On the Kirchhoff and the Wiener Indices of Graphs and Block Decomposition

Ashkan Nikseresht[*] Zahra Sepasdar

Department of Mathematics
Shiraz University
Shiraz, Iran

ashkan_nikseresht@yahoo.com zsepasdar@yahoo.com

Submitted: Jun 25, 2013; Accepted: Jan 19, 2014; Published: Feb 7, 2014
Mathematics Subject Classifications: 05C12, 05C85, 05C50.

Abstract

In this article we state a relation between the Kirchhoff and Wiener indices of a simple connected graph G and the Kirchhoff and Wiener indices of those subgraphs of G which are induced by its blocks. Then as an application, we define a composition of a rooted tree T and a graph G and calculate its Kirchhoff index in terms of parameters of T and G. Finally, we present an algorithm for computing the resistance distances and the Kirchhoff index and a similar one for computing the weighted distances and the Wiener index of a graph. These algorithms are asymptotically faster than the previously known algorithms, on graphs in which the order of the subgraphs induced by blocks is small with respect to the order of the graph.

Keywords: Kirchhoff index; Wiener index; Resistance distance; Shortest path problem; Block Decomposition

1 Introduction

In this article, all graphs are simple, we assume that G is a connected graph and by $\mathrm{V}(G)$, $n(G)$, $\mathrm{E}(G)$ and $\mathrm{N}(x)$ we mean the set of vertices of G, number of its vertices, the set of edges of G and the set of neighbors of a vertex x, respectively. The concept of resistance distance was first introduced by Klein and Randić [7]. Recently this concept has got a wide attention from different authors especially those interested in applications in quantum chemistry, see for example [1, 6, 2, 8, 5, 11, 10, 12].

If we view G as an electrical network and replace each edge e of G with a resistance ρ_e, the *resistance distance* between vertices a and b, denoted by $r_{ab}(G)$ or r_{ab} if there is no

[*]Partially funded by the National Elite Foundation of Iran.

ambiguity, is defined to be the effective resistance between vertices a and b as computed by Ohm's and Kirchhoff's laws. To describe it more concretely, let $I(x, y) = I_{xy}$ be a positive real valued function defined on all pairs of adjacent vertices of G and $P(x)$ be a real valued function defined on $\mathrm{V}(G)$. Fix two vertices a and b of G and assume that I and P satisfy the following conditions.

(i) $I_{xy} = -I_{yx}$,

(ii) $\sum_{y\in \mathrm{N}(x)} I_{xy} = (\delta_{ax} - \delta_{bx})I_0$ for all vertices x where I_0 is a constant,

(iii) $I_{xy}\rho_{xy} = P(x) - P(y)$ for all pairs of adjacent vertices x, y.

Then we say that I is a *flow function* from a to b (or an (a, b)-flow function) with *total flow* I_0 and P a *potential function* for I. In this case we say $r_{ab} = \frac{P(a)-P(b)}{I_0}$ is the resistance distance between a and b (see [7]). Also we set $r_{aa} = 0$.

Note that we can replace (iii) above with

(iii′) $\sum_{1\leqslant i<k} I_{x_i x_{i+1}}\rho_{x_i x_{i+1}} = 0$ for all cycles $x_1, x_2, \ldots, x_{k-1}, x_1$ in G.

It is easy to see that if (iii′) holds, then a potential function satisfying (iii) exists and is unique up to a constant summand. If a, b are two vertices lying in different components of a graph H, we define $r_{ab}(H) = \infty$. Now the *Kirchhoff index* of G is defined as:

$$\mathrm{Kf}(G) = \frac{1}{2} \sum_{a,b\in \mathrm{V}(G)} r_{ab}.$$

If we define the $n \times n$ weighted adjacency matrix $A(G) = (a_{ij})$ of G as:

$$a_{ij} = \begin{cases} \frac{1}{\rho_{ij}} & i \sim j \\ 0 & \text{otherwise} \end{cases}$$

and the diagonal matrix of vertex degrees $D(G)$ as $D(G) = diag(\sum_{i=1}^{n} a_{1i}, \ldots, \sum_{i=1}^{n} a_{ni})$, then the *Laplacian matrix* of G is $L(G) = D(G) - A(G)$. Let $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$ be the eigenvalues of $L(G)$, called the *Laplacian eigenvalues* of G, then $\lambda_1 = 0$ and for $k = 2, \ldots, n$, λ_k are nonzero real numbers (since G is connected, see [9]). Theorem F of [7] implies that:

Theorem 1.1. *For any connected n-vertex graph G ($n \geqslant 2$):*

$$\mathrm{Kf}(G) = n \sum_{k=2}^{n} \frac{1}{\lambda_k}$$

Now let w be a positive weight function on $\mathrm{E}(G)$. If $d(x, y)$ denotes the weighted distance of x and y (that is, $\min\{\sum_{e\in P} w(e)\}$ where P runs through all (x, y)-paths), then $\mathrm{W}(G) = \frac{1}{2}\sum_{x,y\in \mathrm{V}(G)} d(x, y)$ is called the *Wiener index* of G.

Here in Section 2, we state a relation between the Kirchhoff and Wiener indices of a simple connected graph G and the Kirchhoff and Wiener indices of those subgraphs of G which are induced by its blocks. As an application of this relation, we define a composition of a rooted tree T and a graph G and calculate its Kirchhoff index in terms of parameters of T and G. Also we show that using these technics one can compute the Kirchhoff index of the weighted barbell graph easily.

