More on Random Walks, Electrical Networks and the Harmonic k-Server Algorithm

Yair Bartal* Marek Chrobak† John Noga‡ Prabhakar Raghavan§

October 6, 2001

Abstract

Techniques from electrical network theory have been used to establish various properties of random walks. We explore this connection further, by showing how the classical formulas for the determinant and cofactors of the admittance matrix, due to Maxwell and Kirchoff, yield upper bounds on the edge stretch factor of the harmonic random walk. For any complete, n-vertex graph with distances assigned to its edges, we show the upper bound of $(n-1)^2$. If the distance function satisfies the triangle inequality, we give the upper bound of $\frac{1}{2}n(n-1)$. Both bounds are tight. As a consequence, we obtain that the harmonic algorithm for the k server problem is $\frac{1}{2}k(k+1)$-competitive against the lazy adversary.

1 Introduction

Random walks and electrical networks. Let G be a complete undirected graph with n vertices numbered $1, 2, \ldots, n$, in which each edge (u, v) is assigned its *distance* $d_{u,v} > 0$. We study the *harmonic random walk* in G, which, from a vertex u, moves to another vertex v with probability

$$p_{u,v} \quad = \quad \frac{1/d_{u,v}}{\sum_{x \neq u} 1/d_{u,x}}.$$

In the literature, the harmonic random walk was often studied using techniques from electrical network theory. Denote by $H_{u,v}$ the *hitting cost* of the harmonic walk from u to v, defined as the expected cost (total distance) to reach v for the first time when started from u. We also write $H_u = H_{u,n}$ for the hitting cost from u to n. By elementary probability, these hitting costs satisfy the following system of equations:

$$H_u \quad = \quad \sum_{v \neq u} p_{u,v}(d_{u,v} + H_v), \quad \text{for } u \neq n, \tag{1}$$

*Institute of Computer Science, The Hebrew University of Jerusalem, Jerusalem, 91904 Israel.

†Department of Computer Science, University of California, Riverside, CA 92521. Research supported by NSF grant CCR-9988360.

‡Department of Computer Science, California State University, Northridge, CA 91330. Research supported by NSF grant CCR-9988360.

§Verity, Inc. 894 Ross Drive, Sunnyvale, CA 94089.

and $H_n = 0$. As noted in [7], the hitting costs have an interpretation in terms of electrical networks. We can think of G as an electrical network in which each edge (u, v) has resistance $d_{u,v}$. If we inject current of value $n - 1$ into each node and draw current of value $n(n - 1)$ from node n, then the voltages relative to node n established at the nodes satisfy the same equation as (1). Therefore the voltage at u is equal to the hitting cost H_u.

Further connections between random walks and electrical networks were explored by Chandra *et al* [3]. One result from [3] that is directly relevant to our work is that the *cycle stretch factor* of the harmonic random walk is at most $\frac{1}{2}n(n - 1)$, where the cycle stretch factor is defined as the maximum ratio between the cost of a random walk that visits nodes of a cycle (in order) and the length of this cycle. Coppersmith *et al* [6] use the connection to electrical networks for the design of random walks and randomized online algorithms.

The *edge stretch factor* of the harmonic random walk is the the maximum ratio between the cost of a random walk between two nodes and the distance between these nodes, that is $\max_{u \neq v} H_{u,v}/d_{u,v}$. Applying the bound on the cycle stretch factor in [3] to a 2-cycle $u \to v \to u$, we get $H_{u,v} + H_{v,u} \leq \frac{1}{2}n(n - 1)(2d_{u,v})$, so the edge stretch factor is at most $n(n - 1)$ [9]. To our knowledge, this is the best upper bound known to date.

The k-server problem. In the *k-server problem* we have k servers that reside and move in a metric space M. At each time step a request $r \in M$ is specified, and we need to move one server to r to "satisfy" the request. The decision as to which server to move needs to be made *online*, without the knowledge about future requests. An online server algorithm $\mathcal{A}$ for the k-server problem is called *R-competitive* if its cost on any request sequence ϱ is at most R times the optimal (offline) cost on this sequence, plus possibly an additive constant independent of ϱ. For randomized algorithms, by $\mathcal{A}$'s cost we mean its *expected* cost on ϱ. The *competitive ratio* of $\mathcal{A}$ is the smallest R for which $\mathcal{A}$ is R-competitive. We refer the reader to [2] for more background information on this problem.

