w_{\lambda(i)}$. For such values of i we define the *tail* of w_i to be the subsequence $w_{\lambda(i)+1}, \ldots, w_{i-1}$. All elements in the tail can be included without additional constraints when w_i is chosen (however their own tails can not necessarily be added). The following lemma shows that there is always a legal subsequence which consists of at least $\frac{1}{4}$ of the total weight.

Lemma 2.6. *Given a sequence of weights $w_1, \ldots, w_s$ with constraints as described above, there is a greedy algorithm which finds a subsequence of weight $\geq \frac{1}{4}\sum_{i=1}^{s} w_i$ which obeys these constraints.*

Proof. The algorithm is as follows:

- Start at $i = s$ and work from right to left. Halt when $i = 0$.
- Case 1: $\lambda(i) = i$ (no constraints). Add w_i to subset and repeat with $i \leftarrow i - 1$.
- Case 2: $w_{\lambda(i)} < 2 \cdot \sum_{j=\lambda(i)+1}^{i} w_j$. Add w_i (along with its "tail") to subset, skip $w_{\lambda(i)}$, and repeat with $i \leftarrow \lambda(i) - 1$.
- Case 3: $w_{\lambda(i)} \geq 2 \cdot \sum_{j=\lambda(i)+1}^{i} w_j$. Skip w_i (add nothing to subset) and repeat with $i \leftarrow \lambda(i)$.

We'll inductively partition $w_1, \ldots, w_s$ into disjoint segments, following the run of the algorithm, and show that we retain $\frac{1}{4}$ of the weight of each segment. For case 1, there is nothing to show. We consider one element as our segment, retain that element, and move on to the next.

For case 2, our segment is $w_{\lambda(i)}, \ldots, w_i$, and we retain all the elements except for $w_{\lambda(i)}$. By our assumption for case 2, we have

$$\sum_{j=\lambda(i)+1}^{i} w_j = \frac{1}{3}\left(\sum_{j=\lambda(i)+1}^{i} w_j + 2 \cdot \sum_{j=\lambda(i)+1}^{i} w_j\right) > \frac{1}{3}\left(\sum_{j=\lambda(i)+1}^{i} w_j + w_{\lambda(i)}\right)$$

In case 3, consider as our segment all the elements skipped up until the first time we return to case 1 or 2. To be explicit, consider the sequence $l_0 < l_1 < \ldots < l_L$ where l_L is the current index i, $l_k = \lambda(l_{k+1})$ for $k = 1, \ldots, L-1$, and l_1 is the first index for which we find ourselves in case 1 or 2 (we consider $l_0 = \lambda(l_1)$ if l_1 is in case 2). The segment we consider is $w_{l_1}, \ldots, w_i$ if the sequence ends in case 1, and $w_{l_0}, \ldots, w_i$ if it ends in case 2.

Inductively (by our assumption for case 3), we see that for all $k = 1, \ldots, L-1$ we have $w_{l_k} > \sum_{j=l_k+1}^{i} w_j$. Indeed,

$$w_{l_{L-1}} = w_{\lambda(i)} \geq 2 \cdot \sum_{j=\lambda(i)+1}^{i} w_j > \sum_{j=\lambda(i)+1}^{i} w_j$$

and if (we assume inductively) $w_{l_{k+1}} > \sum_{j=l_{k+1}+1}^{i} w_j$, then

$$w_{l_k} \geq 2 \cdot \sum_{j=l_k+1}^{l_{k+1}} w_j \geq \sum_{j=l_k+1}^{l_{k+1}} w_j + w_{l_{k+1}} > \sum_{j=l_k+1}^{l_{k+1}} w_j + \sum_{j=l_{k+1}+1}^{i} w_j$$

