Biharmonic Distance Related Centrality for Edges in Weighted Networks

Yuhao Yi[1,2], **Liren Shan**[1,2], **Huan Li**[1,2] and **Zhongzhi Zhang**[1,2,*]
[1]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, China
[2]School of Computer Science, Fudan University, China
{yhyi15, 13307130150, huanli16, zhangzz}@fudan.edu.cn

Abstract

The Kirchhoff index, defined as the sum of effective resistances over pairs all of nodes, is of primary significance in diverse contexts of complex networks. In this paper, we propose to use the rate at which the Kirchhoff index changes with respect to the change of resistance of an edge as a measure of importance for this edge in weighted networks. For an arbitrary edge, we explicitly determine the change of the Kirchhoff index and express it in terms of the biharmonic distance between its end nodes, and thus call this centrality as biharmonic distance related centrality (BDRC) . We show that BDRC has a better discriminating power than those commonly used metrics, such as edge betweenness and spanning edge centrality. We give an efficient algorithm that provides an approximation of biharmonic distance for all edges in nearly linear time of the number of edges, with a high probability. Experiment results validate the efficiency and accuracy of the presented algorithm.

1 Introduction

Measuring the importance of nodes and edges is a fundamental question at the core of network analysis [You *et al.*, 2017]. A lot of desirable centrality metrics were proposed to capture relative importance of nodes or edges from the perspective of both network topology and dynamics [White and Smyth, 2003], and various algorithms were developed to identify vital nodes and edges [You *et al.*, 2017]. However, most previous work centered on the node level. Actually, edge centrality and its corresponding algorithms also play an indispensable role in network science, as their node counterparts. In the past years, edge centrality has been widely used to a variety of aspects of graph mining and applications, including detecting communities in social and biological networks [Girvan and Newman, 2002], revealing new knowledge in semantic web [Berners-Lee *et al.*, 2001], and identifying social ties in social networks [Ding *et al.*, 2011], designing or protecting infrastructure networks [Bienstock *et al.*, 2014], to name just a few. Thus, it is of theoretical and practical interest to present measures of edge centrality and establish algorithms characterizing the relative importance of edges in networks.

In view of the relevance of edge importance, some metrics for edge centrality have been presented, with frequently used measures including edge betweenness [Brandes, 2001; Bader *et al.*, 2007; Brandes and Pich, 2007; Geisberger *et al.*, 2008], spanning edge centrality [Teixeira *et al.*, 2013; Mavroforakis *et al.*, 2015; Hayashi *et al.*, 2016], and current-flow centrality [Brandes and Fleischer, 2005]. The betweenness of an edge is defined as the fraction of shortest paths between all pairs of nodes that pass through the edge. The spanning edge centrality of an edge is the probability that it is in a uniformly selected spanning tree. While the current-flow centrality of an edge is defined as the amount of current flowing through this edge. Despite their wide applications, these edge centrality metrics have their respective weakness. For instance, edge betweenness only includes the contribution of shortest paths, neglecting those from other longer paths; spanning edge centrality fails to distinguish any pair of cut edges; while the computational complexity for current-flow centrality is high. In addition, for weighted networks these measures either do not apply, or have prohibitively high computational cost.

To overcome the weakness of above mentioned measures for edge centrality, a new index called the Kirchhoff edge centrality was proposed very recently [Li and Zhang, 2018]. Since the criteria of edge importance are dependent on real problems or applications [You *et al.*, 2017], the Kirchhoff edge centrality was defined according to the Kirchhoff index [Klein and Randić, 1993], the sum of effective resistances over all pairs of nodes in the network. The Kirchhoff index is a relevant quantity in many real scenarios and has found wide applications. For example, it has been applied to measure the overall connectedness of a network [Tizghadam and Leon-Garcia, 2010] and the robustness of first-order consensus algorithm in noisy networks [Patterson and Bamieh, 2014]. In contrast to the aforementioned measures, the Kirchhoff edge centrality is more discriminating and is provably approximated in nearly linear time [Li and Zhang, 2018]. However, the definition of the Kirchhoff edge centrality involves a parameter θ, how to select θ is still not solved. Moreover, the computation for the Kirchhoff edge centrality needs computing Schur complements of a matrix, the implementation of

[*]Corresponding author. This work was supported by the National Natural Science Foundation of China under Grant No. 11275049.

which is not available. Therefore, there exists a gap between the performance framework for the Kirchhoff edge centrality and an efficient implementation.

In this paper, based on the popular notion of the Kirchhoff index, we propose a novel edge centrality measure for a weighted network by considering it as an electrical network with the resistance of any edge being the reciprocal of the weight in the original weighted networks. Given an edge e, its centrality is defined as the rate at which the Kirchhoff index changes with respect to the change of resistance associated with the edge: The larger the change of the Kirchhoff index is, the more important the edge e is. We demonstrate that the centrality of an edge can be represented in terms of the biharmonic distance between its end nodes, and thus call it biharmonic distance related centrality (BDRC). We show that BDRC is more discriminating than those frequently used metrics, for example, edge betweenness and spanning edge centrality. We also provide a fast algorithm to compute approximate biharmonic distance for all edges in nearly linear time. Finally, we experimentally show that our algorithm is accurate and is significantly faster than the direct computation of the biharmonic distance according to its definition.

