A Spanning Tree Method for Bounding Hitting Times of Random Walks on Graphs

Randy Cogill   Cheng Peng
Department of Systems and Information Engineering
University of Virginia
{rcogill, cp5pz}@virginia.edu

Abstract

In this paper we consider the problem of computing the expected hitting time to a vertex for random walks on graphs. We give a method for computing an upper bound on the expected hitting time from an arbitrary spanning tree of the graph. We illustrate this method with two examples. In these examples, we show that the bounds obtained from the spanning method are sharper than bounds obtained from other commonly used techniques.

1 Introduction

This paper is concerned with bounding expected hitting times associated with random walks on undirected graphs. Suppose $G = (V, E)$ is a connected undirected graph with $|V| = n$ and $|E| = m$. A random walk on this graph is a Markov chain, where in each time step the state of the chain is associated with a vertex in the graph. In each time step, a neighbor of the current vertex is chosen uniformly at random, and the random walk proceeds to this new vertex. The *hitting time* between s and t, denoted H_{st}, is the number of steps required by the walk to travel from vertex s to vertex t. We are concerned with the computation of $\mathbf{E}[H_{st}]$.

There is an enormous body of literature on the computation of expected hitting times for random walks on graphs. Good starting points for studying this literature are the survey paper by Lovász [8] or the monograph by Aldous [1]. Many techniques exist for computing $\mathbf{E}[H_{st}]$, ranging from solving sets of linear equations to techniques related to spectral properties of the transition matrix [1]. One of the most effective classes of analytical techniques stems from the relationship between random walks on graphs and various properties of electrical network models. The classic monograph [4] discusses these connections in detail. The connection between electrical network properties and expected hitting times is clearly laid out in [11]. There, and also in [3], it is shown that the expected *commute time* between the vertices s and t in the graph G, given by $\mathbf{E}[H_{st}] + \mathbf{E}[H_{ts}]$, is closely related to the effective resistance between the vertices s and t in an electrical network on the graph G. Specifically, if R_{st} is the effective resistance between s and t, it is shown that

$$\mathbf{E}[H_{st}] + \mathbf{E}[H_{ts}] = 2mR_{st}$$

Similar, but more complex, expressions for expected hitting times are also given in terms of effective resistances between pairs of vertices in [11].

The motivation for this paper is the fact that analytically deriving expressions for expected hitting times is often difficult. Although deep connections between hitting times and effective resistances have been shown, deriving expressions for these effective resistances is still not necessarily easy. So, we are motivated by the desire to develop simple, tractable methods for deriving bounds on expected hitting times. One way to construct bounds on expected hitting times is through bounds on effective resistances. That is, it is well known that the length of the shortest path between s and t, denoted l_{st}, is an upper bound on R_{st}. So, one possible bound is $\mathbf{E}[H_{st}] \leq 2ml_{st}$. Furthermore, we know that either $\mathbf{E}[H_{st}] \leq ml_{st}$ or $\mathbf{E}[H_{ts}] \leq ml_{st}$.

We take an alternate approach, which seeks to directly exploit the reversible nature of the random walk process. Specifically, we define a cost function on an arbitrary spanning tree of G, and show that the expected value of this cost function incurred along a random walk from s to t is equal to $\mathbf{E}[H_{st}]$. Furthermore, much of the costs incurred at each step when traveling between s and t cancel in expectation. This leads to several convenient properties that allow us to construct simple upper bounds on $\mathbf{E}[H_{st}]$.

After presenting the details of our upper bounds, we will demonstrate this technique on several examples. For each example, we will compare the upper bounds obtained with those obtained by bounding effective resistance.

2 Overview and Example

Here we will illustrate the the main ideas of this paper with a simple example. The example we consider is a random walk on the graph shown in Figure 1. This graph is known as the 'lollipop graph', and is common used as an example when discussing random walks on graphs due to its interesting extremal properties. Here we will consider a random walk that starts at the vertex marked s, and we would like to determine the expected number of steps required to reach the vertex marked t for the first time.

One way to view the hitting time H_{st} is as follows. For each edge $(i, j) \in E$, we can associate a cost $c(i, j)$. Using, for example, $c(i, j) = 1$ for all $(i, j) \in E$ gives

$$\mathbf{E}[H_{st}] = \mathbf{E}\left[\sum_{k=0}^{H_{st}} c(X_k, X_{k+1})\right].$$