Harmonic is a randomized k-server algorithm proposed by Raghavan and Snir [9]. If the current configuration of Harmonic is X, and the request point is $r \notin X$, it moves a server from $v \in X$ with probability inversely proportional to the distances to the request, that is $p_{r,v}$. The best upper bound on the competitive ratio, by Bartal and Grove [1], is exponential in k (see also [2]), while the best lower bound is $\frac{1}{2}k(k + 1)$ [9]. For $k = 2$, it is known that Harmonic is 3-competitive [4, 5]. The $\frac{1}{2}k(k+1)$ bound on the cycle stretch factor from [3] implies that Harmonic is $\frac{1}{2}k(k+1)$-competitive in all spaces with $k+1$ points [9]. These upper bounds hold against the so-called *adaptive online adversary* that can choose requests using the knowledge of the current server locations of the algorithm. It has been conjectured that Harmonic is $\frac{1}{2}k(k+1)$-competitive in all metric spaces but, except the two cases mentioned above, this conjecture remains open.

The adversary is called *lazy* if, whenever one of his servers is not covered by our server, he requests this server's location. Otherwise, if the servers match, he moves one server to a new point. When the adversary is lazy, all $2k$ servers (ours and the adversary's) occupy at most $k + 1$ points at any given time. However, the argument that Harmonic is $\frac{1}{2}k(k + 1)$-competitive in spaces with $k + 1$ points [9] does not apply to the lazy adversary case, since it is based on the assumption that this set of $k + 1$ points is fixed throughout the computation.

Our results. We show how the classical formulas for the determinant and cofactors of the admittance matrix, due to Maxwell and Kirchoff, yield upper bounds on the edge stretch factor of the harmonic random walk. For arbitrary positive distances, we obtain an upper bound of $(n-1)^2$. If the distances satisfy the triangle inequality, that is, if G is a metric space, then we show an upper bound of $\frac{1}{2}n(n-1)$. Both upper bounds are tight. It is not difficult to see that a bound of R_n on the edge stretch factor of a harmonic random walk in n-point metric spaces implies that HARMONIC is R_{k+1}-competitive against the lazy adversary [9, 2]. As a corollary, we obtain that HARMONIC is $\frac{1}{2}k(k+1)$-competitive against the lazy adversary.

2 Upper Bound on the Stretch Factor

For convenience, we present the proof in terms of inverse distances $c_{u,v} = 1/d_{u,v}$. The triangle inequality translates then to: $c_{x,z}(c_{x,y} + c_{y,z}) \geq c_{x,y}c_{y,z}$, for all distinct nodes x, y, z. The move probability is $p_{u,v} = c_{u,v}/c_u$, where $c_u = \sum_{x \neq u} c_{u,x}$. Since $d_{u,v}$ represents the resistance of edge (u, v), we can think of $c_{u,v}$ as the *conductance* of (u, v). The system (1) can be written as

$$\begin{pmatrix} c_1 & -c_{1,2} & \cdots & -c_{1,n-1} \\ -c_{2,1} & c_2 & \cdots & -c_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{n-1,1} & -c_{n-1,2} & \cdots & c_{n-1} \end{pmatrix} \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_{n-1} \end{pmatrix} = (n-1) \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

or $\widetilde{C}H = (n-1)\mathbf{1}$. In the theory of electrical networks, $\widetilde{C}$ is called the *admittance matrix*. Let $\widetilde{C}_{i,j}$ be the sub-matrix of $\widetilde{C}$ obtained by removing row i and column j. Then

$$H_i = \frac{n-1}{\det(\widetilde{C})} \sum_{j=1}^{n-1} (-1)^{i+j} \det(\widetilde{C}_{j,i}). \tag{2}$$

A *2-tree* is a forest in G that consists of two disjoint trees. For any set of edges F, let $\xi_F = \prod_{(u,v) \in F} c_{u,v}$ be the product of all conductances in F. We use the following theorem from electrical network theory due to Maxwell and Kirchoff (see [8, 10]).

Theorem 1 (a) $\det(\widetilde{C}) = \sum_T \xi_T$, *where the sum is over all spanning trees* T.

(b) $(-1)^{i+j} \det(\widetilde{C}_{j,i}) = \sum_S \xi_S$, *where the sum is over all spanning 2-trees S in which nodes i and j are in one component and node n in the other.*

Now we are ready to prove our upper bounds on the edge stretch factor. All trees and 2-trees considered in this section are spanning subgraphs (that is, they contain all points $1, 2, \ldots, n$), so we will omit the adjective "spanning" from now on, for simplicity.

Theorem 2 (a) *The edge stretch factor of the harmonic random walk is at most* $(n-1)^2$.

(b) *If the distances satisfy the triangle inequality, then the edge stretch factor of the harmonic random walk is at most* $\frac{1}{2}n(n-1)$.