2 Related Works

Here we briefly review some edge centrality metrics and their related algorithms, especially the computational complexity of these algorithms.

Edge betweenness is one of the popular edge centrality measures. The betweenness of an edge is the probability of a shortest path between two nodes passing through the edge. Brandes presented an algorithm computing the exactly edge betweenness [Brandes, 2001]. Given a network with n nodes and m edges, the computational cost of this algorithm is $O(nm)$ for unweighted networks, and $O(nm + n^2 \log n)$ for weighted networks, which is computationally expensive. Afterwards, some approximate approaches were developed to speed up the computation [Bader *et al.*, 2007; Brandes and Pich, 2007; Geisberger *et al.*, 2008]. However, these approximate algorithms provide no approximation guarantees.

Spanning edge centrality is another important measure for edge importance proposed in [Teixeira *et al.*, 2013]. For any edge in a network, its spanning edge centrality is the probability that the edge is included in a uniformly chosen spanning tree of the network. Thus far, the best exact algorithm has complexity $O(mn^{3/2})$. Recently two approximation algorithms have been proposed for the purpose of computing spanning edge centrality of massive networks [Mavroforakis *et al.*, 2015; Hayashi *et al.*, 2016], the accuracy of which have theoretical guarantees.

A third edge centrality measure is current-flow centrality that was first introduced in [Brandes and Fleischer, 2005]. By definition, if an edge participates in lots of short paths between node pairs, then it is relatively important with respect to other edges. An algorithm with time complexity $O(mn^{3/2} \log n)$ was designed for computing current-flow centrality in [Brandes and Fleischer, 2005]. Another algorithm with a lower complexity $O(mn \log n)$ was proposed in [Mavroforakis *et al.*, 2015].

Very recently, the Kirchhoff edge centrality was proposed [Li and Zhang, 2018], which, together with spanning edge centrality and current-flow centrality, belongs to the class of electrical centrality measures, since they all have a direct connection with effective resistance. In comparison with the first three centrality measures, Kirchhoff edge centrality is more discriminating and can be provably approximated in nearly-linear time. However, the Kirchhoff edge centrality involves determining a parameter θ, and there is no criterion for choosing θ. Moreover, algorithms in [Li and Zhang, 2018] are only theoretically good, their overhead, especially the θ-dependent factors has much room for improvement. Finally, the algorithms are only in theory stage, it is still a task to bring theory into practice.

In this paper, based on the Kirchhoff index, we propose a novel measure for edge centrality for weighted networks, called biharmonic distance related centrality (BDRC), since it is dependent on biharmonic distance, as will be shown below. The BDRC has a better discriminating power, and can be efficiently approximated with a high probability.

3 Preliminaries

In this section, we give a brief introduction to some basic concepts about graphs, electrical networks and biharmonic distance.

3.1 Graph and Laplacian Matrix

Let $\mathcal{G} = (V, E, w)$ be a connected undirected weighted graph (network), where V is the node set, E is the edge set, and $w : E \to \mathbb{R}_+$ is the positive edge weight function, with w_e representing the weight of edge e. Then, there are $n = |V|$ nodes and $m = |E|$ edges in $\mathcal{G}$. We use $u \sim v$ to indicate that nodes u and v are adjacent. Let $w_{\max} = \max_{e \in E} w_e$ and $w_{\min} = \min_{e \in E} w_e$.

Let $\boldsymbol{B} \in \mathbb{R}^{|E| \times |V|}$ be the incidence matrix of $\mathcal{G}$. For each edge e connecting two nodes u and v, a direction is assigned arbitrarily. Let $\boldsymbol{b}_e^\top$ be the row of matrix $\boldsymbol{B}$ corresponding to edge e, then $b_{eu} = 1$ if node u is the tail of edge e, $b_{eu} = -1$ if node u is the head of edge e, and $b_{eu} = 0$ otherwise. $\boldsymbol{b}_e$ can also be written as $\boldsymbol{b}_e = \boldsymbol{e}_u - \boldsymbol{e}_v$, where $\boldsymbol{e}_u$, $u \in V$, is the uth canonical basis of the space $\mathbb{R}^{|V|}$. Let $\boldsymbol{W} \in \mathbb{R}^{|E| \times |E|}$ be a diagonal matrix with the (e, e)th entry being w_e. Then the Laplacian matrix $\boldsymbol{L}$ of graph $\mathcal{G}$ can be written as $\boldsymbol{L} = \boldsymbol{B}^T \boldsymbol{W} \boldsymbol{B} = \sum_{e \in E} w_e \boldsymbol{b}_e \boldsymbol{b}_e^\top$.