However, assigning a unit cost per edge is not the only cost assignment that will produce the expected hitting time. Another (albeit not very practical) cost is obtained by assigning

$$c(i, j) = \mathbf{E}[H_{it}] - \mathbf{E}[H_{jt}]$$

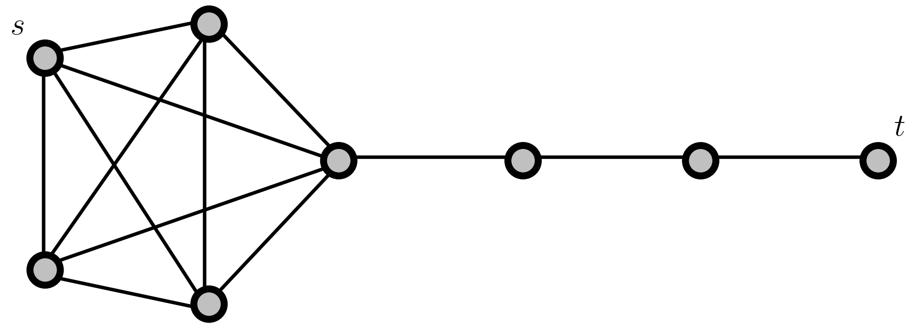

Figure 1: Lollipop graph $L_{5,3}$.

to each edge $(i,j) \in E$. Note that along any path from s to t, the total cost accumulated is $\mathbf{E}[H_{st}]$, so the expected cost accumulated is the expected hitting time from s to t. Clearly this cost assignment is not practical since edge costs are expressed in terms of expected hitting times, precisely the quantity that we want to compute.

In this paper we will consider a cost $c(i,j)$ that has attractive properties of both of the costs discussed above. That is, this cost is easily constructed for any graph, but simple relationships hold between the costs incurred along paths and the expected hitting time.

For a random walk on the graph $G = (V, E)$, the cost we use is determined as follows. First we construct a directed spanning tree of G rooted at vertex t, which we will call Γ. All edges of Γ are directed toward t, meaning that $(i,j) \in \Gamma$ if j is visited after i in the unique path from i to t. Next, for each $v \in V$ let $\mathcal{S}(v) \subseteq V$ be the set of vertices such that $w \in \mathcal{S}(v)$ if there is a path in Γ from w to v. Letting $d(v)$ denote the degree of vertex $v \in V$, we define c so that

$$c(i,j) = \begin{cases} \sum_{w \in \mathcal{S}_i} d(w) & \text{if } (i,j) \in \Gamma \\ -\sum_{w \in \mathcal{S}_j} d(w) & \text{if } (j,i) \in \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We will show that for this c

$$\mathbf{E}[H_{st}] = \mathbf{E}\left[\sum_{k=0}^{H_{st}} c(X_k, X_{k+1})\right],$$

where this expectation has the attractive property that it can be computed by taking the expectation over all *loop-free* paths through the graph. That is, we can construct a probability distribution over all paths from s to t that do not visit any vertex more than once. The expected hitting time from s to t is then the expected cost incurred over these paths. This will be illustrated next through an example.

Consider the example of the lollipop graph $L_{5,3}$ [2], shown in Figure 1. As described previously, we first construct the cost c for a directed spanning tree of the graph. Figure 1 shows one possible directed spanning tree, together with the cost associated with crossing each edge in the forward direction. The cost associated with each edge in Γ is simply the

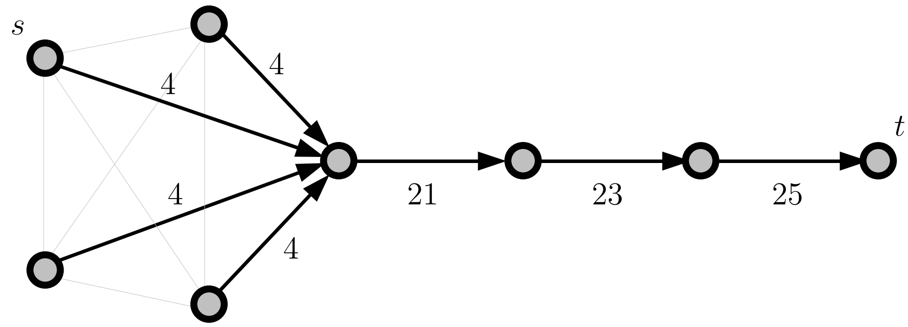

Figure 2: A directed spanning-tree cost on the lollipop graph $L_{5,3}$.

sum of the degrees of the vertices 'upstream' from this edge. The expected number of steps to travel from s to t is then given by the expected cost incurred over all loop-free paths from s to t, where the probabilities assigned to loop-free paths will be discussed shortly.

In this example, note that every loop-free path incurs the same cost. So, here we can easily compute the expected hitting time as $\mathbf{E}[H_{st}] = 73$. In fact, we can easily generalize this analysis to the lollipop graph $L_{m,n}$. Defining the directed spanning tree cost as before, we see again that every loop free path incurs the same total cost. This total cost gives the expected hitting time

$$\begin{aligned} \mathbf{E}[H_{st}] &= (m-1) + \sum_{k=0}^{n-1}(m(m-1)+1+2k) \\ &= (nm+1)(m-1)+n^2. \end{aligned}$$

The previous example has the convenient property that all loop-free paths incur the same cost. In general, each loop-free path can incur a different cost and we must compute an expected cost over all loop-free paths. We will extend this technique to provide bounds on the expected hitting time from s to t. That is, the cost associated with the most costly path is an upper bound on the expected hitting time, while the cost associated with the least costly path is a lower bound. Furthermore, we will present an even simpler upper bound obtained by adding costs associated with a subset of the edges of Γ.