In Section 3, we present an algorithm for computing the effective resistance distances and the Kirchhoff index (or the weighted distances and the Wiener index) of a graph, that is asymptotically faster than the previously known algorithms when the order of the subgraphs induced by blocks is small relative to the order of the graph itself. Note that in case of weighted distances and Wiener index, the presented algorithm enhances the many algorithms for solving the well-known problem of finding all-pairs shortest paths in a graph such as Johnson's Algorithm.

2 Block decomposition and the Kirchhoff and Wiener indices

We start with some lemmas, required to obtain a relation between $\mathrm{Kf}(G)$ and Kirchhoff index of blocks of G. Assume that k is a cut vertex and H_1 and H_2 are subgraphs of G. Here we say that (H_1, H_2) is a *k-separation* of G, when $G = H_1 \cup H_2$ and $\{k\} = H_1 \cap H_2$.

Lemma 2.1 ([7, Lemma E]). *Let k be a cut vertex of a graph G and (H_1, H_2) be a k-separation of G. If $i \in \mathrm{V}(H_1)$ and $j \in \mathrm{V}(H_2)$, then $r_{ij}(G) = r_{ik}(H_1) + r_{kj}(H_2)$.*

Lemma 2.2. *Let k be a cut vertex of G and (H_1, H_2) be a k-separation of G. If $i, j \in \mathrm{V}(H_1)$ then $r_{ij}(G) = r_{ij}(H_1)$.*

Proof. Let I be the (i, j)-flow function of H_1 and extend it to G by setting $I = 0$ in H_2. Now clearly I is an (i, j)-flow of G and hence the result follows. □

In what follows, for each $i \in \mathrm{V}(G)$, we set $\mathrm{Kf}_G(i) = \sum_{i\neq j\in \mathrm{V}(G)} r_{ij}$. Obviously $\mathrm{Kf}(G)$ can be written as $\mathrm{Kf}(G) = \frac{1}{2}\sum_{i\in\mathrm{V}(G)} \mathrm{Kf}_G(i)$. $\mathrm{W}_G(i)$ is defined similarly.

Theorem 2.3. *Suppose that G is a graph, $B_1, \dots, B_r$ are its blocks and S is the set of its cut vertices. For each $x \in S$, if the component of $G - B_i$ that contains x is named H_{ix}, set $C_{ix} = n(H_{ix}) - 1$. Also let $G_i = G[B_i]$. Then*

$$\mathrm{Kf}(G) = \sum_{i=1}^{r}\left(\mathrm{Kf}(G_i) + \sum_{x\in S\cap G_i} C_{ix}\mathrm{Kf}_{G_i}(x) + \sum_{x,y\in S\cap G_i} C_{ix}C_{iy}r_{xy}(G_i)\right). \tag{1}$$

Proof. Let $u \in \mathrm{V}(G_i)$, $v \in \mathrm{V}(G_j)$. If $i = j$, then $r_{uv}(G) = r_{uv}(G_i)$ by (2.2), hence the sum of all such $r_{uv}(G)$'s gives the term $A = \sum_{i=1}^{r} \mathrm{Kf}(G_i)$.

Thus suppose $i \neq j$ and assume that $i = i_0, i_1, \dots, i_k = j$ are the indices of the blocks in any path from u to v. Also suppose that $x_1, x_2, \dots, x_k$ are the cut vertices of G in any such path, that is, $\{x_l\} = G_{i_{l-1}} \cap G_{i_l}$ $(1 \leqslant l \leqslant k)$. We can assume that $u \neq x_1$ and $v \neq x_k$. Then it follows from (2.1) and (2.2), that

$$r_{uv}(G) = r_{ux_1}(G_{i_0}) + r_{x_1x_2}(G_{i_1}) + \cdots + r_{x_{k-1}x_k}(G_{i_{k-1}}) + r_{x_kv}(G_{i_k}). \tag{2}$$

Now we must sum up equations of the form (2), for all vertices u, v of G not lying in the same block of G. Assume that $x, y \in S \cap G_i$. If the term $r_{xy}(G_i)$ appears in (2) for some such u, v, then clearly u, v are vertices of different components of $G - B_i$. In fact u and v are in the same components as x and y (that is, H_{ix} and H_{iy}), respectively. Also by assumption $u \neq x$ and $v \neq y$. Conversely, (2) contains $r_{xy}(G_i)$ for all such u, v. Thus this term shows up in the sum for $C_{ix}C_{iy}$ times, and all such terms contribute $B = \sum_{i=1}^{r} \sum_{x,y \in S \cap G_i} C_{ix}C_{iy}r_{xy}(G_i)$ to $\mathrm{Kf}(G)$.