$\boldsymbol{L}$ is symmetric and positive semidefinite, which means $\boldsymbol{L}$ has a spectral decomposition as $\boldsymbol{L} = \sum_{i=1}^{n-1} \lambda_i u_i u_i^\top$, where $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-1}$ are its $n-1$ positive eigenvalues, and $u_i, i = 1, 2, \ldots, n-1$ are the corresponding mutually orthogonal unit eigenvectors. We denote by $\boldsymbol{L}^\dagger$ the pseudoinverse of $\boldsymbol{L}$, and $\boldsymbol{L}^{2\dagger} = (\boldsymbol{L}^\dagger)^2$. Then $\boldsymbol{L}^\dagger$ can be recast as $\boldsymbol{L}^\dagger = \sum_{i=1}^{n-1} \frac{1}{\lambda_i} u_i u_i^\top$. Thus, $\ker(\boldsymbol{L}) = \ker(\boldsymbol{L}^\dagger)$. Let $\boldsymbol{1}_n$ be the n-dimension vector with all entries being 1, let $\boldsymbol{I}$ be the $n \times n$ identity matrix, and let $\boldsymbol{0}$ be the $n \times n$ zero matrix. Define $\boldsymbol{J} = \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^\top$. Then, we have the following relations: $\boldsymbol{L}^\dagger \boldsymbol{L} = \boldsymbol{L} \boldsymbol{L}^\dagger = \boldsymbol{I} - \boldsymbol{J}$, $(\boldsymbol{I} - \boldsymbol{J})^2 = \boldsymbol{I} - \boldsymbol{J}$, $\boldsymbol{J}^2 = \boldsymbol{J}$, and $\boldsymbol{J} \boldsymbol{L} = \boldsymbol{L} \boldsymbol{J} = \boldsymbol{0}$.

3.2 Electrical Networks and Resistance Distance

For any graph $\mathcal{G} = (V, E, w)$, we can define an electrical network $G = (V, E, r)$ by considering edges as resistors and considering nodes as junctions between resistors. The resistor of an associated edge e is $r_e = w_e^{-1}$. The resistance distance $d_R(u, v)$ between two nodes u and v in graph $\mathcal{G}$ is defined as the effective resistance between u and v in electrical network G. Specifically, $d_R(u, v)$ is defined as the potential difference between u and v when a unit current is injected to u and extracted from v. Let $\mathbf{i} \in \mathbb{R}^{|E|}$ represent the current across all resistors, and let $\mathbf{v} \in \mathbb{R}^{|V|}$ represent the voltages at all junctions. By Kirchoff's law, $\boldsymbol{B}^\top \mathbf{i} = \mathbf{c}$, where $\mathbf{c} \in \mathbb{R}^{|V|}$ denotes the external currents injected at all junctions, and by Ohm's law, $\boldsymbol{W}^{-1}\mathbf{i} = \boldsymbol{B}\mathbf{v}$. It follows that $\boldsymbol{L}\mathbf{v} = \mathbf{c}$. By the definition of resistance distance, we get

$$d_R(u, v) = \boldsymbol{L}^\dagger_{uu} + \boldsymbol{L}^\dagger_{vv} - 2\boldsymbol{L}^\dagger_{uv}\,, \tag{1}$$

which is the standard definition for resistance distance [Klein and Randić, 1993].

Definition 3.1. For a graph $\mathcal{G}$, its Kirchhoff index $R(\mathcal{G})$ is defined as the sum of resistance distance over all its node pairs. That is,

$$R(\mathcal{G}) = \sum_{\substack{u,v \in V \\ u<v}} d_R(u, v)\,. \tag{2}$$

It is easy to verify that $R(\mathcal{G}) = n\mathrm{Tr}\left(\boldsymbol{L}^\dagger\right)$.

3.3 Biharmonic Distance

The notion of biharmonic distance was first proposed in [Lipman *et al.*, 2010] as a measure of distance between two points on a curve surface.

Definition 3.2. For a graph $\mathcal{G}$, the biharmonic distance $d_B(u, v)$ between two nodes u and v is defined by

$$d_B^2(u, v) = \boldsymbol{L}^{2\dagger}_{uu} + \boldsymbol{L}^{2\dagger}_{vv} - 2\boldsymbol{L}^{2\dagger}_{uv}\,. \tag{3}$$

Biharmonic distance is a metric, which satisfies the following properties: non-negativity, nullity, symmetry, and triangle inequality [Lipman *et al.*, 2010]. It has some advantages over resistance distance and geodesic distance in some realistic contexts, e.g., computer graphics. Moreover, biharmonic distance can be used to measure the robustness of the second-order noisy consensus problem without leaders [Bamieh *et al.*, 2012]. Below we will show that biharmonic distance is also related to edge centrality in weighted networks.

4 Biharmonic Distance Related Edge Centrality

Generally, the criterion of edge criticality for a graph depends on particular applications. In many realistic contexts of a graph [Tizghadam and Leon-Garcia, 2010; Patterson and Bamieh, 2014], their performance is determined by the Kirchhoff index of the graph, which is the total effective resistance of the corresponding electrical network. Since the Kirchhoff index for a graph is a function of resistances of all resistors in the associated electrical network, the importance of an edge can be captured by the change of the Kirchhoff index with respect to the modification of resistor for the corresponding edge. As shown below, the change of the Kirchhoff index is closely related to the biharmonic distance, and is thus called biharmonic distance related centrality (BDRC).