3 Bounding Expected Hitting Times

The approach to bounding hitting times discussed in this section is based on bounding an expected value of the cost c given in (1). Specifically, we first show that the expected hitting time from s to t is equal to the expected value of c incurred over a random path from s to t. We then show that the expected value of c incurred along a path from s to t is equal to the expected value of c incurred over a random *loop-free* path (to be defined shortly) from s to t. Therefore, we can obtain upper bounds on the expected hitting time from s to t by

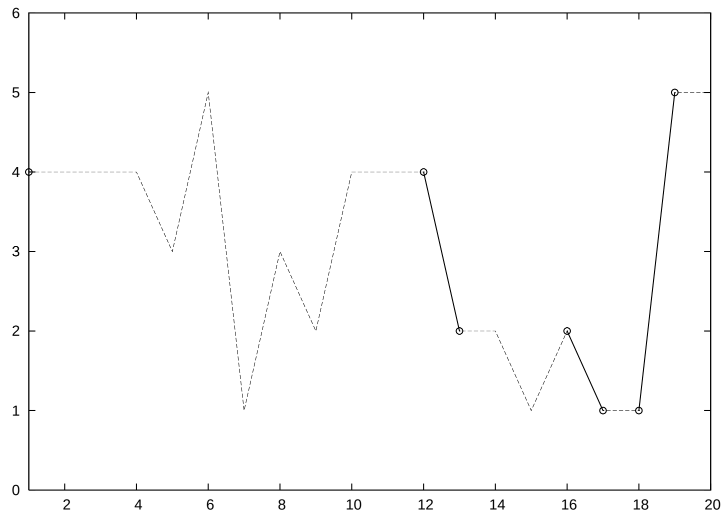

Figure 3: Construction of a loop-free path from a given sequence. In this case, we construct the loop-free path $(4, 2, 1, 5)$.

considering the maximum value of c incurred over any path from s to t, as well as related quantities.

After a random walk on G reaches vertex t, it moves to a randomly chosen neighbor of t. However, we are only concerned with the behavior of this process up to the first time t is reached. Throughout this section, we will equivalently consider the Markov chain where transitions are the same as those for the random walk on G for all $v \neq t$, but t is an absorbing state. That is, once the walk reaches t, it remains at t for all subsequent time steps. Clearly, the expected hitting time from s to t in this process is equal to the expected hitting time from s to t for the random walk on G.

A loop-free path is a finite sequence of vertices $(y_0, \ldots, y_\tau)$ such that $y_k \neq y_l$ for all $k \neq l$. For a sequence of vertices to form a loop-free path, we clearly must have $\tau \leq n$ since $\mathcal{V}$ only contains n distinct vertices. For any path $(x_0, \ldots, x_T)$ through G, we can associate a unique loop-free path $(y_0, \ldots, y_\tau)$. The construction of such a path is discussed in [9]. To construct this path, start by letting $y_0 = x_0$. Elements are then added to this path as follows. Suppose y_k is the last element that was added to the path, and let l be the largest index such that $x_l = y_k$. If $l < T$, then let $y_{k+1} = x_{l+1}$. Otherwise, let y_k be the final element of the loop-free path. We will use L to denote the operator that maps paths to their corresponding loop-free paths. An example of the construction of a loop-free path is depicted in Figure 3.

In Theorem 1, we will make use of the probability that a random loop-free path crosses an edge (i, j) when traveling between s and t. We will denote this probability by $q(i, j)$, and it is defined as follows. The probability of observing the path $(s, x_1, \ldots, x_T)$ is given by

$$\frac{1}{d(s)} \frac{1}{d(x_1)} \cdots \frac{1}{d(x_{T-1})}.$$

Let π denote some loop-free path. The probability of generating π from such a path of length

$T+1$ is

$$\mathbf{prob}_T(\pi) = \sum_{(s,\ldots,x_T):\pi=L(s,\ldots,x_T)} \frac{1}{d(s)} \cdots \frac{1}{d(x_T)}.$$

Finally, for paths of arbitrary length, the probability that a loop-free path crosses (i,j) is given by

$$q(i,j) = \lim_{T\to\infty} \sum_{\pi:(i,j)\in\pi} \mathbf{prob}_T(\pi). \tag{2}$$

Using this probability function together with the cost c we can compute the expected hitting time from s to t. The following theorem, which is the main tool used in this paper, describes this procedure for determining expected hitting times. The main purpose for introducing this theorem is to later use the fact that each $q(i,j) \in [0,1]$ to obtain a bound on the expected hitting time.

