The Combinatorics of Effective Resistances and Resistive Inverses

Stephen Ponzio

Institute of Computer Science, Hebrew University of Jerusalem, 91904 Jerusalem, Israel

Let matrix \((\sigma_{ij})\) denote the edge conductances of an electrical network, so that there is a resistor of \(\frac{1}{\sigma_{ij}}\) ohms between nodes \(i\) and \(j\). This uniquely determines the matrix \((R_{ij})\) of effective resistances, defined such that if a potential of 1 V is applied across nodes \(i\) and \(j\), a current of \(\frac{1}{R_{ij}}\) will flow. We call \((\sigma_{ij})\) the resistive inverse of \((R_{ij})\). One source of interest in the resistive inverse, arising in the design of on-line algorithms, is that it produces an efficient random walk if the walk must pay a cost of \(\frac{1}{\sigma_{ij}}\) for traversing edge \((i, j)\). Coppersmith et al. (1993, J. Assoc. Comput. Mach. 40(3), 421–453) showed that the random walk that makes transitions according to \((\sigma_{ij})\) is more efficient—more "competitive"—than the random walk that makes transitions according to \((R_{ij})\).

Coppersmith et al. gave a simple but obscure four-step algorithm for computing the resistive inverse. We give a complete self-contained combinatorial explanation of this algorithm, including the classical theorems of Kirchhoff and Foster.

1. RANDOM WALKS AND COMPETITIVENESS

The weights \(d_{ij}\) of an undirected graph naturally define a random walk where the next edge is chosen with probability inversely proportional to its weight: \(p_{ij} = \frac{1}{d_{ij}}\sum_k \frac{1}{d_{ik}}\). We call this the harmonic random walk. Suppose the walk incurs a cost \(d_{ij}\) when it traverses edge \((i, j)\). Then the harmonic random walk has the interesting property that the expected cost of a walk from \(u\) to \(v\) is at most \(2 |E|\) times the cost of the cheapest path from \(u\) to \(v\) ([CRRST89]; Corollary 2 below).

Alternatively, we may interpret the weights as effective resistances and instead make transitions according to the edge resistances \(1/\sigma_{ij}\); We let \(R_{ij} = \frac{d_{ij}}{\sigma_{ij}}\) and take \(p_{ij} = \frac{\sigma_{ij}}{\sum_k \sigma_{ik}}\), where \((\sigma_{ij})\) is the resistive inverse of \((R_{ij})\). We call this the resistive random walk\(^1\) and it is more efficient than harmonic: The expected cost of a walk from \(u\) to \(v\) is only \(2 |V| - 3\) times the cost of the cheapest path from \(u\) to \(v\) (Corollary 3). The proof of this uses Foster’s Theorem, \(\sum_{(i, j) \in E} R_{ij} \sigma_{ij} = n - 1\) (Corollary 1), which

\(^{1}\) Not all matrices \((R_{ij})\) have a resistive inverse \((\sigma_{ij})\). In this case we may instead use a “generalized resistive inverse”; see Section 1.3.
follows almost immediately from Kirchhoff’s Rule (Theorem 1). The latter is essential to understanding the computation of the resistive inverse.

1. Context

We shall not attempt to cite the vast literature on random walks, Markov processes, electrical network theory, and algebraic graph theory. Each result mentioned here is surely well known within certain communities, and proofs have undoubtedly appeared before in one form or another. However, the beautiful connection between the matrix tree theorem, Kirchhoff’s Rule, and effective resistances does not seem to appear in one place, concisely summarized in elementary terms. This is our contribution.

One interesting piece of related work that bears mentioning is an extension of Foster’s Theorem due to Tetali. It is well known that random walks such as we have described correspond exactly to reversible Markov chains. By interpreting Foster’s Theorem solely in terms of Markov chains, Tetali [Te94a] was able to generalize it to arbitrary (nonreversible) ergodic Markov chains. His generalization states that

$$n = \sum_{(i,j) \in E} (\pi_i H_{ij} + \pi_j H_{ji})$$

where \(\pi_i\) is the stationary probability of going from \(i\) to \(j\) and \(H_{ij}\) is the expected number of steps to go from \(j\) to \(i\). This statement is further generalized in [Te94b].

1.2. Random Walks as Competitive Strategies

The motivation for considering resistive random walks lies in the design of on-line algorithms [CDRS93, Te94b]. We first briefly explain that connection.

Consider the following “cat-and-mouse” game, played in rounds on a weighted undirected graph [AF94, CDRS93]. A round begins with the cat and the mouse occupying the same vertex. The mouse walks to another vertex in the graph, unknown to the cat, and then the cat tries to find the mouse by exploring the graph. For every edge traversed, the cat or mouse is charged the weight of that edge.

The cat would like to minimize the ratio of his cost to the cost of the mouse. A useful strategy for the cat would therefore be a random walk that is \(c\)-competitive in the sense we have already seen: for all pairs of vertices \(u\) and \(v\), the expected cost of a walk starting at \(u\) and continuing until \(v\) is at most \(c\) times the cost of the shortest path from \(u\) to \(v\).