Definition 4.1. For a graph $\mathcal{G} = (V, E, w)$ or its corresponding electrical network $G = (V, E, r)$, let $e = (u, v)$ be an edge with weight w_e or resistance $r_e = 1/w_e$. The biharmonic distance related centrality $\mathcal{C}(e)$ for edge $e = (u, v)$ is defined by

$$\mathcal{C}(e) := \frac{\partial R(\mathcal{G})}{\partial r_e}. \tag{4}$$

The BDRC $\mathcal{C}(e)$ is the rate at which $R(\mathcal{G})$ changes with regard to the change of resistance corresponding to edge $e = (u, v)$. Therefore, this edge centrality index quantifies the influence of an edge on the performance of the network characterized by the Kirchhoff index. The larger the change $\mathcal{C}(e)$ of $R(\mathcal{G})$, the more important the edge e.

An advantage of BDRC is that it can be explicitly determined.

Theorem 4.2. *For an edge $e = (u, v)$ in graph $\mathcal{G} = (V, E, w)$, its BDRC can by expressed by*

$$\mathcal{C}(e) = n w_e^2\, d_B^2(u, v)\,. \tag{5}$$

Proof. By definition, we have

$$\mathcal{C}(e) = \frac{\partial R(\mathcal{G})}{\partial r(e)} = n\frac{\partial \mathrm{Tr}\left(\boldsymbol{L}^\dagger\right)}{\partial w_e}\frac{\mathrm{d}w_e}{\mathrm{d}r_e} = n\frac{\partial \mathrm{Tr}\left(\boldsymbol{L}^\dagger + \boldsymbol{J}\right)}{\partial w_e}\frac{\mathrm{d}w_e}{\mathrm{d}r_e}$$
$$= n\frac{\partial \mathrm{Tr}\left((\boldsymbol{L}+\boldsymbol{J})^{-1}\right)}{\partial w_e}\frac{\mathrm{d}\left(r_e^{-1}\right)}{\mathrm{d}r_e} = \frac{n}{r_e^2}\mathrm{Tr}\left((\boldsymbol{L}+\boldsymbol{J})^{-2}\frac{\partial\left(\boldsymbol{L}+\boldsymbol{J}\right)}{\partial w_e}\right).$$

On the other hand,

$$\frac{\partial\left(\boldsymbol{L}+\boldsymbol{J}\right)}{\partial w_e} = \frac{\partial \boldsymbol{L}}{\partial w_e} = \frac{\partial\left(\sum_{e\in E} w_e \boldsymbol{b}_e \boldsymbol{b}_e^\top\right)}{\partial w_e} = \boldsymbol{b}_e\boldsymbol{b}_e^\top\,.$$

Then, we can derive that

$$\begin{aligned}\mathcal{C}(e) &= n r_e^{-2}\boldsymbol{b}_e^\top(\boldsymbol{L}^2 + \boldsymbol{J}^2 + \boldsymbol{L}\boldsymbol{J} + \boldsymbol{J}\boldsymbol{L})^{-1}\boldsymbol{b}_e \\ &= n w_e^2 \boldsymbol{b}_e^\top(\boldsymbol{L}^2 + \boldsymbol{J})^{-1}\boldsymbol{b}_e \\ &= n w_e^2 \boldsymbol{b}_e^\top \boldsymbol{L}^{2\dagger}\boldsymbol{b}_e = n w_e^2\, d_B^2(u, v)\,.\end{aligned} \tag{6}$$

This completes the proof. □

Theorem 4.2 indicates that the BDRC $\mathcal{C}(e)$ of an edge $e = (u, v)$ is simultaneously determined by its weight w_e and the biharmonic distance $d_B(u, v)$ between the end nodes u and v of edge e. Thus, the BDRC $\mathcal{C}(e)$ reflects both the structural and dynamical importance of edge e.

5 Comparison with Different Measures

In this section, we show that the BDRC has a more discriminating power than other measures, e.g., edge betweenness centrality and spanning edge centrality.

Let us consider the two edges e_1 and e_2 in the disturbed ring graph in Figure 1. By intuition, e_1 is more important

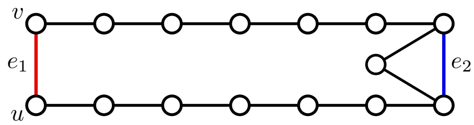

Figure 1: A disturbed ring graph. It is obtained from a 15-node ring, by adding an edge connecting two nodes with distance 2.

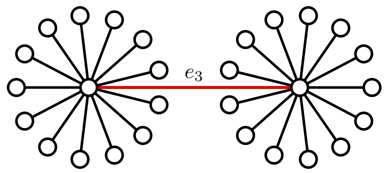

Figure 2: A double-star graph.

than e_2. This can be understood from the following arguments If we delete e_1, the shortest path length between u and v increases by 12; if we remove e_2, the length of shortest paths between any node pair will increase by at most 1. However, betweenness centrality for e_1 and e_2 are both equal to 24.5, implying the two edges cannot be differentiated by betweenness centrality. According to Theorem 4.2, the BDRC for edges e_1 and e_2 is $\mathcal{C}(e_1) = 1.1327$ and $\mathcal{C}(e_2) = 0.5413$, respectively. This shows that e_2 is relatively less important than e_1, agreeing with our human intuition.