Theorem 1. *Let c be the cost defined in (1) and q be the probability function defined in (2). The expected hitting time from s to t is given by*

$$\mathbf{E}[H_{st}] = \sum_{i=1}^{n}\sum_{j=1}^{n} q(i,j)c(i,j).$$

Proof. First we will show that the expected value of the cost c incurred along a walk from s to t is equal to $\mathbf{E}[H_{st}]$. Here we are concerned with the sum of costs c up to the first hitting time to t

$$\sum_{k=0}^{H_{st}-1} c(X_k, X_{k+1}).$$

Recall that we can equivalently consider the Markov chain where transitions are the same as those for the random walk on G for all $v \neq t$, but t is an absorbing state. Since $c(t,t) = 0$, this process has

$$\sum_{k=0}^{H_{st}-1} c(X_k, X_{k+1}) = \sum_{k=0}^{\infty} c(X_k, X_{k+1}).$$

So, when considering the expected cost incurred between s and t we can consider the expectation

$$\begin{aligned}\lim_{T\to\infty} \mathbf{E}\left[\sum_{k=0}^{T-1} c(X_k, X_{k+1})\right] &= \sum_{k=0}^{\infty} \mathbf{E}\left[c(X_k, X_{k+1})\right] \\ &= \sum_{k=0}^{\infty} \mathbf{E}\big[\mathbf{E}[c(X_k, X_{k+1}) \,|\, X_k]\big].\end{aligned}$$

For all $x \neq t$, the inner expectation is

$$\begin{aligned}\mathbf{E}[c(X_k, X_{k+1}) \,|\, X_k = x] &= \sum_{y \in \mathcal{N}(x)} \frac{1}{d(x)} c(x,y) \\ &= \frac{1}{d(x)} \sum_{w \in \mathcal{S}(x)} d(w) - \frac{1}{d(x)} \sum_{y \in \mathcal{S}(x) \cap \mathcal{N}(x)} \sum_{w \in \mathcal{S}(y)} d(w) \\ &= 1.\end{aligned}$$

When we reach the target vertex t,

$$\begin{aligned}\mathbf{E}[c(X_k, X_{k+1}) \,|\, X_k = t] &= c(t,t) \\ &= 0.\end{aligned}$$

Therefore,

$$\begin{aligned}\sum_{k=0}^{\infty} \mathbf{E}\big[\mathbf{E}[c(X_k, X_{k+1}) \,|\, X_k]\big] &= \mathbf{E}\left[\sum_{k=0}^{H_{st}-1} 1\right] \\ &= \mathbf{E}[H_{st}].\end{aligned}$$

Now we will show the expected cost incurred along a random path from s to t is equal to the expected cost incurred along a random loop-free path from s to t. For any T and initial vertex x_0,

$$\mathbf{E}\left[\sum_{k=0}^{T-1} c(X_k, X_{k+1})\right] = \sum_{x_1 \in \mathcal{N}(x_0)} \cdots \sum_{x_T \in \mathcal{N}(x_{T-1})} \frac{1}{d(x_0)} \cdots \frac{1}{d(x_{T-1})} \left(\sum_{k=0}^{T-1} c(x_k, x_{k+1})\right).$$

Now we can exploit reversibility of the random walk to simplify the last sum. For a path $(x_0, \ldots, x_T)$ with $T \geq n$, there is at least one vertex that is visited at least twice. Without loss of generality, suppose vertex x_0 is visited more than once, and let k be the index of the last visit to this vertex. The cycle $(x_0, \ldots, x_k)$ and its reverse, $(x_k, x_{k-1}, \ldots, x_0)$, are equally probable and each appear as the first $k+1$ vertices of the path $(x_0, \ldots, x_T)$ with probability

$$\frac{1}{d(x_0)} \cdots \frac{1}{d(x_{k-1})}.$$

Since the cost c is skew-symmetric,

$$c(x_0, x_1) + \cdots + c(x_{k-1}, x_k) = -\big(c(x_k, x_{k-1}) + \cdots + c(x_1, x_0)\big).$$

Consider the paths $(x_0, \ldots, x_T)$ and $(x_k, \ldots, x_0, x_{k+1}, \ldots, x_T)$. Either $(x_0, \ldots, x_k) = (x_k, \ldots, x_0)$ and

$$(x_0, \ldots, x_T) = (x_k, \ldots, x_0, x_{k+1}, \ldots, x_T),$$

or $(x_0,\ldots,x_k) \neq (x_k,\ldots,x_0)$ and

$$(x_0,\ldots,x_T) \neq (x_k,\ldots,x_0,x_{k+1},\ldots,x_T).$$

In the first case, if $(x_0,\ldots,x_k) = (x_k,\ldots,x_0)$, then $c(x_0,x_1)+\cdots+c(x_{k-1},x_k)=0$ and

$$\frac{1}{d(x_0)}\cdots\frac{1}{d(x_{T-1})}\left(\sum_{l=0}^{T-1} c(x_l,x_{l+1})\right) = \frac{1}{d(x_0)}\cdots\frac{1}{d(x_{T-1})}\left(\sum_{l=k}^{T-1} c(x_l,x_{l+1})\right).$$