Coppersmith et al. [CDRS93] showed that special cases of the \(k\)-server problem reduce to this cat-and-mouse game. In the \(k\)-server problem, an on-line algorithm maintains a set of \(k\) points in a metric space (the locations of \(k\) servers) and must respond to repeated requests desiring a server at some point in the space. The algorithm must decide which of its servers to move (unless one is there already), incurring a cost equal to the distance that server moves. The performance of the algorithm is measured by its competitive ratio: the maximum ratio of its cost to that of any “adversary,” where an adversary is another algorithm that solves the same problem by managing its own set of \(k\) servers but also issues the requests with full knowledge of the moves taken so far by the original algorithm.

The \(k\)-server problem is easily reduced to the cat-and-mouse game if the adversary is “lazy”: a lazy adversary moves its servers only when it must—if possible it simply
requests a point occupied by one of its servers but not by the algorithms. In this case, each move of a server by the adversary marks the beginning of a round of the cat-and-mouse game. The graph consists of the $k$ points occupied by the algorithm's servers plus the newly requested point. Since the adversary's servers occupy exactly $k$ of these $k+1$ points, we may think of a “hole” in the graph where the adversary does not have a server. Similarly, the algorithm has a hole in this graph, which moves when the algorithm chooses a server to service the request. The round finishes when the algorithm’s hole finds the adversary’s hole, indicating that their servers once again occupy the same points. Thus in the case of a lazy adversary, the resistive random walk yields a $(2(2k+1) - 3)$-competitive algorithm for the $k$-server problem. (Coppersmith et al. presented these results from a slightly different perspective; see Appendix).

However, such a reduction does not hold in general and in fact [CDRS93] gives a simple counter-example to demonstrate this. In the absence of this reduction for the general case, amortized analysis can be used instead to obtain weaker bounds. Grove [Gr91] showed that the harmonic random walk yields a competitive ratio of $O(k^2)$ in any metric space. Coppersmith et al. [CDRS93] showed that the resistive random walk gives a ratio of $k$ in the special case that the distance matrix of every $k+1$ points has a resistive inverse. This also gives a $\lambda k$-competitive algorithm if the distance matrix of every $k+1$ points in the metric space can be $\lambda$-approximated by a matrix with a resistive inverse (each entry is between 1 and $\lambda$ times the original entry). Unfortunately, it is also shown in [CDRS93] that even in the Euclidean plane, for any $\lambda$ there exists a finite set of points whose distance matrix is not $\lambda$-approximable.

We note that the $k$-server problem is solved almost definitively by a different algorithm altogether, called the “work-function algorithm”; Koutsoupias and Papadimitriou [KP94] proved that this algorithm is $2k-1$ competitive for all metric spaces. On the other hand, Manasse et al. [MMS] showed that no algorithm can achieve a ratio better than $k$.

1.3. Nonresistive Matrices

Coppersmith et al. showed that even when a given matrix $(d_{ij})$ has no resistive inverse, it is still possible to construct a “generalized resistive inverse” $(r_{ij})$: an assignment of resistances $r_{ij}$ such that either the effective resistance $R_{ij}$ equals $d_{ij}$ or $R_{ij} < d_{ij}$ and $r_{ij} = \infty$. Then a “generalized” resistive random walk with respect to $(r_{ij})$ still has a competitive ratio of $2|V| - 3$: We know that if the costs were $(R_{ij})$ then this would be the correct ratio. The costs are in fact $(d_{ij})$, but the random walk never traverses an edge whose cost is underestimated by $(R_{ij})$, so the walk’s cost is the same with respect to $(d_{ij})$ and $(R_{ij})$. On the other hand, the shortest path between two nodes is at least as great according to $(d_{ij})$ as it is according to $(R_{ij})$, so the ratio according to $(d_{ij})$ at most the ratio according to $(R_{ij})$. It is shown in [CDRS93] that such a generalized inverse always exists and is unique. They also describe an iterative algorithm whose value converges to this assignment. However, their analysis of the resistive random walk algorithm for the $k$-server problem...
2. COMPUTING THE RESISTIVE INVERSE

In this section, we restrict our attention to computing the resistive inverse in the case that one exists. We consider only connected networks. We first consider some basic properties of effective resistances and then describe a concise algorithm for the forward computation, computing the matrix of effective resistances from the matrix of conductances. This algorithm has three steps, each of which is invertible; the reverse computation is exactly the algorithm given in [CDRS93] for computing the resistive inverse.

First recall our basic definitions: By \( \sigma_{ij} = 1/r_{ij} \) we denote the conductance of edge \((i, j)\); \( \sigma_{ij} = 0 \) if edge \((i, j)\) is not present. The matrix \((\sigma_{ij})\) is finite, symmetric, non-negative, and zero on the diagonal. For each pair of nodes \(u\) and \(v\), we may compute the effective resistance \( R_{ij} \) so that if a potential of one volt is applied across nodes \(i\) and \(j\), a current of \(1/R_{ij}\) will flow. Formally, \( R_{ij} \) is equal to \((\sum_k v(k) \sigma_{kj})^{-1}\) where \(v: V \to [0, 1]\) is the unique function on the vertices that satisfies \(v(i) = 1, v(j) = 0\) and for \(k \neq i, j, \sum_k (v(k) - v(i)) \sigma_{ik} = 0\); see [DS84]. We call \((\sigma_{ij})\) the resistive inverse of \((R_{ij})\), and we call the matrix \((R_{ij})\) resistive.