In particular, we have

$$w_{l_1} > \sum_{j=l_1+1}^{i} w_j$$

If l_1 is in case 1, then we are done, as w_{l_1} has over half the weight of the entire segment. Otherwise, by our assumption for case 2, we have

$$\begin{aligned}
\sum_{l_0+1}^{l_1} w_j = \frac{1}{4} \cdot 2 \cdot \sum_{j=l_0+1}^{l_1} w_j + \frac{1}{2} \cdot \sum_{j=l_0+1}^{l_1} w_j &> \frac{1}{4} w_{l_0} + \frac{1}{2} \cdot \sum_{j=l_0+1}^{l_1} w_j \\
&\geq \frac{1}{4} w_{l_0} + \frac{1}{4} \cdot \sum_{j=l_0+1}^{l_1} w_j + \frac{1}{4} w_{l_1} \\
&> \frac{1}{4} \cdot \left(w_{l_0} + \sum_{j=l_0+1}^{l_1} w_j + \sum_{j=l_1+1}^{i} w_j \right)
\end{aligned}$$

Note that this analysis is tight for the above algorithm. Consider the sequence $\frac{1}{2} - \varepsilon, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots, 2^{-s}$, where $\lambda(i) = i - 1$ for all $i > 1$. The algorithm will choose $w_1 = \frac{1}{4}$, whereas the total weight is $1 - (\varepsilon + 2^{-s})$. $\square$

2.5 Cover times on subgraphs

Let us return to the technical point discussed earlier regarding the use of cover time approximation algorithms to approximate the cover time of a subset of vertices in a random walk or Markov chain. Consider a reversible Markov chain G with states $V = \{1, \ldots, n\}$ and transition matrix $P = (p_{ij})$ (recall that the reversibility condition states that $\pi(i) \cdot p_{ij} = \pi(j) \cdot p_{ji}$ for all $i, j \in V$). Suppose we want to approximate the expected time to cover some subset of states $S \subset V$ maximized over all possible initial states in S.

Some methods, such as the Matthews bound, generalize without any modification to subsets $S \subset V$. However, if our only tool is an algorithm which approximates the maximum cover time of (all the states of) a reversible Markov chain, then we need some reduction which will allow us to approximate $C_{max}(S)$ using such an algorithm. A natural approach is to find some reversible Markov chain $M = (S, Q)$ such that for any $v \in S$ we have $C_v^G(S) = C_v^M(S)(= C_v(M))$. That is, for any initial state in S, the expected time to cover all of S in a random walk in G is simply the cover time of M.

First, let us show how to define and compute such a chain. For all $i \in S$, define

$$h(i) \stackrel{\text{def}}{=} H^+(i, S) = \mathbb{E}[min\{t > 0 \mid X_t \in S\}]$$

where $\{X_t\}_t$ are the states of a random walk in G starting at i ($X_0 = i$). For all $i, j \in S$, define

$$\begin{aligned} p_{ij}^* &\stackrel{\text{def}}{=} \Pr\left(T_S^+ = T_j^+ \mid X_0 = i\right) \\ &= \Pr\left(j \text{ is first state in } S \text{ reached after starting from state } i\right) \end{aligned}$$

The hitting times $h(i)$ can be computed by solving an appropriate system of linear equations, as discussed in the introduction. The probabilities p_{ij}^* can be computed using the same method. To be explicit, for some fixed j, consider the function

$$f_j(k) \stackrel{\text{def}}{=} \Pr(T_S = T_j \mid X_0 = k)$$

This function obeys the following harmonic system of equations:

$$f_j(k) = \begin{cases} \sum_{l=1}^n p_{kl} \cdot f_j(l) & k \notin S \\ 0 & k \in S \setminus \{j\} \\ 1 & k = j \end{cases}$$

We know that there exists a unique solution, hence solving for $\{f_j(k)\}_k$ we find $p_{ij}^* = \sum_{k=1}^n p_{ik} \cdot f_j(k)$ (for all $i \in S$). Now we can define the transition matrix $Q = (q_{ij})$ for our Markov chain. For all $i, j \in S$, let

$$q_{ij} \stackrel{\text{def}}{=} \begin{cases} \left(1 - \frac{1}{h(i)}\right) + \frac{p_{ii}^*}{h(i)} & i = j \\ \frac{p_{ij}^*}{h(i)} & i \neq j \end{cases}$$