In the second case, if $(x_0,\ldots,x_k) \neq (x_k,\ldots,x_0)$, then

$$\begin{aligned}\frac{1}{d(x_0)}\cdots\frac{1}{d(x_{T-1})}\left(\sum_{l=0}^{T-1} c(x_l,x_{l+1})\right) + \frac{1}{d(x_0)}\cdots\frac{1}{d(x_{T-1})}\left(\sum_{l=1}^{k} c(x_l,x_{l-1}) + \sum_{l=k}^{T-1} c(x_l,x_{l+1})\right)\\ = \frac{1}{d(x_0)}\cdots\frac{1}{d(x_{T-1})}\left(2\sum_{l=k}^{T-1} c(x_l,x_{l+1})\right).\end{aligned}$$

So, the costs associated with the loop starting and ending at x_0 cancel when summing the costs incurred along the two paths weighed by the probabilities of these paths.

More generally, for any path $(x_0,\ldots,x_T)$, we can associate this with the path where all loops, as identified by the operator L, have been reversed. These paths are equiprobable and the loop costs of the first path are the negative of the loop costs of the second path. When summing the costs incurred along these paths weighed by the probabilities of these paths, the costs associated with all loops cancel. So, when computing the expected cost incurred over all paths, this is equal to the expected cost incurred over all loop-free paths. This implies that

$$\begin{aligned}\mathbf{E}\left[\sum_{k=0}^{T-1} c(X_k,X_{k+1})\right] &= \sum_{x_1\in\mathcal{N}(x_0)}\cdots\sum_{x_T\in\mathcal{N}(x_{T-1})}\frac{1}{d(x_0)}\cdots\frac{1}{d(x_{T-1})}\left(\sum_{(i,j)\in\mathbf{L}(x_0,\ldots,x_T)} c(i,j)\right)\\ &= \sum_{\pi}\mathbf{prob}_T(\pi)\left(\sum_{(i,j)\in\pi} c(i,j)\right)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sum_{\pi:(i,j)\in\pi}\mathbf{prob}_T(\pi)\right)c(i,j)\end{aligned}$$

Finally, by letting $T\to\infty$ we get the expected hitting time

$$\begin{aligned}\mathbf{E}[H_{st}] &= \lim_{T\to\infty}\mathbf{E}\left[\sum_{k=0}^{T-1} c(X_k,X_{k+1})\right]\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lim_{T\to\infty}\sum_{\pi:(i,j)\in\pi}\mathbf{prob}_T(\pi)\right)c(i,j)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} q(i,j)c(i,j).\end{aligned}$$

$\square$

In practice, determining the cost c is generally not difficult. The difficulty in applying this procedure lies in determining the probability $q(i,j)$ that a loop-free path crosses (i,j). However, we know that each $q(i,j)$ satisfies the properties $q(i,j) \geq 0$, $\sum_{j=1}^{n} q(i,j) \leq 1$, and $\sum_{i=1}^{n} q(i,j) \leq 1$. This leads to a practical method for bounding the expected hitting time from s to t.

Corollary 1. *The expected hitting time from any s to t satisfies*

$$\mathbf{E}[H_{st}] \leq \sum_{j=1}^{n} \max_i \{c(i,j)\}.$$

Proof. Since $q(i,j) \geq 0$ for all $i, j \in \{1, \ldots, n\}$ and $\sum_{i=1}^{n} q(i,j) \leq 1$ for all j,

$$\sum_{i=1}^{n} q(i,j)c(i,j) \leq \max_i \{c(i,j)\}.$$

Therefore,

$$\sum_{j=1}^{n} \sum_{i=1}^{n} q(i,j)c(i,j) \leq \sum_{j=1}^{n} \max_i \{c(i,j)\}.$$

$\square$

In the next section we will show several examples of the application of this bound. With each example, we will compare the bounds obtained with those obtained by bounding effective resistance by the length of the shortest path from s to t.

4 Examples

4.1 n-dimensional hypercube

The first example we consider is the expected hitting time associated with a random walk between opposite corners of the n-dimensional hypercube. We will denote each vertex as $(b_1, \ldots, b_n)$, where each $b_i \in \{0, 1\}$. Without loss of generality, we will use $s = (1, \ldots, 1)$ and $t = (0, \ldots, 0)$

To construct the cost c, we will use a tree constructed by taking a union of shortest paths to t from each vertex. For the n-dimensional hypercube we denote this tree as Γ_n. Specifically, for the vertex $(b_1, \ldots, b_n)$ suppose k is the largest index with $b_k = 1$. This vertex can be written as $(b_1, \ldots, b_{k-1}, 1, 0, \ldots, 0)$. The unique downstream neighbor of this vertex in this tree is $(b_1, \ldots, b_{k-1}, 0, 0, \ldots, 0)$. This is depicted in Figure 4 for the 3-dimensional cube.