2.1. Characteristics of Resistive Matrices

By physical considerations, it is clear that resistive matrices are finite, symmetric, zero on the diagonal, and positive off the diagonal. Furthermore, they satisfy the triangle inequality: \( R_{ij} \leq R_{ik} + R_{kj} \) for all \(i, j, k\). In dimension three, the triangle inequality is sufficient to ensure that a matrix is resistive, but this is not true in higher dimensions. Let us quickly examine these statements.

To see why \( R_{ij} \leq R_{ik} + R_{kj} \), we consider this statement in terms of two experiments. In the first experiment we apply a potential across nodes \(i\) and \(j\). In the second experiment we take two copies of the network and connect them with a wire (an edge of zero resistance) between the two copies of node \(k\), then apply a potential between node \(i\) in the first copy and node \(j\) in the second copy. The effective resistance of this combined network is \( R_{ik} + R_{kj} \); if a potential of 1 V is applied a current of \(1/(R_{ik} + R_{kj})\) flows. To say that \( R_{ij} < R_{ik} + R_{kj} \) is to say that the same current will flow in the first experiment using a potential of less than 1 V.

To see this, we simply superpose the resulting potentials in the first and second copies in the second experiment. This gives a solution to the system where the same current flows, entering at node \(i\) and exiting at node \(j\). The potential difference between nodes \(i\) and \(j\) is the sum of their two differences in the two copies. We know that in the first copy, node \(i\) has the highest potential and node \(k\) has the lowest, and in the second copy node \(k\) has the highest potential and node \(j\) has the lowest. Since the sum of \(v_i - v_k\) in the first copy and \(v_k - v_j\) in the second copy is 1, the sum of \(v_i - v_j\) in the first copy and \(v_i - v_j\) in the second copy is at most 1.

From this argument, we can see that \( R_{ij} = R_{ik} + R_{kj} \) exactly when \( v_j = v_k \) in the first copy and \( v_i = v_k \) in the second copy. This happens exactly when node \(k\) is a
Demonstrating the triangle inequality of effective resistances. The circuit on the left requires at most 1 V to draw the same current as the circuit on the right, so \( R_{ij} \leq R_{ik} + R_{kj} \).

cut-point in-between nodes \( i \) and \( j \), that is, when there is no path of finite resistance between \( i \) and \( j \) except through \( k \). (See [Te91] for an interesting interpretation of the gap when the triangle inequality is not tight.)

This makes it easy to see why the triangle inequality does not in general ensure that a matrix is resistive. In particular, the matrix of Euclidean distances for four points in the plane will not be resistive if exactly three are collinear. If points 1, 2, and 3 are collinear then the triangle inequality is tight among them, say \( R_{13} = R_{12} + R_{23} \), so 2 must be a cut point between 1 and 3 in the graph of resistances. Point 4 must be on either 1’s side or 3’s side of the cut, say 1’s. But this implies that \( R_{43} = R_{42} + R_{23} \), and hence point 4 must also lie on the line.

Let us now see why any \( 3 \times 3 \) matrix satisfying the triangle inequality is resistive. Suppose we wish to find the inverse of three points 1, 2, 3 with effective resistances \( A, B, \) and \( C \). First notice that we can form a “star” on four vertices, with a new vertex 4 at the center and edge resistances \( x = (A + B - C)/2 \), \( y = (A + C - B)/2 \), and \( z = (B + C - A)/2 \), so the effective resistances are the pairwise sums \( A, B, \) and \( C \). Now we use the “star-delta transformation” [Bo79, p. 31]: create a network on the three vertices 1, 2, 3 with edge resistances \( S/x, S/y, \) and \( S/z \), where \( S = xy + yz + xz \).

It is easy to check that this network has the same effective resistances between nodes as our star network (e.g., \( S/x \) in parallel with \( S/y + S/z \) yields a resistance of \( y + z = C \)) and we are done. Finally, notice that we need only that \( x, y, \) and \( z \) are positive—that is, \( A, B, \) and \( C \) satisfy the triangle inequality.

2.2. The Algorithm

In principle, \( R_{ij} \) may be computed in a straightforward fashion by solving for the potentials and currents that result in the network when 1 V is applied between nodes \( i \) and \( j \). The procedure is given in the proof of Kirchhoff’s rule (Section 3). Below, however, we give a simple but somewhat obscure algorithm for computing
1. Compute the Laplacian, $L$, of $(\sigma_{ij})$:
   \begin{align*}
   L_{ii} &= \sum_{j=1}^{n} \sigma_{ij}; \\
   L_{ij} &= -\sigma_{ij};
   \end{align*}
   and let $Q$ be the principal submatrix of $L$ obtained by deleting the last row and column.