Note that this transition matrix defines a reversible Markov chain (provided the original Markov chain was also reversible). Recall that to show reversibility (say, of Q) it suffices to demonstrate that we can give every (undirected) edge $(i, j) \in S \times S$ some nonnegative weight w_{ij} such that $q_{ij} = w_{ij} / \sum_k w_{ik}$. In fact, in this case it suffices to show that for all $i, j \in S$, $i \neq j$, we have $\pi(i) \cdot p_{ij}^* = \pi(j) \cdot p_{ji}^*$ (where $\pi(\cdot)$ is the stationary distribution of $G = (V, P)$). Once this is established, it is easy to see that edge-weights $w_{ii} = \pi(i) \cdot (h(i) - 1 + p_{ii}^*)$, $w_{ij} = \pi(i) \cdot p_{ij}^*$ yield the transition matrix Q.

That $\pi(i) \cdot p^*_{ij} = \pi(j) \cdot p^*_{ji}$ follows directly from the reversibility of P. If $i = x_0 \to x_1 \to \ldots \to x_s = j$ is a path between i and j through $V \setminus S$, then the probability of this path from i to j is $\prod_{k=0}^{s-1} p_{x_k x_{k+1}}$, whereas the probability of the path in the opposite direction is $\prod_{k=1}^{s} p_{x_k x_{k-1}}$. Now simply observe that

$$\pi(i) \cdot \prod_{k=0}^{s-1} p_{x_k x_{k+1}} = \pi(x_1) \cdot p_{x_1 x_0} \cdot \prod_{k=1}^{s-1} p_{x_k x_{k+1}} = \ldots = \pi(j) \cdot \prod_{k=1}^{s} p_{x_k x_{k-1}}$$

It will be helpful to think of the transition from state i back to itself as composed of two distinct self-loops; one *delay* (with probability $1 - \frac{1}{h(i)}$), and one *step* (with probability $\frac{p^*_{ii}}{h(i)}$). Note that without the delays, the chain $M = (S, Q)$ is simply a random walk in G, observed only on S. With the delays, for every vertex i, the expected time to stop cycling in i's "delay" loop and make a step (either to i or to any other vertex in S) is $h(i)$, which is also the expected time for a walk in G to return to S after leaving i. So, in expectation, M simulates a random walk on G as seen from S. Formalizing this intuition is a purely syntactical matter.

Lemma 2.7. *Let $G = (V, P)$, $M = (S, Q)$ be two Markov chains as described and defined above. Then for any initial state $x \in S$, we have $C_x(M) = C^G_x(S)$.*

Proof. For any $i \in S$, let

$$\tau^M_x(i) \stackrel{\text{def}}{=} \mathbb{E}^M[\text{number of visits to } i \text{ before covering } S \mid X_0 = x]$$

$$\tau^G_x(i) \stackrel{\text{def}}{=} \mathbb{E}^G[\text{number of visits to } i \text{ before covering } S \mid X_0 = x]$$

and

$$\epsilon_x(i) \stackrel{\text{def}}{=} \mathbb{E}^G[\text{total length of excursions from } i \text{ back to } S \text{ before covering } S \mid X_0 = x]$$

We would like to show $\tau^M_x(i) = \epsilon_x(i)$, since clearly $C_x(M) = \sum_{i \in S} \tau^M_x(i)$, and $C^G_x(S) = \sum_{i \in S} \epsilon_x(i)$.

Recalling what was said about M, we see that in fact $\tau^M_x(i) = \tau^G_x(i) \cdot h(i)$, since $\tau^G_x(i)$ corresponds to the expected number of visits to i in M not counting "delays". We can also extend our notation for any $i, j \in S$ as follows:

$$\begin{aligned}
\tau_x(i,j) &\stackrel{\text{def}}{=} \mathbb{E}^M[\text{number of steps from } i \text{ to } j \text{ before covering } S \mid X_0 = x] \\
&= \mathbb{E}^G[\text{number of walks from } i \text{ to } j \text{ through } V \setminus S \text{ before covering } S \mid X_0 = x] \\
\epsilon_x(i,j) &\stackrel{\text{def}}{=} \mathbb{E}^G[\text{total length of walks from } i \text{ to } j \text{ through } V \setminus S \text{ before covering } S \mid X_0 = x] \\
h(i,j) &\stackrel{\text{def}}{=} \mathbb{E}^G[T^+_j \mid T^+_S = T^+_j,\ X_0 = i] \\
&= \mathbb{E}^G[\text{time to reach } j \text{ from } i \mid j \text{ is first state reached in } S]
\end{aligned}$$