Proof: (a) We show that $H_1 \le (n-1)^2 d_{1,n}$. By (2), it is sufficient to prove that

$$c_{1,n} \sum_{j=1}^{n-1} (-1)^{j+1} \det(\widetilde{C}_{j,1}) \quad \le \quad (n-1)\det(\widetilde{C}). \tag{3}$$

By Theorem 1, part (a), the right-hand side is $(n-1)\sum_T \xi_T$. By Theorem 1, part (b), the left-hand side of (3) is a sum of terms ξ_T, for trees T that contain edge $(1,n)$. Each tree appears on the left-hand side at most $n-1$ times, and (3) follows.

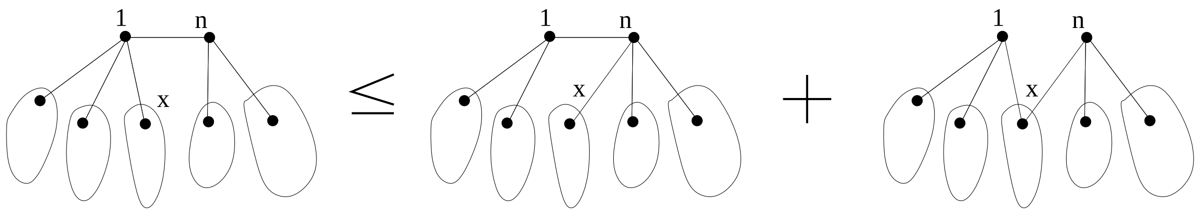
(b) Similar to (a), it is sufficient to prove that

$$2c_{1,n} \sum_{j=1}^{n-1} (-1)^{j+1} \det(\widetilde{C}_{j,1}) \quad \le \quad n\det(\widetilde{C}). \tag{4}$$

As in (a), the right-hand side of (4) is $n\sum_T \xi_T$, and the left-hand side is a sum of terms ξ_T, for trees T that contain edge $(1,n)$. Given such a tree T, T appears once in each term $c_{1,n}(-1)^{j+1}\det(\widetilde{C}_{j,1})$ where j is in the component of vertex 1 in the 2-tree $T-(1,n)$. We conclude that the left-hand side equals $2\sum_{T\ni(1,n)} r_T\xi_T$, where r_T is the size of the component of 1 in $T-(1,n)$. We call r_T the *rank of* T. Thus we reduced (4) to the following inequality:

$$2\sum_{T\ni(1,n)} r_T\xi_T \quad \le \quad n\sum_T \xi_T. \tag{5}$$

To prove (5), we will apply iteratively triangle inequalities to the tree products ξ_T that appear in the left-hand side of (5). For simplicity, in the rest of the proof we will identify ξ_T with the tree T itself and manipulate directly on trees rather than their products. We think of each such tree T as being "rooted" at edge $(1,n)$. The triangle inequalities that we apply to T are $c_{1,x}c_{1,n} \le c_{1,n}c_{x,n} + c_{1,x}c_{x,n}$, where x is a child of node 1. In terms of trees, the application of this inequality to T is equivalent to replacing T by two trees $T-(1,x)+(n,x)$ and $T-(1,n)+(n,x)$, as illustrated in the following picture:



Using this operation, we will convert the trees in the left-hand side of (5) into another collection of trees, each with multiplicity at most n. This will prove (5) and the theorem.

The procedure works as follows. We process all trees containing edge $(1,n)$ in order of decreasing rank. Suppose that T is a tree that we are currently processing. For any child x of node 1, if the subtree rooted at x has cardinality a, then replace a copies of T by a copies of $T-(1,x)+(n,x)$ and a copies of $T-(1,n)+(n,x)$.

We claim that, after we are done, the coefficient of any tree T that contains edge $(1,n)$ is n. Let $a_1,\ldots,a_l$ be the cardinalities of the subtrees of T rooted at the children of 1, and $b_1,\ldots,b_m$

the cardinalities of the subtrees rooted at the children of n. When processing T, we start with $2r_T$ copies of T. We will convert $\sum_{s=1}^{l} a_s = r_T - 1$ copies of T into trees of smaller rank. However, T can itself be obtained from converting trees of higher rank in previous steps in $\sum_{s=1}^{m} b_s = n - r_T - 1$ ways. So the number of copies of T that remain is $2r_T - (r_T - 1) + (n - r_T - 1) = n$.

In addition to trees that contain edge $(1, n)$, this process will also produce trees in which nodes 1 and n are at distance 2 apart. When we process T, as described above, we create $a \le n - 2$ copies of $T - (1, n) + (n, x)$. This tree cannot be produced in any other way during this process, so its coefficient is at most $n - 2$. This completes the proof of (b). □

Both upper bounds in Theorem 2 are tight. The lower bound of $\frac{1}{2}n(n-1)$ for metric spaces appears in [9, 2]. To show that part (a) is tight, consider the graph G with distances $d_{i,j} = 1$ for $i < j < n$, $d_{1,n} = 1$, and $d_{i,n} = D$ for $i \ne 1, n$. By elementary calculations, we get $H_{1,n} \to (n-1)^2$ for $D \to \infty$.