For vertex t, the vertex i with $\max_i\{c(i,t)\}$ is $(1, 0, \ldots, 0)$, having $c(i,j) = n2^{n-1}$. This comes from the fact that this i has 2^{n-1} upstream vertices, each of degree n. Every vertex $j \neq t$ is of the form $(b_1, \ldots, b_{k-1}, 1, 0, \ldots, 0)$, where $k = 1$ is associated with

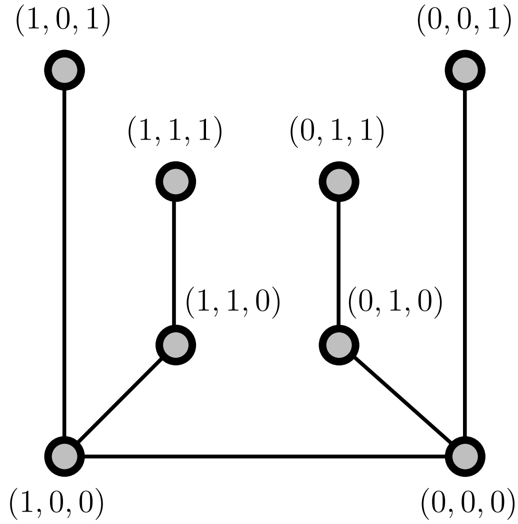

Figure 4: The spanning tree Γ_3 for the hypercube example.

the vertex $(1, 0, \ldots, 0)$. For each of these j, the upstream vertex i with $\max_i\{c(i, j)\}$ is $(b_1, \ldots, b_{k-1}, 1, 1, \ldots, 0)$, having $c(i, j) = n2^{n-k-1}$. For each $k \in \{1, \ldots, n-1\}$, there are 2^{k-1} vertices of the form $(b_1, \ldots, b_{k-1}, 1, 0, \ldots, 0)$ with upstream vertices. Hence,

$$\begin{aligned}
\mathbf{E}[H_{st}] &\leq \sum_{j=1}^{n} \max_i\{c(i, j)\} \\
&= n2^{n-1} + \sum_{k=1}^{n-1} n2^{n-k-1}2^{k-1} \\
&= \frac{1}{2}n2^n + \frac{1}{4}(n-1)n2^n \\
&= \frac{1}{4}(n+1)n2^n
\end{aligned}$$

We can compare this with bound obtained from using effective resistance methods. Note that, due to the symmetry of this graph, $\mathbf{E}[H_{st}] = \mathbf{E}[H_{ts}]$. Therefore, $\mathbf{E}[H_{st}]+\mathbf{E}[H_{ts}] \leq 2ml_{st}$ implies $\mathbf{E}[H_{st}] \leq ml_{st}$. Note that each vertex has degree n and there are 2^n vertices, so the total number of edges in this graph is $m = n2^{n-1}$. Also, the length of the path from s to t when s and t are on opposite corners is $l_{st} = n$. Hence, this produces the bound

$$\mathbf{E}[H_{st}] \leq \frac{1}{2}n^2 2^n.$$

For large n, this is nearly twice the bound obtained from the spanning tree method.

4.2 Random transpositions

The second example we consider involves computing the expected number of random pairwise transpositions required to sort a permutation of $(1,\ldots,n)$. At time step k, let $X_k = (b_1,\ldots,b_n)$ denote some permutation of $(1,\ldots,n)$. These permutations evolve randomly over time as follows. In each time step, a distinct pair of indices $(i,j) \in \{1,\ldots,n\}^2$ is chosen uniformly at random. A new permutation is then created by exchanging the positions of the elements b_i and b_j, giving

$$X_{k+1} = \{b_1,\ldots,b_{i-1},b_j,b_{i+1},\ldots,,b_{j-1},b_i,b_{j+1},\ldots,b_n\}$$

Starting from some initial permutation, we would like to compute the expected number of random transpositions required to reach the permutation $(1,\ldots,n)$.

The process described above is a random walk on a regular graph $G_n = (V_n, E_n)$ with $n!$ vertices. Each vertex in this graph has degree $d = \frac{1}{2}n(n-1)$. Although this problem can be modeled as a random walk on a graph, computing the expected hitting time is still quite difficult. However, we can apply the techniques discussed in Section 3 to compute an upper bound on the largest expected hitting time over all initial permutations s. To apply the bounds presented in Section 3, we first must construct a spanning tree on G_n.