Now suppose $x \in S \cap G_i$ and $x \neq u \in \mathrm{V}(G_i)$. Then r_{ux} occurs in (2) (as the first or the last term), exactly when $v \neq x$ is in the same component of $G - B_i$ as x, that is, H_{ix}. Therefore, summing up all the flanking terms of (2), we get

$$C = \sum_{i=1}^{r} \sum_{x \in S \cap G_i} \sum_{x \neq u \in \mathrm{V}(G_i)} C_{ix} r_{ux}(G_i) = \sum_{i=1}^{r} \sum_{x \in S \cap G_i} C_{ix} \mathrm{Kf}_{G_i}(x).$$

Now $\mathrm{Kf}(G) = A + B + C$ and the result is established. □

Remark 2.4. Assume that in the above theorem B_i's, instead of blocks, were unions of blocks such that $\{B_1, \ldots, B_r\}$ were a partition of $\mathrm{E}(G)$ with each $G[B_i]$ connected, and S were the set of cut vertices of G which lie on at least two of the B_i's. Then from the proof of the theorem it can be seen that Formula (1) would still be true.

Remark 2.5. It is clear that Lemmas (2.1) and (2.2) remain true, if we replace $r_{x,y}$ with the weighted distance function $d(x, y)$. Therefore in (2.3), if we replace every occurrence of Kf with W (Wiener index), then the same argument shows that the theorem will still be true.

As an example for computing Kirchhoff and Wiener indices by (2.3) we define a new composition of graphs and compute its Kirchhoff and Wiener indices.

Definition 2.6. Let T be a rooted tree, $x, y \in \mathrm{V}(G)$ and G_i's be $n(T)$ disjoint copies of G with x_i and y_i denoting the copies of x and y in G_i, respectively. By $T[G, x, y]$ (or $T[G]$ if there is no ambiguity) we mean the graph obtained by unifying y_i and x_j for each $i \in \mathrm{V}(T)$ and each child j of i, in the disjoint union of G_i's.

Figure 1, illustrates this composition of T and G.

Theorem 2.7. *Let T be a rooted tree, G' a connected graph and $x, y \in \mathrm{V}(G')$. Set $G = T[G', x, y]$, $n = n(T)$, $m = n(G')$. Also assume that $\mathrm{W}(T)$ is the Wiener index of T when all edges have unit weight. Then*

$$\begin{aligned}\mathrm{Kf}(T[G]) =& n\mathrm{Kf}(G') + \mathrm{W}_T(1)(m-1)\mathrm{Kf}_{G'}(y) + \big(n(n-1) - \mathrm{W}_T(1)\big)(m-1)\mathrm{Kf}_{G'}(x) \\ &+ \big(\mathrm{W}(T) + \mathrm{W}_T(1) - n(n-1)\big)(m-1)^2 r_{xy}(G')\end{aligned}$$

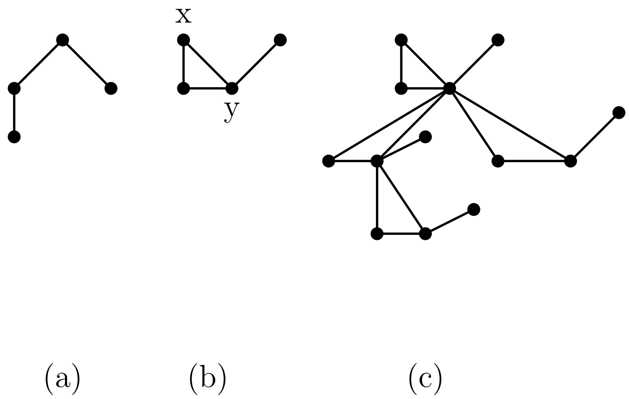

Figure 1: (a) a rooted tree T; (b) a graph G; (c) $T[G]$.

Proof. Suppose that $\mathrm{V}(T) = \{1, \ldots, n\}$ and 1 is the root of T. Let G'_i denote the i'th copy of G' and x_i and y_i denote the copies of x and y in G'_i, respectively. According to (2.4), we can apply (2.3) with $B_i = \mathrm{E}(G'_i)$ (and hence $G_i = G'i$) and S the set of all x_i's and y_i's. Assume that d_i denotes the number of descendants of i in T different from i itself and $C_{ix} = C_{ix_i}$ and $C_{iy} = C_{iy_i}$ are as in (2.3). It is easy to see that $C_{iy} = d_i(m-1)$ and $C_{ix} = (n - d_i - 1)(m-1)$.

Now in $\sum_{i=1}^{n} d_i$ a vertex i of T with level l_i is counted $l_i = d(1, i)$ times. Therefore

$$\sum_{i=1}^{n} d_i = \sum_{i=1}^{n} l_i = W_T(1) \tag{3}$$

and hence if we set $a_i = n - d_i - 1$, then

$$\sum_{i=1}^{n} a_i = n(n-1) - \mathrm{W}_T(1). \tag{4}$$

Also $(d_i + 1)a_i$ is the number of paths in T which pass the edge connecting i and its parent. So $\sum_{i=1}^{n}(d_i + 1)a_i = \sum_{\{i,j\} \subseteq \mathrm{V}(T)} d(i, j) = \mathrm{W}(T)$. Consequently:

$$\sum_{i=1}^{n} d_i a_i = \mathrm{W}(T) - \sum_{i=1}^{n} a_i = \mathrm{W}(T) + \mathrm{W}_T(1) - n(n-1). \tag{5}$$

Therefore, the proof is concluded by substituting C_{ix}'s and C_{iy}'s in Equation (1), applying Equations (3)–(5) and noting that $\mathrm{Kf}(G'_i) = \mathrm{Kf}(G')$, $\mathrm{Kf}_{G'_i}(x_i) = \mathrm{Kf}_{G'}(x)$, $\mathrm{Kf}_{G'_i}(y_i) = \mathrm{Kf}_{G'}(y)$ and $r_{x_i y_i}(G'_i) = r_{xy}(G')$. □