3 The Harmonic Algorithm

The relation between the edge stretch factor and the competitiveness of Harmonic against the lazy adversary is known [9]. Divide the computation into phases, each phase starting and ending at matched configurations, when Harmonic and the adversary occupy the same set of k points. From a matched configuration X, the adversary moves a server, say from a point v to a new point u, at cost $d_{u,v}$. Then Harmonic executes its random moves until it converges to configuration $X - v + u$. The changes of the *hole*, that is, the unique point in $X + u$ not occupied by Harmonic constitute the harmonic random walk from u to v in the $k+1$-point space $X + u$. Thus Harmonic's cost in this phase is $H_{u,v}$, which, by Theorem 2, is at most $\frac{1}{2}k(k+1)$ times the adversary cost. Summing it over all phases, we get the following theorem.

Theorem 3 *Algorithm* Harmonic *is $\frac{1}{2}k(k+1)$-competitive against the lazy adversary.*

Unplugged proof. For interested readers, we also outline a direct competitiveness proof that does not use electrical network theory. At any time step all servers cover at most $k+1$ points. Number these points $1, 2, \ldots, k+1$, with the adversary hole on $n = k+1$ and Harmonic's hole on 1. Recall that $c_i = \sum_{j \ne i} c_{i,j}$.

For $b, q \le k$ we define the numbers $H'_{b,q}$ to be the solution of

$$\sum_{l \ne b,n} c_{b,l} H'_{l,q} + k \quad = \quad (c_b + c_{b,q} - c_{b,n}) H'_{b,q} \quad \text{for all } b \ne q, \tag{6}$$

and $H'_{q,q} = 0$. Note that $H'_{b,q}$ can be interpreted as a cost of a certain random walk. This random walk takes place in an auxiliary $k+1$-vertex graph induced by the vertices $1, \ldots, k$, plus one extra copy of vertex q. Then $H'_{b,q}$ is the cost of the harmonic random walk in this graph that starts at b and ends at q or its copy. This implies that $H'_{b,q}$ is uniquely defined.

Let $\Phi = \min_q \left\{ \frac{1}{2}k(k+1)d_{q,n} + H'_{1,q} \right\}$. Denote by $\Delta\Phi$ the change of the potential in one move. We need two show that (i) when the adversary moves then $\Delta\Phi$ is at most $\frac{1}{2}k(k+1)$ times his cost, and (ii) when HARMONIC moves then its cost is at most $-\Delta\Phi$. The competitiveness follows from (i) and (ii) by amortization.

To show (i), suppose that the adversary has just moved his server from n to 1, paying $d_{1,n}$. By choosing $q = 1$ in the definition of Φ, we get $\Delta\Phi \leq \frac{1}{2}k(k+1)d_{1,n}$, and (i) follows.

To prove (ii), note that the next request is on vertex 1 and that HARMONIC's cost in this move is k/c_1. Thus it is sufficient to prove that $\Delta\Phi \leq -k/c_1$. This inequality can be derived from (6) by somewhat tedious calculations and by considering separately the cases when $q \neq 1$ and $q = 1$ in the definition of Φ. We leave the details to the reader.

References

[1] Yair Bartal and Edward Grove. The harmonic k-server algorithm is competitive. *Journal of the ACM*, 47(1):1–15, 2000.

[2] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[3] Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. In *Proc. 21st Symp. Theory of Computing*, pages 574–586, 1989.

[4] Marek Chrobak and Lawrence L. Larmore. HARMONIC is three-competitive for two servers. *Theoretical Computer Science*, 98:339–346, 1992.

[5] Marek Chrobak and Jiří Sgall. A simple analysis of the harmonic algorithm for two servers. *Information Processing Letters*, 75:75–77, 2000.

[6] Don Coppersmith, Peter G. Doyle, Prabhakar Raghavan, and Marc Snir. Random walks on weighted graphs and applications to on-line algorithms. *Journal of the ACM*, 40:421–453, 1993.

[7] Peter G. Doyle and J. Laurie Snell. *Random Walks and Electrical Networks.* Mathematical Association of America, 1984.

[8] W.S. Percival. Solution of passive electrical networks by means of mathematical trees. *J. Inst. Elect. Engrs.*, 100:143–150, 1953.

[9] Prabhakar Raghavan and Marc Snir. Memory versus randomization in online algorithms. In *Proc. 16th International Colloquium on Automata, Languages, and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 687–703. Springer, 1989.

[10] Sundaram Seshu and Myril B. Reed. *Linear Graphs and Electrical Networks.* Addison-Wesley, 1961.