First we will define Γ_n, the spanning tree constructed on G_n. The root of Γ_n is the permutation $t = (1,\ldots,n)$. The upstream neighbors of t are obtained from all possible transpositions of pairs of elements in $\{1,\ldots,n\}$. Let v be an arbitrary non-root vertex in Γ_n. Let i,j be the pair of elements that are transposed to bring v to its unique downstream neighbor, where $j > i$. If $j = 2$, then v is a leaf in Γ_n. Otherwise, all upstream neighbors of v are obtained from all possible transpositions of the elements $1,\ldots,j-1$. As an example, the tree Γ_3 is illustrated in Figure 5.

The tree Γ_n has the following property. From the root vertex $t = (1,\ldots,n)$, there are $n-1$ permutations directly upstream with element n in positions $1,\ldots,n-1$. Let $v_1,\ldots,v_{n-1}$ be the vertices corresponding to these permutations. For each v_k, the subtree composed of v_k and all vertices upstream from v_k is isomorphic to the tree Γ_{n-1}. This is because all permutations upstream from each v_k can be obtained by leaving the position of element n fixed, and permuting all other elements. The remaining $\frac{1}{2}(n-1)(n-2)$ permutations directly upstream from t have element n in position n. The subtree composed of these permutations together with t is also isomorphic to Γ_{n-1}. This is because these permutations can be obtained by leaving element n fixed at position n and permuting all other elements.

Now we will use this tree to compute an upper bound on $\mathbf{E}[H_{st}]$. For each edge $(i,j) \in \Gamma_n$ directed toward the root t, let $f(i,j)$ denote the total number of vertices upstream from edge (i,j). Since the graph in this example is regular, the edge cost $c(i,j)$ defined in previous sections satisfies $c(i,j) = df(i,j)$ for all edges $(i,j) \in \Gamma_n$ directed toward t. Also,

$$\begin{aligned}
\mathbf{E}[H_{st}] &\leq \sum_{j \in V_n} \max_i \{c(i,j)\} \\
&= d \sum_{j \in V_n} \max_i \{f(i,j)\}
\end{aligned}$$

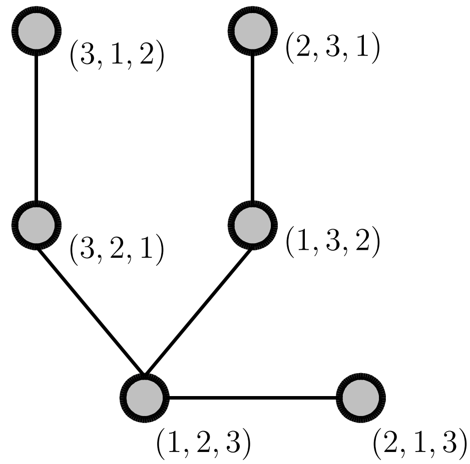

Figure 5: The spanning tree Γ_3 for the transposition example.

Recall that in Γ_n, $n-1$ of the vertices upstream from $(1, \ldots, n)$ have an upstream subtree with $(n-1)!$ vertices. The remaining vertices upstream from $(1, \ldots, n)$ are non-root vertices in a subtree isomorphic to Γ_{n-1}, and hence have an upstream subtree with fewer than $(n-1)!$ vertices. So, $\max_i\{f(i,t)\} = (n-1)!$. Let $C_n = \sum_{j \in V} \max_i\{f(i,j)\}$. Note that C_n is a sum of edge costs $f(i,j)$ composed of:

(a) The cost associated with the most costly edge entering the root of Γ_n, computed previously as $\max_i\{f(i,t)\} = (n-1)!$

(b) The total costs associated with $(n-1)$ subtrees isomorphic to Γ_{n-1}

(c) The total cost associated with one additional subtree isomorphic to Γ_{n-1}, not including the cost of the most costly edge directly entering the root from this subtree. The cost of the edge that is not included in the sum is $(n-2)!$.

Figure 6 illustrates the set of edges that are summed to compute C_4. In this figure, the edges that are highlighted are the edges with costs that are summed. Also, the labeled portions of the figure correspond to the components of the cost described in the list above. Incorporating all of the components described above, C_n satisfies the recursion

$$\begin{aligned} C_n &= (n-1)! + (n-1)C_{n-1} + (C_{n-1} - (n-2)!) \\ &= (n-1)! - (n-2)! + nC_{n-1} \end{aligned}$$

Starting with $C_2 = 1$, this recursion is easily solved to give

$$C_n = n! \left(\frac{1}{2} + \sum_{k=3}^{n} \frac{(k-1)! - (k-2)!}{k!} \right)$$

for all $n \geq 2$.