Remark 2.8. By (2.5), we have a result similar to (2.7), for $\mathrm{W}(T[G])$. But one must makes a difference between the weighted Wiener index of a graph and the "unweighted" Wiener index. The former is based on an arbitrary weight function on $\mathrm{E}(G)$, as in the introduction. But the latter assumes the weight of all edges to be unit. Suppose that $\mathrm{W}(G)$ denotes the unweighted one and $\mathrm{W}'(G)$ the weighted one. Then to obtain a formula for $\mathrm{W}(T[G])$ one must replace all Kf's in (2.7) with W′, the weighted wiener index. But the W's in the formula must remain unchanged as the unweighted Wiener index. Also $r_{xy}(G')$ must be replaced with $d_{G'}(x, y)$ (weighted distance).

As another example we calculate the Kirchhoff index of the weighted barbell graph using (2.3). This value has been calculated in [1] using another method but we will see that (2.3) allows us to apply some elementary methods and hence makes the computation easier.

The *weighted barbell graph* is a simple weighted graph G on $n = k + m + r$ vertices, where $m \geqslant 2$ and $k, r \geqslant 1$: start with a weighted path G_2 on m vertices, labelled as $\{x_{k+1}, \dots, x_{k+m}\}$ and attach a complete graph G_1 of order $k + 1$ at vertex $y_1 = x_{k+1}$ and a complete graph G_3 of order $r + 1$ at vertex $y_2 = x_{k+m}$. Denote by $\{x_1, \dots, x_k\}$ the set of new vertices of the complete graph attached to y_1 and by $\{x_{k+m+1}, \dots, x_{k+m+r}\}$ the set of new vertices of the complete graph attached to y_2.

Moreover, the weights are given by $\rho_{ij} = a$, $1 \leqslant i \leqslant j \leqslant k$; $\rho_{i,k+1} = b$, $1 \leqslant i \leqslant k$; $\rho_{k+i,k+i+1} = c_i$, $1 \leqslant i \leqslant m - 1$; $\rho_{k+m,k+m+i} = b'$, $1 \leqslant i \leqslant r$, and $\rho_{k+m+i,k+m+j} = a'$, $1 \leqslant i \leqslant j \leqslant r$. It should be mentioned that in [1], the weights on the graph denote the so called conductances not the resistances. The relation is: an edge with conductance c has resistance $\frac{1}{c}$.

Note that when $k = r = 1$, then G is nothing than a weighted path on $m + 2$ vertices whose weights are $b, c_1, c_2, \dots, c_{m-1}, b'$. Figure 2 shows a weighted barbell graph with $k = m = 3$, $r = 2$, $a = c_1 = 1$, $b = c_2 = 2$, $a' = 4$ and $b' = 3$.

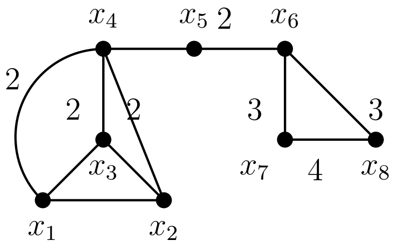

Figure 2: A weighted barbell graph with $k = m = 3$ and $r = 2$. The resistances of the edges without label is assumed to be 1.

We want to apply (2.3) and (2.4) on G with $B_i = \mathrm{E}(G_i)$ $(i = 1, 2, 3)$ and $S = \{y_1, y_2\}$. Thus first we need to compute Kirchhoff index and resistance distances of G_1 and G_3. We can do this using some elementary technics, which could not be applied on the whole G at once.

Lemma 2.9. *Let $0 \leqslant a, b \in \mathbb{R}$ and assume that $A = (a_{ij})$ is an $n \times n$ matrix with $n \geqslant 2$*

such that:

$$a_{ij}=\begin{cases} -a & i\neq j,\ i,j\neq 1\\ (n-2)a+b & i=j\neq 1\\ (n-1)b & i=j=1\\ -b & else\end{cases}.$$

Then the eigenvalues of A are 0, nb with multiplicity one, $(n-1)a+b$ with multiplicity $(n-2)$.

Proof. Let $B=\lambda I-A$. Then by adding the sum of rows $2,3,\dots,n$ to the first row and then factorizing λ, we see that $\det B=\lambda\det C$ where C is the same as B except on the first row, where every entry of C is 1. Next by successively subtracting the i'th row of C from its $(i-1)$'th row for $i=3,\dots,n$, we get that:

$$\det B=\lambda\begin{vmatrix} 1 & 1 & 1 & 1 & \dots & 1\\ 0 & \lambda-(n-1)a-b & -(\lambda-(n-1)a-b) & 0 & \dots & 0\\ 0 & 0 & \lambda-(n-1)a-b & \dots & \dots & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\ b & a & a & a & \dots & \lambda-(n-2)a-b\end{vmatrix}.$$

Now by dividing rows $2,\dots,n-1$ of the above matrix by $\lambda-(n-1)a-b$ and then subtracting b times the first row from the n'th row, we see:

$$\det B=\lambda(\lambda-(n-1)a-b)^{n-2}\begin{vmatrix} 1 & 1 & 1 & 1 & \dots & 1\\ 0 & 1 & -1 & 0 & \dots & 0\\ 0 & 0 & 1 & -1 & \dots & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & a-b & a-b & a-b & \dots & \lambda-(n-2)a-2b\end{vmatrix}.$$

At last, by successively adding $(i-1)(b-a)$ times the i'th row to the n'th row for $i=2,\dots,n-1$, the last row of the above matrix transforms to $0,0,\dots,0,\lambda-nb$. Consequently, $\det B=\lambda(\lambda-(n-1)a-b)^{n-2}(\lambda-nb)$, as required. □

In the rest of this section, we will fix the notations before Lemma (2.9).