2. Compute $D = Q^{-1}$.

3. Construct $(R_{ij})$ of dimension $n$ as follows:
   \begin{align*}
   R_{ii} &= 0 \quad (1 \leq i \leq n); \\
   R_{im} &= R_{mi} = D_{ii} \quad (1 \leq i \leq n); \\
   R_{ij} &= D_{ii} + D_{jj} - 2D_{ij} = R_{in} + R_{nj} - 2D_{ij}, \quad (1 \leq i, j \leq n - 1).
   \end{align*}

We will prove that this algorithm is correct, which implies that the effective resistances are unique, a physically obvious fact. When run in the reverse direction, the algorithm will fail if (and only if) the input matrix $(R_{ij})$ has no resistive inverse. It can fail either because $D$ is not invertible or because the output $(\sigma_{ij})$ contains negative entries.

The matrix tree theorem says that if $L$ is the Laplacian of the adjacency matrix then the number of spanning trees in the graph is equal to the determinant of any $(n-1) \times (n-1)$ submatrix (such as $Q$) of $L$. In the same way, if we define the weight of a tree $T$ (or any set of edges) to be $w(T) = \prod_{(x,y) \in T} \sigma_{xy}$, we obtain the following claim, which we prove in Section 4.

**Claim 1.**

\[
\det Q = \sum_{\text{trees } T} w(T) \stackrel{\text{def}}{=} A.
\]

Also in Section 4, we will prove that for a submatrix $Q(s, t)$ of $Q$ obtained by deleting row $s$ and column $t$,

**Claim 2.**

\[
\det Q(s, t) = (-1)^{r+s} \cdot \frac{1}{2} (A_{ss} + A_{tt} - A_{st}),
\]

where $A_{ij}$ is the sum of $w(F)$ over acyclic sets ("forests") $F$ of $n-2$ edges such that $F \cup \{(i, j)\}$ forms a tree. The algorithm is now explained using Kirchhoff's rule, which we prove in Section 3:

**Theorem 1 (Kirchhoff's rule).**

\[
R_{ij} = A_{ij}/A.
\]
We can now verify the correctness of the algorithm by calculating $D_{ij}$ according to Cramer’s rule:

$$D_{ij} = D_{ji} = (-1)^{i+j} \frac{\det Q(i,j)}{\det Q} = \frac{1}{2} \left( A_m + A_{xy} - A_y \right) = \frac{1}{2} (R_m + R_{xy} - R_y),$$

which is in agreement with our algorithm (recalling that $R_{ii} = 0$ and $R_{ij} = R_{ji}$).

As a corollary to Kirchhoff’s rule, we immediately obtain Foster’s theorem:

**Corollary 1 (Foster’s Theorem).**

$$(x, y) \in E$$

$$R_{xy} = \sum_{(x, y) \in F} v_{x}$$

$$= n - 1,$$

since in the numerator, the product of each spanning tree occurs once for each edge it contains.

### 3. KIRCHHOFF’S RULE

First recall our definitions:

- $w(T) \overset{\text{def}}{=} \prod_{(x, y) \in T} \sigma_{xy}$, for any set of edges $T$;
- $A \overset{\text{def}}{=} \sum_{\text{trees} T} w(T)$;
- $E(i, j)$ denotes the set of all forests (acyclic subsets of edges) $F$ such that $F \cup \{i, j\}$ is a tree;
- $A_y \overset{\text{def}}{=} \sum_{F \in E(i, j)} v(F)$.

**Theorem 1 (Kirchhoff’s rule).**

$$R_y = A_y / A.$$

We can quickly verify this expression in the case that the graph is itself a tree: If $p$ is the path from $i$ to $j$ then the formula correctly gives $R_y = \sum_{(a, b) \in p} 1/\sigma_{ab}$.

**Proof.** Let $n$ and $m$ be the number of nodes and edges, respectively, in the underlying graph. It is natural here to assume that all weights $\sigma_{xy}$ are positive (edge $(i, j)$ is absent if $\sigma_{ij} = 0$). We assign each edge $e$ an arbitrary but fixed orientation $e = (i, j)$. Let us first prove the theorem for the case when $(i, j) \in E$.

We will compute $R_y$ by solving the network equations when a potential difference of one volt is maintained between nodes $i$ and $j$. Introduce $n - 1$ variables $v_x$ for the voltage at each node $x \neq n$, defining the voltage at $n$ to be 0. Introduce also $m$ variables $i_e$ for the directed current through each edge $e$ (so if $e = (i, j)$ then current $i_e$ flows from $i$ to $j$ and current $-i_e$ flows from $j$ to $i$). In general the system of linear equations

$$\begin{bmatrix} 0 & G \\ G^T & R \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} = \begin{bmatrix} J_m \\ E_m \end{bmatrix}$$
expresses Kirchhoff’s law of conservation of current at the nodes and Kirchhoff’s voltage law across each edge, where

- \( V \) is the vector of node voltages \( v_x \);
- \( I \) is the vector of edge currents \( i_e \);
- \( J_m \) is the vector of currents supplied at the nodes by external sources;
- \( E_m \) is the vector of potential differences supplied (in series) by external sources on each edge;
- \( G \) is the incidence matrix of the graph with node \( n \) deleted: it is an \((n - 1) \times m\) matrix containing, for each edge \( e = (i, j) \), a column that has a 1 in row \( i \) and a \(-1\) in row \( j \) and 0s elsewhere;
- \( R \) is an \( m \times m \) diagonal matrix with \( R_e = r_e = 1/\sigma_e \).