In addition, BDRC also has a better discriminating ability than the spanning edge centrality. Let us consider the edge e_3 and any other edge in the double-star graph shown in Figure 2. Spanning edge centrality cannot distinguish them, since their spanning edge centrality is both equal to 1. By contrast, the BDRC is $\mathcal{C}(e_3) = 7$ for edge e_3, and is 0.9643 for any other edge. Thus, e_3 has a higher score, reflecting our intuition.

In order to further demonstrate the ability of BDRC to differentiate between distinct edges in realistic networks, we also experimentally compare BDRC with three frequently used edge centrality measures, edge betweenness, spanning edge centrality, and current-flow edge centrality. For each metric, we numerically calculate the exact centrality value of each edge for some classic real networks listed in Table 1. Similarly to what has been done in [Bergamini *et al.*, 2016], we evaluate the relative standard deviation for each edge centrality measure, which is defined as the standard deviation divided by the average. Figure 3 reports the relative standard deviation for all measures of edge centrality. It is always higher for BDRC than it is for other three metrics, indicating that BDRC can better distinguish between different edges.

Network	n	m
Lesmis [Knuth, 1993]	77	254
Adjnoun [Newman, 2006]	112	425
Dolphins [Lusseau *et al.*, 2003]	62	159
Celegansneural [Watts and Strogatz, 1998]	297	2 148
High-energy [Newman, 2001]	5 835	13 815
Condensed matter [Newman, 2001]	13 861	44 619

Table 1: Some classic real-world networks.[1]

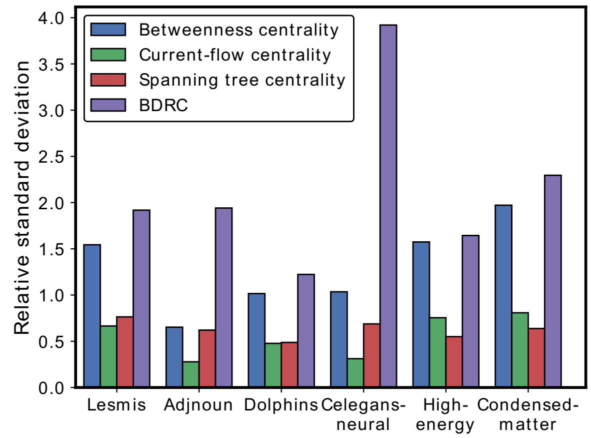

Figure 3: Relative standard deviation for different edge centrality measures.

6 Fast Approximation Algorithm

A straightforward way to calculate BDRC $\mathcal{C}(e)$ of an edge e involves computing the pseudoinverse of $\boldsymbol{L}^2$, which has a complexity of $O(n^3)$, making it infeasible to huge networks. In this section, we propose an algorithm to compute an approximation of $\mathcal{C}(e)$ for all $e \in E$ with a high probability in nearly linear time with respect to the number of edges. Before introducing our algorithm, we recall two lemmas that are critical to proving the approximation factor of our algorithm. In what follows we use the notation $\widetilde{O}(\cdot)$ to hide $\mathrm{poly}(\log n)$ factors.

Lemma 6.1 (Johnson-Lindenstrauss Lemma, [Achlioptas, 2001])**.** *Given fixed vectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ and $\epsilon > 0$, let $\boldsymbol{Q}_{k\times d}$ be a random $\pm 1/\sqrt{k}$ matrix (i.e., independent Bernoulli entries) with $k \geq 24 \log n/\epsilon^2$. Then with probability at least $1 - 1/n$,*

$$(1-\epsilon)\|v_i - v_j\|_2^2 \leq \|\boldsymbol{Q}v_i - \boldsymbol{Q}v_j\|_2^2 \leq (1+\epsilon)\|v_i - v_j\|_2^2$$

for all pairs $i, j \leq n$.

Lemma 6.2 ([Spielman and Teng, 2004])**.** *There is an algorithm $x = \texttt{LaplSolve}(L, y, \delta)$ which takes a Laplacian matrix $\boldsymbol{L}$, a column vector y, and an error parameter $\delta > 0$, and returns a column vector x satisfying*

$$\|x - \boldsymbol{L}^{\dagger}y\|_{\boldsymbol{L}} \leq \delta\|\boldsymbol{L}^{\dagger}y\|_{\boldsymbol{L}},$$

where $\|y\|_{\boldsymbol{L}} = \sqrt{y^T \boldsymbol{L} y}$. The algorithm runs in expected time $\widetilde{O}\left(m \log(1/\delta)\right)$.

From (3), for any pair of nodes u and v in $\mathcal{G}$, $d_B^2(u,v)$ can be rewritten as $d_B^2(u,v) = \|\boldsymbol{L}^{\dagger}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2$. Thus, $d_B(u,v) = \|\boldsymbol{L}^{\dagger}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|$, which is actually the Euclidean distance between vectors $\boldsymbol{L}^{\dagger}\boldsymbol{e}_u$ and $\boldsymbol{L}^{\dagger}\boldsymbol{e}_v$. Then, by projecting vectors $\boldsymbol{L}^{\dagger}\boldsymbol{e}_v$, $v \in V$, into a subspace spanned by $k =$

[1]All data can be found at http://www-personal.umich.edu/˜mejn/netdata/

$O(\log n)$-dimensional random vectors and using LaplSolve, we can introduce a fast algorithm to approximate the biharmonic distance $d_B(u, v)$ for each edge $e \in E$.