Corollary 2.10. *In the notations before (2.9), if $1\leqslant i\neq j\leqslant k$, then:*

$$\mathrm{Kf}(G_1)=b+(k^2-1)\frac{ab}{kb+a},\quad r_{y_1,x_i}(G_1)=\frac{ab+b^2}{kb+a},\quad r_{x_i,x_j}(G_1)=\frac{2ab}{kb+a}.$$

Proof. Noting that $L(G_1)$ is in the form of the statement of (2.9) (with $\frac{1}{a}$ and $\frac{1}{b}$ instead of a and b, respectively), the formula for $\mathrm{Kf}(G_1)$ follows from (2.9) and (1.1).

Let I be an (y_1,x_1)-flow and P be the related potential function on G_1. The situation of $x_2,\dots x_k$ are symmetrical, hence $P(x_2)=P(x_3)=\cdots=P(x_k)$. So $I_{x_ix_j}=0$ for each $1<i\neq j\leqslant k$. Thus indeed, I consists of $k-1$ parallel flows from y_1 to x_1 passing x_2, x_3,

$\ldots$, x_k and one directly from y_1 to x_1. Therefore applying the well-known parallel and serial formulas we get

$$r_{y_1x_1} = \frac{1}{\frac{k-1}{a+b} + \frac{1}{b}} = \frac{ab+b^2}{kb+a}.$$

Again by symmetry it is clear that $r_{y_1x_i} = r_{y_1x_1}$ and $r_{x_ix_j} = r_{x_{i'}x_{j'}}$ for each $1 \leqslant i \neq j \leqslant k$ and $1 \leqslant i' \neq j' \leqslant k$. Consequently, if $k > 1$, then $\mathrm{Kf}(G) = kr_{y_1x_1} + \binom{k}{2}r_{x_1x_2}$ and solving this equation for $r_{x_1x_2} = r_{x_ix_j}$, we get the claimed formula. □

Theorem 2.11. *Let G be the weighted barbell graph. The Kirchhoff index of G is:*

$$\mathrm{Kf}(G) = \frac{(n-1)kab + (n-k)kb^2}{a+kb} + \frac{(n-1)ra'b' + (n-r)rb'^2}{a'+rb'} + \sum_{i=1}^{m-1}(k+i)(r+m-i)c_i$$

Proof. By (2.10) we have:

$$\mathrm{Kf}(G_1) = b + (k^2-1)\frac{ab}{a+kb}, \quad \mathrm{Kf}(G_3) = b' + (r^2-1)\frac{a'b'}{a'+rb'},$$

$$\mathrm{Kf}_{G_1}(y_1) = \sum_{j=1}^{k} r_{y_1x_j} = k\frac{ab+b^2}{kb+a} \quad \text{and} \quad \mathrm{Kf}_{G_3}(y_2) = \sum_{j=1}^{r} r_{y_3x_{k+m+j}} = r\frac{a'b'+b'^2}{rb'+a'}.$$

Also since G_2 is a weighted path we have:

$$\mathrm{Kf}_{G_2}(y_1) = \sum_{j=2}^{m} r_{y_1x_{k+j}} = \sum_{i=1}^{m-1}(m-i)c_i, \quad \mathrm{Kf}_{G_2}(y_2) = \sum_{j=1}^{m-1} r_{y_2x_{k+j}} = \sum_{i=1}^{m-1} ic_i \quad \text{and}$$

$$\mathrm{Kf}(G_2) = \sum_{i=1}^{m-1} i(m-i)c_i.$$

Now according to (2.3), the result follows by substituting these quantities in

$$\begin{aligned}\mathrm{Kf}(G) = &\sum_{i=1}^{3}\mathrm{Kf}(G_i) + (n(G_1)-1)\mathrm{Kf}_{G_2}(y_1) + (n(G_3)-1)\mathrm{Kf}_{G_2}(y_2)\\ &+(n(G_2\cup G_3)-1)\mathrm{Kf}_{G_1}(y_1) + (n(G_2\cup G_1)-1)\mathrm{Kf}_{G_3}(y_2)\\ &+(n(G_1)-1)(n(G_3)-1)r_{y_1y_2}(G_2).\end{aligned}$$

□

3 A fast algorithm for computing the Kirchhoff and Wiener indices of a graph with small blocks

Let G be a graph with vertices $\{1, 2, \dots, n\}$ and edge resistances (ρ_{ij}). According to [7, Theorem A & Corollary A], $\mathrm{Kf}(G)$ and the matrix $(r_{ij}(G))$ could be calculated using a finite number of matrix operations such as taking inverses and multiplication. The running time of algorithms for these two operations, is $\Theta(n^t)$ where $2 < t \leqslant 3$ (see [4, Sections 28.2 and 28.4] for algorithms with $t = \lg 7$). Here we give an algorithm (see Algorithm 3) for computing $\mathrm{Kf}(G)$ using (2.3). This algorithm first computes the terms $\mathrm{Kf}(G_i)$, $\mathrm{Kf}_{G_i}(x)$ and $r_{xy}(G_i)$ of Formula (1) using the previously known algorithms and then calculates $\mathrm{Kf}(G)$ from this formula.