If the only external source is a potential difference of 1 V inserted (in series) into a single edge, then the system of equations has the following form:

\[
\begin{bmatrix}
0 & 0 & 0 & -1 & \\
1 & 0 & 0 & 1 & \\
-1 & -1 & 1 & 0 & \\
0 & 0 & -1 & 0 & \\
\end{bmatrix}
\begin{bmatrix}
V_1 \\
V_2 \\
V_3 \\
V_m \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

This system can be solved for \( V \) and \( I \), the voltages at each node and the current through each edge. We will solve for \( I_e \), the current drawn through edge \( e \) when a 1-V source is added in series on edge \( e \). The resistance \( S_e = 1/I_e \) is equal to the edge resistance \( r_e \) in series with the rest of the network. The effective resistance \( R_e \) is equal to \( r_e \) in parallel with the resistance of the rest of the network. Thus, \( R_e \) is equal to \( r_e \) in parallel with \( S_e - r_e \):

\[
R_e = \frac{r_e(S_e - r_e)}{r_e + (S_e - r_e)} = r_e(1 - r_e I_e).
\]

We will show that

\[
I_y = \frac{\sum_{T \in S_{(x,y)}} ((1/r_y) \prod_{(x,y) \in \bar{T}} r_{xy})}{\sum_T (\prod_{(x,y) \in \bar{T}} r_{xy})} = \frac{\sum_{T \in S_{(x,y)}} (1/r_y) \cdot w(T)}{\sum_T w(T)}.
\]

(1)
It then follows that
\[
R_{ij} = r_{ij} \cdot \left[ 1 - r_{ij} \frac{\sum_{T} w(T) \cdot (1/r_{ij})}{\sum_{T} w(T)} \right] = r_{ij} \cdot \frac{\sum_{T} w(T) \cdot w(T)}{\sum_{T} w(T)} = \frac{A_{ij}}{A}
\]
and the case of \((i, j) \notin E\) will be proved.

For the case \((i, j) \notin E\), simply notice that the calculation holds algebraically with \(\sigma_{ij}\) as a variable and our final expression for \(R_{ij}\) is perfectly well defined for \(\sigma_{ij} = 0\):

It only remains to show Eq. (1). Let \(M\) denote the matrix above comprised of \(0, G, G^T, \) and \(R\). For edge \(e\), the system of linear equations yields \(I_{e}\) equal to the \((n-1+e)\)th diagonal element of \(M^{-1}\), which is equal to \(det \ M(e, e)/det \ M\), where \(M(e, e)\) is the submatrix obtained by deleting the row and column containing \(r_e\).

We will show that \(det \ M(e, e)\) and \(det \ M\) correspond to the numerator and denominator of Eq. (1).

We have
\[
det \ M = \sum_{\pi} \text{sgn}(\pi) \prod_{i=1}^{m+n-1} M_{i, \pi(i)}. \]

First note that the only permutations \(\pi\) that produce nonzero products are those that contain \(n-1\) entries in \(G, n-1\) entries in \(G^T\), and the remaining \(m-2(n-1)\) entries on the diagonal of \(R\). The following claim is easy and goes back to Poincaré:

**Claim 3.** Let \(G(S)\) be an \((n-1) \times (n-1)\) submatrix of \(G\) corresponding to a set of \(n-1\) edges \(S\). Then \(det \ G(S) = \pm 1\) if \(S\) is a tree, and otherwise \(det \ G(S) = 0\). Moreover, if \(det \ G(S) \neq 0\), then for only one permutation \(\tau\) is \(\prod_{i=1}^{m+n-1} G(S)_{i, \pi(i)}\) nonzero.

**Proof.** First notice that if \(S\) is not a tree then it must contain a cycle and the columns corresponding to the cycle sum to zero, so \(G(S)\) is singular. So suppose \(S\) is a tree and \(\tau\) is a permutation that makes a nonzero contribution to the determinant of \(G(S)\). The column of each edge \((i, n)\) incident to \(n\) has only one nonzero entry, so \(\tau\) must choose it. Similarly, \(\tau\) must include the \(j\)th entry of the column corresponding to an edge \((j, i)\) if \((i, n) \in S\). Repeating this argument, we see that in the column of any edge \((a, b) \in S\), \(\tau\) must select the node further from \(n\) in the tree \(S\).

It follows that for any set \(S\) of edges containing a cycle, zero is the net contribution of all permutations \(\pi\) that have entries in columns \(S\) of \(G\) and are fixed elsewhere. Because \(\pi\) must have \(m - 2(n-1)\) entries on the diagonal of \(R\), the columns \(S\) containing the \(n-1\) entries in \(G\) must in fact be the same as the rows containing the \(n-1\) entries in \(G^T\). Thus the product of \(\pi\)'s entries in \(G\) and \(G^T\) is 1, and
\[
\prod_{i=1}^{m+n-1} M_{i, \pi(i)} = \prod_{(s, t) \notin S} r_{st},
\]
Moreover, \( \text{sgn}(\pi) = (-1)^{n-1} \) because \( \pi \) is the product of \( n-1 \) transpositions corresponding to the matching entries in \( G \) and \( G^T \). Thus, we have

\[
\det M = \sum_{\text{trees } T} (-1)^{n-1} \prod_{(x, y) \notin T} r_{xy}.
\]

The situation with \( \det M(e, e) \) is the same except that \( S \) cannot contain \( e \), nor will \( r_e \) be included in the product:

\[
\det M(e, e) = \sum_{\text{trees } T \neq e} (-1)^{n-1} \frac{1}{r_e} \prod_{(x, y) \notin T} r_{xy}.
\]

The sign of the permutation is still \((-1)^{n-1}\), the same sign as \( \det M \). This proves Eq. (1) and completes the proof of Kirchhoff’s theorem.