We now claim that $\tau_x(i,j) = p^*_{ij} \cdot \tau^G_x(i)$. In other words, the probability of hitting j first after i is not altered by sampling only excursions in a walk which stops at the cover time. Though this may not be completely intuitive, it is a direct consequence of the Markov property. Let $\{X_t\}_t$ be the Markov chain corresponding to a random walk on G observed only on S (i.e. M without the delays), where $X_0 = x$, and let C be the (random) cover time. Then we have

$$\begin{aligned}
\tau_x(i,j) &= \sum_{t=0}^{\infty} \Pr\left(X_t = i,\ X_{t+1} = j,\ t < C\right) \\
&= \sum_{t=0}^{\infty} p^*_{ij} \cdot \Pr\left(X_t = i,\ t < C\right) \\
&= p^*_{ij} \cdot \sum_{t=0}^{\infty} \Pr\left(X_t = i,\ t < C\right) \\
&= p^*_{ij} \cdot \tau^G_x(i)
\end{aligned}$$

Having already observed that $\tau^M_x(i) = \tau^G_x(i) \cdot h(i)$, it remains to show $\epsilon_x(i) = \tau^G_x(i) \cdot h(i)$. This is now immediate, as

$$\begin{aligned}
\epsilon_x(i) &= \sum_{j \in S} \epsilon_x(i,j) \\
&= \sum_{j \in S} \tau_x(i,j) \cdot h(i,j) \\
&= \tau^G_x(i) \cdot \sum_{j \in S} p^*_{ij} \cdot h(i,j) \\
&= \tau^G_x(i) \cdot h(i)
\end{aligned}$$

□

References

[AF] D. Aldous, J.A. Fill: *Reversible Markov chains and random walks on graphs*, Draft, 1999, Available from `http://www-stat.berkley.edu/users/aldous/RWG/book.html`.

[AKLLR] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, C. Rackoff: *Random walks, universal traversal sequences, and the complexity of maze traversal*, FOCS 20, 218-233, 1979.

[CRRST] A.K. Chandra, P. Raghavan, W.L. Ruzzo, R. Smolenski, P. Tiwari: *The electrical resistance of a graph, and its applications to random walks*, Computational Complexity 6(4):312-340, 1997.

[CTW] D. Coppersmith, P. Tetali, P. Winkler: *Collisions among random walks on a graph*, SIAM J. on Discrete Mathematics, 6(3):363-374, 1993.

[DS] P.G. Doyle, J.L. Snell: *Random walk and electrical networks*, The Mathematical Association of America, 1984.

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

[FR] U. Feige, Y. Rabinovich: *Deterministic approximation of the cover time*, Random Structures and Algorithms, 23(1):1-22, 2003.

[KKLV] J.D. Kahn, J.H. Kim, L. Lovász, V.H. Vu: *The cover time, the blanket time, and the Matthews bound*, FOCS 41, 467-475, 2000.

[KLNS] J.D. Kahn, N. Linial, N. Nisan, M.E. Saks: *On the cover time of random walks on graphs*, J. Theoretical Probab., 2:121-128, 1989.

[Mat] P.C. Matthews: *Covering problems for Brownian motion on spheres*, Ann. Probability, 16:1215-1228, 1988.

[Tet1] P. Tetali: *Design of online algorithms using hitting times*, SIAM Journal on Computing 28(4):1232-1246, 1999.

[Tet2] P. Tetali: *An extension of Foster's network theorem*, Combinatorics, Probability and Computing, 3:421-427, 1994.

[Tet3] P. Tetali: *Random walks and effective resistance of networks.* J. Theoretical Probability 4(1):101-109, 1991.