The procedure Block-dec(G) in Algorithm 3, takes as input a graph G and decomposes it into its blocks. More concretely, it returns the list $(B, (BL_i)_{1 \leqslant i \leqslant B}, (n_{B_i})_{1 \leqslant i \leqslant B}, C, CL)$ where B is the number of blocks of G, BL_i is the list of vertices occurring in $B_i =$ block i, n_{B_i}'s are the number of vertices of $G_i = G[B_i]$, C is the number of cut vertices and CL is the list of cut vertices of G. It is well-known that this job could be done using a depth first search with $O(n^2)$ time complexity (see for example [3, Section 6.1]).

Also in Algorithm 3, the procedure Direct-ERS(G) computes the Kirchhoff index of its input graph G using matrix operations as marked above. In more details, it returns $((r_{xy}(G))_{x,y \in G}, (\mathrm{Kf}_G(x))_{x \in G}, \mathrm{Kf}(G))$.

An important part of computing $\mathrm{Kf}(G)$ using (2.3) is calculating C_{ix}'s of Formula (1). This job is done by the procedure compute-const-mult, which is described in Algorithm 1. Note that T in this procedure is the *block tree* of G (that is, a tree with blocks and cut vertices of G as vertices and edges xb's, where x is a cut vertex on a block b of G, see [3, p. 121]), hence the running time of this procedure, which is in fact a breadth first search (BFS), is $\Theta(n(T)) = \Theta(B + C) = O(n)$, where B and C are as above. It should be mentioned that in line 5 of Algorithm 1, the ending '-1' is to avoid doubly counting

Algorithm 1 Compute-const-mult$(T, b, (n_{B_i})_{1 \leqslant i \leqslant B})$ (computes the constant multipliers in Formula (1) for G).

Input: $T =$ Block tree of G via the adjacency lists of "block vertices" and "cut vertices" of G, an integer b with $1 \leqslant b \leqslant B$ and $n_{B_i} =$ the number of vertices in $G[B_i]$, where B_i is the i'th block of G $(1 \leqslant i \leqslant B)$.

Output: C_{bx} for all cut vertices which are on $G[B_b]$, where (C_{bx}) is as in (2.3).

```
1: for each $x$ adjacent to $b$ in $T$ do
2:     remove the edge $xb$ of $T$ and set $C_{bx} = 0$
3:     run a BFS on $T$ starting with $x$
4:     for each block $B_i$ that is met in this search do
5:         $C_{bx} = C_{bx} + n_{B_i} - 1$
6:     end for
7: end for
8: return $C_{bx}$'s
```

Algorithm 2 Compute-block-dist(T, b)

```
Input: $T$ = Block tree of $G$ via the adjacency lists of “block vertices” and “cut vertices”
    of $G$ and an integer $b$ with $1 \leqslant b \leqslant B$.
Output: $\mathrm{Out}_{bb'}$, $\mathrm{In}_{bb'}$ and $d_{bb'}$ for all $1 \leqslant b' \leqslant B$ as described above.
 1: for each $x$ adjacent in $T$ to $b$ do
 2:     remove the edge $xb$ of $T$ and set $d_x = 0$
 3:     run a BFS on $T$ starting with $x$ and let $P(z)$ denote the parent of $z$ in this BFS
        tree for each $z \in \mathrm{V}(T)$
 4:     for all cut vertices $y \neq x$ met in this search do
 5:         $d_y = d_{P(P(y))} + r_{y,P(P(y)}\left(G_{P(y)}\right)$
 6:     end for
 7:     for all blocks $b'$ met in this search do
 8:         $\mathrm{Out}_{bb'} = x$, $\mathrm{In}_{bb'} = P(b')$ and $d_{bb'} = d_{P(b')}$
 9:     end for
10: end for
11: return $((\mathrm{Out}_{bi})_{i=1}^{B}, (\mathrm{In}_{bi})_{i=1}^{B}, (d_{bi})_{i=1}^{B})$
```

cut vertices common in two or more G_i’s.

Since we want the procedure compute-const-mult to have linear running time, T must be represented by its adjacency list not its adjacency matrix. Lines 6–25 of Algorithm 3, construct this list. Also in these lines, block(x) is set to be a block on which x lies, for each $x \in \mathrm{V}(G)$.

Another important procedure in Algorithm 3 is Compute-block-dist(T, b). This procedure (described in Algorithm 2) gets the block tree T of G and a block b as input. For each block $b \neq b'$ of G, there are unique cut vertices x and $y \in G$, through which any path in G from block b to block b', exits block b and enters block b', respectively. We set $\mathrm{Out}_{bb'} = x$ and $\mathrm{In}_{bb'} = y$ and also $d_{bb'} = r_{xy}(G)$. Note that if $u \in G_b$ and $v \in G_{b'}$ where $G_b = G[B_b]$ and $G_{b'} = G[B_{b'}]$, then we have $r_{uv}(G) = r_{ux}(G_b) + d_{bb'} + r_{vb'}(G_{b'})$.