4. THE MATRIX TREE THEOREMS

In this section, we prove Claims 1 and 2. Claim 1 with the nonzero \( \sigma_{ij} \) set to 1 is exactly the matrix tree theorem (e.g., [Bi93, BR91, CK78, Ch82, VW92]). Claim 2 is in the same spirit, but is less standard. The proofs of both depend on the Binet–Cauchy theorem of linear algebra (see [Ga59, p. 8] or [VW92, p. 450]):

**Lemma 1 (Binet–Cauchy).** If \( M \) and \( N \) are matrices of order \((n-1) \times m\) and \( m \times (n-1)\), respectively, \((m \geq n - 1)\), then

\[
\det MN = \sum_{S \subset [m], |S| = n-1} (\det M_S)(\det N_S),
\]

where \( M_S \) (resp. \( N_S \)) is the submatrix formed by the \( n-1 \) columns (resp. rows) \( S \) of \( M \) (resp. \( N \)).

Recall that \( Q \) is an \((n-1) \times (n-1)\) principal submatrix of the Laplacian of \((\sigma_{ij})\).

**Claim 1.**

\[
\det Q = \sum_{\text{trees } T} w(T).
\]

**Proof.** Let \( M = N^T \) be similar to the incidence matrix \( G \) with the row of node \( n \) deleted (as in the proof of Theorem 1), but replacing the 1 in column of edge \((i, j)\) with \( \sqrt{\sigma_{ij}} \) and the -1 with \(-\sqrt{\sigma_{ij}}\).

It should be clear that \( MN = Q \): In the diagonal position \((MN)_{ii}\) we have the dot product of node \( i \)'s row in \( M \) with itself, which is equal to the sum of the conductances of the edges incident to node \( i \). In the off-diagonal position \((MN)_{ij}\), we have the dot product of node \( i \)'s row with node \( j \)'s row, which is nonzero only if there is an edge between \( i \) and \( j \), and in this case equals \(-\sigma_{ij}\).

Now from Claim 3, we can see that \( \det MN = \Delta \). Claim 3 says that an \((n-1) \times (n-1)\) submatrix of \( M \) has a nonzero determinant if and only if the columns \( S \)
correspond to a tree. Furthermore, we showed that if the columns $S$ do form a tree, then there is only one permutation $\tau$ such that $\prod_{i=1}^m M(s)_{i,\tau(i)}$ is nonzero. Here, that product is equal to $\pm \prod_{(x,y) \in S} \sigma_{xy}$. As $N = M^T$, it follows that $(\det M_S)(\det N_S) = \prod_{(x,y) \in S} \sigma_{xy}$ if $S$ is a tree. The claim now follows from the Binet–Cauchy theorem. 

Claim 2.

$$\det Q(s, t) = (-1)^{s + t} \cdot \frac{1}{2}(A_m + A_n - A_s).$$

Proof. Using notation from the preceding proof, since $Q = MN$ we have $Q(s, t) = M(s) N(t)$ where $M(s)$ is $M$ with row $s$ deleted and $N(t)$ is $N$ with column $t$ deleted. We know that $\det M(s) N(t) = \sum_{S \in [m], \#S = n-2} \det M_S(s) \cdot \det N_S(t)$.

For subsets $X$ and $Y$ of nodes, let $F(X, Y)$ denote the set of maximal forests $F$ such that for any edge $(x, y)$ with $x \in X$ and $y \in Y$, $F \cup (x, y)$ is a tree. Equivalently, any $F \in F(X, Y)$ is a forest of $n - 2$ edges containing all of $X$ in one component and all of $Y$ in the other. We will prove that

$$(-1)^{s + t} \det Q(s, t) = \sum_{F \in F\{s, t\}, \{n\}} w(F) = \frac{1}{2}(A_m + A_n - A_s).$$

Using the fact that $F\{s\}, \{n\} = F\{s, t\}, \{n\} \cup F\{s\}, \{n, t\}$, it is easy to see that the right-hand side corresponds to this sum: By definition,

$$A_m$$ is the sum over $F\{s, t\}, \{n\} \cup F\{s\}, \{t, n\}$,

$$A_n$$ is the sum over $F\{n, s\}, \{t\} \cup F\{n\}, \{s, t\}$,

and $A_s$ is the sum over $F\{s, n\}, \{t\} \cup F\{s\}, \{n, t\}$, and so $(A_m + A_n - A_s)/2$ is the sum over $F\{s, t\}, \{n\}$.