Let $\boldsymbol{Q}$ be a $k \times n$ random projection matrix. By Lemma 6.1, $\boldsymbol{Q}\boldsymbol{L}^{\dagger}(\boldsymbol{e}_u - \boldsymbol{e}_v)$ is a good approximation for $d_B^2(u, v)$. However, computing $\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{L}^{\dagger}$ directly involves calculating $\boldsymbol{L}^{\dagger}$. We avoid this by solving the system of equations $\boldsymbol{L}\boldsymbol{z}_i = \boldsymbol{q}_i$, $i = 1, \dots, k$, where $\boldsymbol{z}_i^{\top}$ and $\boldsymbol{q}_i^{\top}$ are the ith row of $\boldsymbol{Z}$ and $\boldsymbol{Q}$, respectively. Lemma 6.2 shows that $\boldsymbol{z}_i^{\top}$ can be efficiently approximated by using LaplSolve.

Lemma 6.3. *Given a $k \times n$ matrix $\boldsymbol{Z}$ that satisfies*

$$(1-\epsilon) d_B^2(u, v) \leq \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2 \leq (1+\epsilon) d_B^2(u, v),$$

for every pair $u, v \in V$. If for all $u \in V$, let $\boldsymbol{z}_u = \boldsymbol{Z}\boldsymbol{e}_u$, and let $\tilde{\boldsymbol{z}}_u$ be an approximation of $\boldsymbol{z}_u$, satisfying

$$\|\boldsymbol{z}_u - \tilde{\boldsymbol{z}}_u\|_{\boldsymbol{L}} \leq \delta \|\boldsymbol{z}_u\|_{\boldsymbol{L}}, \tag{7}$$

where

$$\delta \leq \frac{\epsilon}{3} \sqrt{\frac{6(1-\epsilon) w_{\min}}{(1+\epsilon) n^5 w_{\max}^2}}. \tag{8}$$

Then for any pair of nodes u and v belonging to V,

$$(1-\epsilon)^2 d_B^2(u, v) \leq \|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2 \leq (1+\epsilon)^2 d_B^2(u, v). \tag{9}$$

Proof. To prove (9), it suffices to show that for any pair of nodes u and v,

$$\begin{aligned}
&\left| \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2 - \|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2 \right| \\
&= \left| \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| - \|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| \right| \times \\
&\quad \left| \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| + \|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| \right| \\
&\leq \left(\frac{2\epsilon}{3} + \frac{\epsilon^2}{9} \right) \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2,
\end{aligned}$$

which is satisfied if

$$\left| \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| - \|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| \right| \leq \frac{\epsilon}{3} \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|. \tag{10}$$

We now prove that (10) is true. Note that in a connected graph $\mathcal{G}$, there is a simple path P between any two nodes u and v. Applying the triangle inequality along P, we obtain

$$\begin{aligned}
&\left| \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| - \left\|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\right\| \right| \\
&\leq \left\| (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})(\boldsymbol{e}_u - \boldsymbol{e}_v) \right\| \leq \sum_{a \sim b \in P} \left\| (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})(\boldsymbol{e}_a - \boldsymbol{e}_b) \right\|.
\end{aligned}$$

The square of the last sum term can be evaluated as:

$$\begin{aligned}
&\left(\sum_{a \sim b \in P} \left\| (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})(\boldsymbol{e}_a - \boldsymbol{e}_b) \right\| \right)^2 \\
&\leq n \sum_{a \sim b \in P} \left\| (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})(\boldsymbol{e}_a - \boldsymbol{e}_b) \right\|^2 \\
&\leq n \sum_{a \sim b \in E} \left\| (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})(\boldsymbol{e}_a - \boldsymbol{e}_b) \right\|^2 \\
&= n \left\| (\boldsymbol{Z} - \tilde{\boldsymbol{Z}}) \boldsymbol{B}^{\top} \right\|_F^2 = n \left\| \boldsymbol{B} (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})^{\top} \right\|_F^2 \\
&\leq \frac{n}{w_{\min}} \left\| \boldsymbol{W}^{1/2} \boldsymbol{B} (\boldsymbol{Z} - \tilde{\boldsymbol{Z}})^{\top} \right\|_F^2 \leq \frac{n\delta^2}{w_{\min}} \left\| \boldsymbol{W}^{1/2} \boldsymbol{B} \boldsymbol{Z}^T \right\|_F^2,
\end{aligned}$$

where the first inequality follows from Cauchy-Schwarz inequality, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and the last inequality follows from (7) and can be further evaluated as

$$\begin{aligned}
&\frac{n\delta^2}{w_{\min}} \left\| \boldsymbol{W}^{1/2} \boldsymbol{B} \boldsymbol{Z}^T \right\|_F^2 = \frac{n\delta^2}{w_{\min}} \sum_{a \sim b \in E} w_{a \sim b} \|\boldsymbol{Z}(\boldsymbol{e}_a - \boldsymbol{e}_b)\|^2 \\
&\leq \delta^2 \frac{n(1+\epsilon)}{w_{\min}} \mathrm{Tr}\left(\boldsymbol{L}^{\dagger}\right) \leq \frac{\delta^2 n(n^2-1)(1+\epsilon)}{6 w_{\min}}.
\end{aligned}$$