[TW] P. Tetali, P. Winkler: *Simultaneous reversible Markov chains*, Combinatorics, Paul Erdős is Eighty, Vol. 1 (ed. D. Miklós, V.T. Sós, T. Szőnyi), Janos Bolyai Mathematics Institute, Budapest, 422-452, 1993.

Appendix A
Arbitrary reversible Markov chains

So far we've proven Theorem 2.1 for the case of random walks. This proof generalizes easily to arbitrary reversible Markov chains, much as in [FR]. In fact, the only fine point here is that $\rho(J)$ may no longer be confined to a logarithmic number of values. To reduce the possibilities, we cut off the sequence $\{v_i^J\}_i$, leaving only the extreme points, and the original points whose difference distance from $\mathrm{left}(J)$ is in the range $(FR(G)/n^2, FR(G))$. Strictly speaking, we take $r(J)$ to be the most probable value of $\tilde{\rho}(J)$, where

$$\tilde{\rho}(J) \stackrel{\text{def}}{=} \begin{cases} 0 & \rho(J) < \log(FR(G)) - 2\log n, \\ FR(G)+1 & \rho(J) > \log(FR(G)), \\ \rho(J) & \text{otherwise.} \end{cases}$$

Now we partition the intervals J_i into three disjoint sets.

$$\begin{aligned} \mathcal{J}_1 &\stackrel{\text{def}}{=} \{J \mid r(J) = 0\} \\ \mathcal{J}_2 &\stackrel{\text{def}}{=} \{J \mid \log(FR(G)) - 2\log n \le r(J) \le \log(FR(G))\} \\ \mathcal{J}_3 &\stackrel{\text{def}}{=} \{J \mid r(J) = FR(G)+1\} \end{aligned}$$

As the total weight of these sets is $FR(G)$, at least one of them must weigh at least $\frac{1}{3}FR(G)$ (by the *weight* of a set $\mathcal{J}$, we mean $w(\mathcal{J}) = \sum_{J\in\mathcal{J}} w(J)$).

If $w(\mathcal{J}_1) \ge \frac{1}{3}FR(G)$, then if we discard all the intervals in

$$\mathcal{J}' \stackrel{\text{def}}{=} \{J' \mid 0 < D(\mathrm{left}(J), \mathrm{left}(J')) \le FR(G)/n^2 \text{ for some } J \in \mathcal{J}_1\}$$

then using equation (1), and the corresponding analysis, we get

$$C_1(G) = \Omega\left(\frac{1}{c(n)\cdot\log n}\right) w(\mathcal{J}_1 \setminus \mathcal{J}')$$

So it suffices to show that $w(\mathcal{J}')$ is negligible (compared to $FR(G)$). But using claim 2.3, we have

$$
\begin{aligned}
w(\mathcal{J}') &\leq \sum_{J \in \mathcal{J}_1} \left(\sum_{J' : 0 < D(\text{left}(J), \text{left}(J')) \leq FR(G)/n^2} w(J') \right) \\
&\leq O(c(n)) \cdot \sum_{J \in \mathcal{J}_1} \left(\sum_{J' : 0 < D(\text{left}(J), \text{left}(J')) \leq FR(G)/n^2} D(J') \right) \\
&\leq O(c(n)) \cdot \sum_{J \in \mathcal{J}_1} \frac{FR(G)}{n^2} \\
&= O\left(\frac{c(n)}{n} \right) \cdot FR(G)
\end{aligned}
$$

If $w(\mathcal{J}_2) \geq \frac{1}{3} FR(G)$, then the analysis is identical to that of the previous section. Finally, if $w(\mathcal{J}_3) \geq \frac{1}{3} FR(G)$ (in fact, as long as $\mathcal{J}_3 \neq \emptyset$), then for any $J \in \mathcal{J}_3$, with probability $\Omega(1/\log n)$ there is a digression from J to some vertex u, for which $D(\text{left}(J), u) > FR(G)$, before G is entirely covered, hence $C_1(G) = \Omega(1/\log n) FR(G)$.