So we must prove that

$$\det M(s) N(t) = (-1)^{s + t} \sum_{F \in F\{s, t\}, \{n\}} w(F).$$

We first claim that if $\det M_S(s) \neq 0$ then $S$ is a set of $n - 2$ acyclic edges with no path from $s$ to $n$—that is, $S \notin F\{s\}, \{n\}$. We know already that if $S$ contains a cycle then the determinant is zero, so we must show that $S$ contains no path from $s$ to $n$. Let $\pi$ be a permutation whose entries have a nonzero product. For such a $\pi$, assign a direction to each edge $(x, y) \in S$: $x \rightarrow y$ if in the column of edge $(x, y)$, $\pi$ chooses the entry in row $y$. Note that a node can have at most one edge directed into it because $\pi$ chooses exactly one entry in each row. Now consider any path from $s$ to $n$ in $S$: nodes $s$ and $n$ must have their incident path edges directed away from them because rows $s$ and $n$ are not present in $M_S(s)$. This implies that some node along the path must have two edges directed into it, a contradiction.

Similarly, if $\det N_S(t) \neq 0$ then $S \in F\{t\}, \{n\}$. Thus, $\det M_S(s) \cdot \det N_S(t) \neq 0$ only if $S \in F\{s, t\}, \{n\}$. Now by the same reasoning as Claim 3, if $S \in F\{s, t\}, \{n\}$
there is exactly one permutation \( \pi \) whose entries have a nonzero product, whose magnitude will be \( \prod_{(x, y) \in S} \sqrt{\sigma_{xy}} \). So for \( S \in F(\{s, t\}, \{n\}) \), we have \( \det M_g(s) N_g(t) = \omega(S) \).

Thus, it only remains to show that the sign of \( \det M_g(s) \det N_g(t) \) is \((-1)^{s+t}\). Let us replace the entries of \( M_g = (N_g)^T \) by their signs, \( \pm 1 \). Then \( M_g \) is simply the incidence matrix for nodes \( \{1, \ldots, n-1\} \) and the \( n-2 \) edges in \( S \). If we add to \( M_g \) an extra column corresponding to the (hypothetical) edge \((s, t)\) with \(+1\) in row \( s \) and \(-1\) in row \( t \), then the determinant of this new \((n-1) \times (n-1)\) matrix is 0: Either the new column already existed in \( M_g \) or the addition of edge \((s, t)\) produces a cycle, since \( S \in F(\{s, t\}, \{n\}) \) already contains a path from \( s \) to \( t \). Expanding the determinant along this new column, we have

\[
0 = (-1)^{s+t-1} \det M_g(s) - (-1)^{t-1} \det M_g(t),
\]

and since \( \det M_g(s) \) and \( \det M_g(t) = \det N_g(t) \) are each \( \pm 1 \), this gives the desired result.

### 5. RANDOM WALKS AND ELECTRICAL NETWORKS

Finally, we briefly mention how the competitive ratios of the two random walks are obtained using the beautiful analogy with electrical networks. A very nice presentation of this analogy is given in the monograph of Doyle and Snell ([DS84]). (See also [AF94].) Chandra et al. ([CRRST89]) extended the analogy with a theorem that will give us bounds on the expected cost of both random walks.

Let \( v_x \) denote the voltage at node \( x \) and \( r_{xy} = 1/\sigma_{xy} \) the resistance of each edge in \( E \) \( (\sigma_{xy} = 0 \text{ if } (x, y) \notin E) \), and let \( \sigma_x = \sum \sigma_{xy} \) so \( p_{xy} = \sigma_{xy}/\sigma_x \). Doyle and Snell proved the following correspondence:

**Theorem 2 [DS84].** Apply a unit voltage between nodes \( a \) and \( b \), making \( v_a = 1 \) and \( v_b = 0 \). Then for any node \( x \), \( v_x \) equals the probability that a random walk starting from \( x \) will reach \( a \) before \( b \).

The commute cost \( C_{ab} \), is the expected cost of a random walk started at node \( a \) to reach node \( b \) and return to \( a \). Chandra et al. extended the analogy with a theorem bounding the commute cost in terms of the effective resistance \( R_{ab} \). We will outline the proof of the general case where each edge \((x, y)\) has both a resistance \( r_{xy} = r_{yx} \), which determines the transition probability according to the harmonic random walk and a possibly asymmetric cost function \( f_{xy} \) which we use to compute the cost of traversal. Define \( f_{xy} = f_{yx} = \sigma_{xy} = 0 \) if edge \((x, y)\) is not present. Let \( F = \sum_{(x, y)} (f_{xy} + f_{yx})/r_{xy} \).

**Theorem 3.** [CRRST89]. \( C_{ab} \leq R_{ab} \cdot F \).

**Proof.** Consider the behavior of the electrical network when into each node \( x \) we inject current \( i_x = \sum_y f_{xy}/r_{xy} \) and from node \( b \) we remove \( \sum_x i_x = F \) units of current. Let \( v_{ab} = v_a - v_b \). Consider also the setup when current \( i_x \) is removed from each node \( x \) and \( F \) units are injected into node \( a \); let \( v_{ba} = v_a - v_b \) in this setup. It is easy to see that the superposition, or sum, of the two voltage functions gives
another function such that current is conserved at all nodes except \( a \) and \( b \), where we have \( F \) units going into the network at \( a \) and \( F \) leaving at \( b \). The voltage at \( a \) minus that at \( b \) is now \( v_{ab} \), which is equal to \( F \cdot R_{ab} \) by the definition of \( R_{ab} \).