The output of Compute-block-dist(T, b) is $\mathrm{Out}_{bb'}$, $\mathrm{In}_{bb'}$ and $d_{bb'}$ for all $1 \leqslant b' \leqslant B$ and the specified b. These are computed using a BFS started from b in T, thus the running time of this procedure is $\theta(n(T)) = O(n(G))$. In line 5 of Algorithm 2, note that $P(y)$ is a block and $P(P(y))$ is a cut vertex of G and both y and $P(P(y))$ lie on $G_{P(y)}$.

Clearly, lines 6–16 and 17–25 of Algorithm 3, have running time $O(Bn)$ and $O(BC)$, respectively. Moreover, lines 41 and 3 have running time $O\left(\sum_{i=1}^{B} n_{B_i}^2\right)$. Lines 26–29 are $O(Bn)$ and lines 30–40 are $O(n^2)$. Since line 4 is $O\left(\sum_{i=1}^{B} n_{B_i}^t\right)$, where $2 < t \leqslant 3$ and the procedures Block-dec has $O(n^2)$ time complexity, we deduce that the running time of Algorithm 3 is $O\left(\sum_{i=1}^{B} n_{B_i}^t + n^2\right)$.

Theorem 3.1. *Let $0 < c \in \mathbb{R}$ and $0 \leqslant \alpha \in \mathbb{R}$ with $\alpha < 1$. Assume that $\mathcal{F}$ is a family of graphs with the property that for each $G \in \mathcal{F}$ and each block B_i of G, the number vertices lying on B_i is $\leqslant c(n(G))^\alpha$. On $\mathcal{F}$, Algorithm 3 has a running time of $O\left(n^2 + n^{1+\alpha(t-1)}\right)$,*

Algorithm 3 Calculates effective resistance distances of G and $\mathrm{Kf}(G)$.

```
Input: A weighted graph $G$ via the edge resistance matrix $(\rho_{xy})$.
Output: Effective resistance distances $(r_{xy}(G))_{x,y\in \mathrm{V}(G)}$ and $\mathrm{Kf}(G)$.
$(B, (BL_i)_{1\leqslant i\leqslant B}, (n_{B_i})_{1\leqslant i\leqslant B}, C, CL)$ =Block-dec($G$)
for $i = 1$ to $B$ do
    set $G_i = G[BL_i]$ and compute it's edge resistance matrix
    $\big(((r_{xy}(G_i))_{x,y\in \mathrm{V}(G_i)}, (\mathrm{Kf}_x(G_i))_{x\in \mathrm{V}(G_i)}, \mathrm{Kf}(G_i)\big)$ =Direct-ERS($G_i$)
end for
for $i = 1$ to $B$ do
    for all $x$ on $BL_i$ do
        $M[i, x] = 1$
        if block($x$) is not set then
            block($x$) $= i$
        end if
    end for
    for all $x$ on $\mathrm{V}(G) - BL_i$ do
        $M[i, x] = 0$
    end for
end for
for all $x$ on $CL$ do
    for $i = 1$ to $B$ do
        if $M[i, x] = 1$ then
            add $x$ to the list $L_i$
            add $i$ to the list $L'_x$
        end if
    end for
end for
set $T$ to be the tree with adjacency list $(L_i)_{1\leqslant i\leqslant B}$ joint with $(L'_x)_{x\in CL}$
for $b = 1$ to $B$ do
    $(C_{bx})_x$ =Compute-const-mult($T, b, (n_{B_i})_{1\leqslant i\leqslant B}$)
    $((\mathrm{Out}_{bi})_{i=1}^{B}, (\mathrm{In}_{bi})_{i=1}^{B}, (d_{bi})_{i=1}^{B})$ =Compute-block-dist($T, b$)
end for
for all $x \in \mathrm{V}(G)$ do
    $b$ =block($x$)
    for all $y \in \mathrm{V}(G)$ do
        $b'$ =block($y$)
        if $b \neq b'$ then
            $r_{xy} = d_{bb'} + r_{x,\mathrm{Out}_{bb'}}(G_b) + r_{y,\mathrm{In}_{bb'}}(G_{b'})$
        else
            $r_{xy} = r_{xy}(G_b)$
        end if
    end for
end for
set Kf to be the sum in Formula (1)
return $(r_{xy})_{x,y\in \mathrm{V}(G)}$ and Kf
```

where t is as above. In particular, if $\alpha = \frac{1}{2}$, then the time complexity of Algorithm 3 is $O(n^2)$.

Proof. We use the notations in the above remarks. According to these remarks, we just need to show that $\left(\sum_{i=1}^{B} n_{B_i}^t\right)$ is $O\left(n^{1+\alpha(t-1)}\right)$. First note that if x is a cut vertex of G, then the number of blocks of G on which x lies, is $\deg_T(x)$ where T is the block tree of G. But $\sum \deg_T(x) = |\mathrm{E}(T)| = B + C - 1 \leqslant 2n$, where the sum is taken over all cut vertices x of G. Since each non-cut vertex of G lies in exactly one block of G, we conclude that $\sum_{i=1}^{B} n_{B_i} \leqslant 3n$.