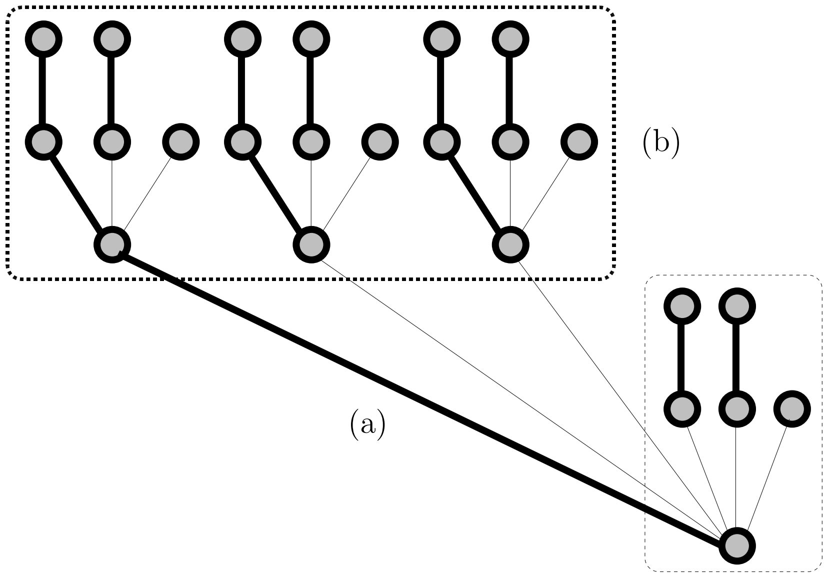

Figure 6: The spanning tree Γ_4 for the transposition example. Costs on the highlighted edges are summed when computing the upper bound.

The value of C_n then gives the upper bound

$$\begin{aligned}
\mathbf{E}[H_{st}] &\leq dC_n \\
&= \frac{1}{2}n(n-1)n!\left(\frac{1}{2}+\sum_{k=3}^{n}\frac{(k-1)!-(k-2)!}{k!}\right) \\
&= \frac{1}{2}n(n-1)n!\left(\frac{1}{2}+\sum_{k=3}^{n}\frac{k-2}{k(k-1)}\right) \\
&\leq \frac{1}{2}n(n-1)n!\ln(n)
\end{aligned}$$

Now we compare our bound with the bound obtained using effective resistance methods. In particular, we consider the expected hitting time from $s=\{n,n-1,\cdots,1\}$ to $t=\{1,2,\cdots,n\}$. Although we do not discuss the details here, one can express $\{n,n-1,\cdots,1\}$ as $\frac{n}{2}$ disjoint cycles. It can be shown that any element containing m disjoint cycles be brought to identity by using at least $n-m$ transpositions. Hence the minimum number of transpositions required to bring $\{n,n-1,\cdots,1\}$ to $\{1,2,\cdots,n\}$ is greater than or equal to $\frac{n}{2}$. Also, the number of edges in the graph G_n is $\frac{1}{2}n(n-1)n!$. Therefore, we have the bound

$$\begin{aligned}
\mathbf{E}[H_{st}] &\leq \frac{n}{2}n!\frac{n(n-1)}{2} \\
&= \frac{1}{4}n^2(n-1)n!
\end{aligned}$$

Note that the ratio between these bounds is $\frac{2\ln(n)}{n}$.

5 Conclusions

In this paper we considered the problem of bounding hitting times associated with random walks on graphs. We presented a method based on constructing costs on spanning trees of the underlying graph. These costs are chosen so that the expected cost incurred around any loop is zero. Once these costs have been constructed, upper bounds can be obtained on the largest total cost incurred over any loop-free path, effectively bounding the expected hitting time. In addition to presenting a general bounding technique, we provide two examples of the application of this technique. In both cases, the bounds produced by the spanning tree method are sharper than bounds obtained using effective resistance methods.

References

[1] D. Aldous. Reversible markov chains and random walks of graphs. In *Unpublished Monograph, Berkeley*.

[2] G. Brightwell and P. Winkler. Maximum hitting time for random walks on graphs. *RSA: Random Structures & Algorithms*, 1, 1990.

[3] A.K. Chandra, P. Raghavan, W.L. Ruzzo, R. Smolensky, and P. Tiwari. The electrical resistance of a graph captures its commute and cover times. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 574–586, 1989.

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

[5] F. Göbel and A.A. Jagers. Random walks on graphs. *Stochastic Processes and their Applications*, 2:311–336, 1974.

[6] F. Kelly. *Reversibility and Stochastic Networks*. Wiley, 1979.

[7] J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov Chains*. Van Nostrand, New Jersey, 1966.

[8] L. Lovász. Random walks on graphs: A survey. *Combinatorics: Paul Erdos Is Eighty*, 1993.

[9] P. Marchal. Loop-erased random walks, spanning trees and hamiltonian cycles. *Electronic Communications in Probability*, 5, 1999.

[10] Palacios. Expected hitting and cover times of random walks on some special graphs. *RSA: Random Structures & Algorithms*, 5, 1994.

[11] P. Tetali. Random walks and the effective resistance of networks. *Journal of Theoretical Probability*, 1:101–109, 1991.