The last inequality follows by the fact that $\mathrm{Tr}(\boldsymbol{L}^{\dagger}) = R(\mathcal{G})/n$, where $R(\mathcal{G})$ achieves the maximum value $n(n^2-1)/6$ when $\mathcal{G}$ is an n-node path graph. On the other hand,

$$\begin{aligned}
&\|Z(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2 \geq (1-\epsilon) d_B^2(u, v) = (1-\epsilon) \boldsymbol{b}_e^{\top} \boldsymbol{L}^{2\dagger} \boldsymbol{b}_e \\
&\geq (1-\epsilon)(\lambda_{n-1})^{-2} \geq (1-\epsilon)(n w_{\max})^{-2}.
\end{aligned} \tag{11}$$

The last inequality in (11) is due to the fact that $w_{\max} \boldsymbol{L}_{\mathcal{K}_n} - \boldsymbol{L}_{\mathcal{G}}$ is positive semidefinite, while $\boldsymbol{L}_{\mathcal{K}_n}$ is the Laplacian of an n-node clique whose edge weights are all equal to 1. Thus,

$$\begin{aligned}
&\frac{\left| \|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\| - \left\|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\right\| \right|}{\|\boldsymbol{Z}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|} \\
&\leq \delta \left(\frac{n(n^2-1)(1+\epsilon)}{6 w_{\min}} \right)^{1/2} \left(\frac{n^2 w_{\max}^2}{1-\epsilon} \right)^{1/2} \leq \frac{\epsilon}{3},
\end{aligned}$$

where δ is given by (8). □

Lemma 6.3 leads to the following theorem.

Theorem 6.4. *There is an $\widetilde{O}(m \log c/\epsilon^2)$ time algorithm, which inputs $0 < \epsilon < 1$ and $\mathcal{G} = (V, E, w)$ where $c = \frac{w_{\max}^2}{w_{\min}}$, and returns a $(24 \log n/\epsilon^2) \times n$ matrix $\tilde{\boldsymbol{Z}}$ such that with probability at least $1 - 1/n$,*

$$(1-\epsilon)^2 d_B^2(u, v) \leq \|\tilde{\boldsymbol{Z}}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|^2 \leq (1+\epsilon)^2 d_B^2(u, v)$$

for any pair of nodes $u, v \in V$.

Based on Theorem 6.4, we propose an algorithm AppxBDRC to approximately compute $\mathcal{C}(e)$ for all edges $e \in E$, the pseudocode of which is shown in Algorithm 1.

Algorithm 1: AppxBDRC($\mathcal{G}, \epsilon$)

Input : G: a connected undirected graph.
ϵ: approximation parameter of edge centrality
Output : $\hat{C} = \{e, \hat{c}(e) | e \in E\}$: $\hat{c}$ is an approximation of $\mathcal{C}(e)$, the BDRC of e.

$\boldsymbol{L}$ = Laplacian of $\mathcal{G}$, $\hat{c}(e)$=0 for all $e \in E$
$k = \lceil 24 \log n/\epsilon^2 \rceil$
for $i = 1$ *to* k **do**
  Construct a $\pm 1/\sqrt{k}$ random vector $\boldsymbol{q}_i$ of size $n \times 1$
  $\tilde{\boldsymbol{z}}_i$ = LaplSolve($\boldsymbol{L}, \boldsymbol{q}_i, \delta$) where δ is given by (8)
  for *each* $e \in E$ **do**
    $\hat{c}(e) = \hat{c}(e) + n w_e^2 |\tilde{\boldsymbol{z}}_{i,u} - \tilde{\boldsymbol{z}}_{i,v}|^2$
return $\hat{C} = \{e, \hat{c}(e) | e \in E\}$

Network	# nodes in LCC	# edges in LCC	ExactBDRC (s)	AppxBDRC (s) with various ϵ				
				0.3	0.25	0.2	0.15	0.1
Chicago	823	822	0.1655	0.0496	0.0692	0.1067	0.1870	0.4184
Facebook (NIPS)	2 888	2 981	6.9439	0.6099	0.8944	1.4355	2.5606	5.4581
Vidal	2 783	6 007	6.2797	1.5683	2.2292	3.4950	6.1910	13.950
Powergrid	4 941	6 594	34.851	3.8330	5.7525	9.0637	16.826	40.202
Reactome	5 973	145 778	61.673	18.916	32.787	44.870	74.794	176.13
Route views	6 474	12 572	78.039	3.1174	4.0897	6.9604	11.849	27.826
Pretty Good Privacy	10 680	24 316	272.87	10.622	15.451	23.832	38.945	90.053
Astro-ph	17 903	196 972	1648.6	51.266	71.565	116.55	202.10	457.30
CAIDA	26 475	53 381	7396.0	19.173	25.823	43.78	73.903	158.43
Brightkite	56 739	212 945	35063	103.27	149.25	226.34	393.76	935.14
Livemocha*	104 103	2 193 083	–	3078.2	3226.5	3558.8	4131.1	6114.4
WordNet*	145 145	656 230	–	294.80	373.24	493.66	776.26	1818.4
Gowalla*	196 591	950 327	–	612.67	777.28	1005.7	1518.1	3029.5
Amazon*	334 863	925 872	–	1307.5	1749.1	2570.3	4648.9	10517
Pennsylvania*	1 087 562	1 541 514	–	5314.5	7288.3	11404	20410	45560