Let $a_i = n_{B_i}$. By the binomial Theorem, $a^t + b^t \leqslant (a+b)^t$ for each $a, b \geqslant 0$. Thus if for some $j \neq i$, $a_i, a_j < \frac{cn^\alpha}{2}$, then by replacing a_i and a_j by $a_i + a_j$, the sum $\sum a_i$ does not change and $\sum a_i^t$ does not decrease. Therefore by repeating this operation, we can assume that for all except possibly one i we have $\frac{cn^\alpha}{2} \leqslant a_i \leqslant cn^\alpha$. Consequently, if the number of a_i's in the final sum is B', then $(B'-1)\frac{cn^\alpha}{2} \leqslant \sum_{i=1}^{B'} a_i = \sum_{i=1}^{B} n_{B_i} \leqslant 3n$. Hence $B'\frac{cn^\alpha}{2} \leqslant (3+c)n$ and so $B' \leqslant dn^{1-\alpha}$, where $d = \frac{6+2c}{c}$. Finally:

$$\sum_{i=1}^{B} n_{B_i}^t \leqslant \sum_{i=1}^{B'} a_i^t \leqslant B'(cn^\alpha)^t \leqslant (dc^t)n^{t\alpha+1-\alpha},$$

as required. The final assertion follows from $t \leqslant 3$. □

Note that, in the above notations, since $1 < t$, we have $1 + (t-1)\alpha < t$. Therefore Algorithm 3 is asymptotically faster than the methods that use matrix operations (see the starting paragraph of this section) on the family $\mathcal{F}$ of the above theorem.

Remark 3.2. According to (2.5), if we replace the procedure Direct-ERS(G_i) in Algorithm 3, with Direct-dist(G_i) which computes $d_{G_i}(x,y)$, $\mathrm{W}(G_i)$ and $\mathrm{W}_{G_i}(x)$ for all $x, y \in \mathrm{V}(G_i)$ (such as Johnson's Algorithm, see [4, Section 25.3]), then the obtained algorithm will return $\mathrm{W}(G)$ and $d_G(x,y)$ for each $x, y \in \mathrm{V}(G)$. Also if t is such that the procedure Direct-dist(G_i) can be run in $O(n^t)$, then (3.1) remains true for the obtained algorithm. Note that in particular, Johnson's Algorithm has a time complexity of $\Theta(n^2 \lg n + mn)$ where $m = |\mathrm{E}(G)|$, and hence $t = 3$ works for this algorithm.

The problem of finding all-pairs shortest paths is a well-known problem and many algorithms are devised to solve this problem (see [4, Chapter 25]). It should be mentioned that the Johnson's Algorithm has a running time of $O\left(n^2 \lg n + n\sum_{i=1}^{B} n_{B_i}^2\right)$, because $m = |\mathrm{E}(G)| \leqslant \sum_{i=1}^{B} n_{B_i}^2$. For example if we restrict ourselves to the family $\mathcal{F}$ of (3.1), according to the proof of (3.1), $\sum_{i=1}^{B} n_{B_i}^2$ is $O(n^{1+\alpha})$, hence Johnson's Algorithm runs in $O(n^{2+\alpha})$ time but Algorithm 3 in $O(n^{1+2\alpha})$ time. Since $\alpha < 1$, this means that Algorithm 3 is asymptotically faster on $\mathcal{F}$.

References

[1] E. Bendito, A. Carmon, A. M. Encinas and J. M. Gesto. A formula for the Kirchhoff index. *Int. J. Quantum Chem.*, 108(6):1200–1206, 2008.

[2] D. Bonchev, A. T. Balaban, X. Liu and D. J. Klein. Molecular cyclicity and centricity of polycyclic graphs. I. Cyclicity based on resistance distances or reciprocal distances. *Int. J. Quantum Chem.*, 50:1–20, 1994.

[3] J. A. Bondy and U. S. R. Murty. *Graph Theory*, Springer, 2008.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, MIT press, New York, 2001.

[5] P. W. Fowler. Resistance distances in fullerene graphs. *Croat. Chem. Acta*, 75:401–408, 2002.

[6] X. Gao, Y. Luo and W. Liu. Resistance distance and the Kirchhoff index in Cayley graphs. *Discrete Applied Mathematics*, 46:1–8, 2011.

[7] D. J. Klein and M. Randić. Resistance distance. *J. Math. Chem*, 12:81–95, 1993.

[8] I. Lukovits, S. Nikolić and N. Trinajstić. Note on the resistance distances in the dodecahedron. *Croat. Chem. Acta*, 73:957–967, 2000.

[9] Bojan Mohar. Some applications of Laplace eigenvalues of graphs. *Graph Symmetry: Algebraic Methods and Applications*, 497:225–275, 1997.

[10] J. L. Palacios. Closed-form formulas for Kirchhoff index. *Int. J. Quantum Chem.*, 81:135–140, 2001.

[11] J. L. Palacios. Resistance distance in graphs and random walks. *Int. J. Quantum Chem.*, 81:29–33, 2001.

[12] H. Zhang, Y. Yang and Ch. Li. Kirchhoff index of composite graphs. *Discrete Applied Mathematics*, 157:2918–2927, 2009.