It only remains to show that \( v_{ab} \) is equal to the expected cost of a random walk that starts at \( a \) and ends when it reaches \( b \) (the same argument will hold for \( v_{ba} \)). Let us define \( v_{b} = 0 \), so that \( v_{x} \) denotes the voltage difference between nodes \( x \) and \( b \), and \( v_{ab} = v_{a} \). At any node \( x \), conservation of current gives

\[
\sum_y f_{xy} \sigma_{xy} = i_x = \sum_y (v_x - v_y) \sigma_{xy}.
\]

Let \( h_x \) denote the expected cost of a walk that starts at \( a \) and continues until reaching \( b \). Then

\[
h_x = \sum_y p_{xy}(f_{xy} + h_y) = \sum_y \sigma_{xy} (f_{xy} + h_y),
\]

or

\[
h_x \sigma_{xy} - \sum_y h_y \sigma_{xy} = \sum_y \sigma_{xy} f_{xy}.
\]

We see that these are the same systems of \( n - 1 \) linear equations (note \( v_b = h_b = 0 \)) having a unique solution, so \( v_x = h_x \).

We now use Theorem 3 to deduce the competitive ratios for the harmonic random walk and resistive random walks. In our framework, both random walks determine transition probabilities according to the edge conductances \( (\sigma_{ij}) \), but the harmonic random walk pays costs according to \( (\sigma_{ij}) \) while the resistive random walk pays according to \( (R_{ij}) \).

**Corollary 2.** The harmonic random walk is \( 2|E| \)-competitive.

**Proof.** Apply Theorem 3 with \( f_{xy} = f_{yx} = r_{xy} \) so \( F = 2|E| \). The expected cost of a walk from \( a \) to \( b \) is obviously less than the commute cost \( C_{ab} = 2|E| R_{ab} \). On the other hand, the length of the shortest path from \( a \) to \( b \) is at least \( R_{ab} \). The effective resistance \((1/r_1) + (1/r_2)\)^\(-1\) of two resistors \( r_1 \) and \( r_2 \) in parallel is always less than either \( r_1 \) or \( r_2 \), so the effective resistance between \( a \) and \( b \) will be less than the resistance of any single path between them.

**Corollary 3.** The resistive random walk is \( (2n - 3) \)-competitive.

**Proof.** We have \( f_{xy} = f_{yx} = R_{xy} \) in the theorem, so \( F = \sum_{(x,y)} 2R_{xy} \sigma_{xy} \). By Foster’s theorem, this is equal to \( 2(n - 1) \). Now the expected cost of a walk from \( a \) to \( b \) is at most the commute cost \( C_{ab} \) minus the least possible cost of a walk from \( b \) to \( a \). Again, any path from \( b \) to \( a \) costs at least \( R_{ab} \), here because costs are calculated with respect to \( (R_{ij}) \), which satisfy the triangle inequality. So the expected cost for the random walk is at most \( F \cdot R_{ab} - R_{ab} = (2(n - 1) - 1) R_{ab} \), whereas the shortest path costs at least \( R_{ab} \).
APPENDIX

Differences from [CDRS93]

The analysis of competitive ratios presented here differs somewhat from that of Coppersmith et al. The source of the difference is in the definition of competitiveness. Whereas we defined competitiveness with respect to any pair of nodes—that is, one round of the cat-and-mouse game—the definition in [CDRS93] is with respect to multiple rounds of this game and allows for an additive constant: A strategy for the cat-and-mouse game is defined to be $c$-competitive if there is an additive constant $a$, such that for any sequence of nodes $v_1, \ldots, v_n$ visited by the mouse over the course of $n$ rounds, the expected cost to the cat is bounded by $a + c$ times the mouse’s cost.

This definition of competitiveness reduces the analysis to only the simple cycles of the graph, by letting $a$ equal $c$ times the largest weight of any edge in the graph. To show an upper bound for an arbitrary path, first close the path to form a cycle, and then divide it into the sum of simple cycles. If the expected cost to traverse all the cycles is at most $c$ times the sum of the weights of the cycles, then it is at most $c$ times the weight of the original path plus $c$ times the weight of the added edge.

A nice advantage of reducing the problem to the analysis of cycles is that a tight bound of $n - 1$ follows. First, Coppersmith et al. we are able to prove a lower bound of $n - 1$: For any weighted graph and any random walk, there exists a cycle on which the ratio is $n - 1$. Second, an upper bound of $n - 1$ is achieved by reducing the problem to analyzing two cycles, which is due to the fact that the resistive random walk is a reversible Markov process (see [CDRS93] for details). Now for any edge $(a, b)$ Theorem 3 gives $C_{ab} = 2(n - 1)R_{ab}$, and the weight of the cycle is exactly $2R_{ab}$, yielding a ratio of $n - 1$.

In deriving results for the $k$-server problem in infinite metric spaces, it is not possible to choose a bounded additive constant $a$ and so the only bound obtained (for the case of a lazy adversary) is $2(k + 1) - 3 = 2k - 1$, as stated in [CDRS93].

Received April 1, 1997; final manuscript received January 1, 1998

REFERENCES