Table 2: The running time (seconds, s) of ExactBDRC and ApExactBDRC with various ϵ on several real-world networks

7 Experimental Evaluation

In this section, we experimentally evaluate the efficiency and accuracy of our approximation algorithm.

We evaluate the algorithm on a large set of real-world networks from different domains. The data of these networks are taken from the Koblenz Network Collection [Kunegis, 2013]. We run our experiments on the largest connected components (LCC) of these networks, related information of which is shown in Table 2.

We run all the experiments on a Linux box with an *Intel i7-7700K @ 4.2-GHz (4 Cores)* and with *32GB* memory. We implement the algorithm AppxBDRC in *Julia v0.6.0*, where the LaplSolve is from [Kyng and Sachdeva, 2016], the *Julia language* implementation of which is accessible on the website[2].

To demonstrate the efficiency of our approximation algorithm AppxBDRC, in Table 2, we compare the running time of AppxBDRC with that of the accurate algorithm called ExactBDRC that calculates biharmonic distance by inverting $\boldsymbol{L}^2 + \boldsymbol{J}$ and computing the BDRC of every edge according to (6). To objectively evaluate the running time of the algorithms, for both ExactBDRC and AppxBDRC on all considered networks excluding the last five ones marked with $*$, we enforce the program to run on a single thread. The results show that for moderate ϵ, AppxBDRC is significantly faster than ExactBDRC, especially for large networks. For the last five networks with node number ranging from 10^5 to 10^6, we cannot run the ExactBDRC algorithm due to memory limit and high time cost. In contrast, for these networks, we approximately compute BDRC for all edges. This further show that AppxBDRC is efficient and scalable, which is suitable for large networks.

In addition to the efficiency, we also evaluate the accuracy of the approximation algorithm AppxBDRC. To this end, we compare the approximate result of AppxBDRC with the exact result calculated by ExactBDRC. In Table 3, we provide the mean relative error σ of our approximation algorithm, where σ is defined as $\sigma = \frac{1}{m}\sum_{e\in E} |\mathcal{C}(e) - \hat{c}(e)|/\mathcal{C}(e)$. The results show that the actual mean relative errors for all ϵ and all networks are insignificant, which are magnitudes smaller than the theoretical guarantee. Thus, the approximation algorithm AppxBDRC leads to very accurate results in practice.

Table 3: Mean relative errors of AppxBDRC with various ϵ.

Network	Mean relative error with various ϵ		
	0.3	0.2	0.1
Chicago	2.75×10^{-2}	1.80×10^{-2}	9.01×10^{-3}
Facebook (NIPS)	2.46×10^{-2}	1.62×10^{-2}	8.14×10^{-3}
Vidal	2.42×10^{-2}	1.64×10^{-2}	7.97×10^{-3}
Powergrid	2.35×10^{-2}	1.57×10^{-2}	8.00×10^{-3}
Reactome	2.36×10^{-2}	1.58×10^{-2}	7.75×10^{-3}
Route views	2.27×10^{-2}	1.54×10^{-2}	7.93×10^{-3}
Pretty Good Privacy	2.26×10^{-2}	1.52×10^{-2}	7.50×10^{-3}
Astro-ph	2.18×10^{-2}	1.47×10^{-2}	7.36×10^{-3}
CAIDA	2.17×10^{-2}	1.45×10^{-2}	7.24×10^{-3}
Brightkite	2.08×10^{-2}	1.39×10^{-2}	6.97×10^{-3}

8 Conclusion

In this paper, we introduced a definition of centrality to measure the importance of edges in weighted networks. For every edge, this new centrality is defined as the sensitivity of the Kirchhoff index on the modification of resistance associated with the edge. We derived that for any edge, its centrality is proportional to the square of the product of its weight and the biharmonic distance between its two endnodes, which is thus called biharmonic distance related centrality (BDRC). We showed that the BDRC measure is more discriminating than edge betweenness and spanning tree centrality. Moreover, we provided an approximation algorithm with probabilistic guarantee, which computes BDRC for all edges in a network in nearly-linear time. Extensive experiments demonstrated that the proposed algorithm is both efficient and accurate in large networks.

It deserves to mention that the proposed edge centrality can be easily extended to the node level, by defining the centrality of a node as the sum of centrality of all edges incident to it. Moreover, the approximate method for computing biharmonic distance can be applied to related areas, such as computer graphics [Lipman *et al.*, 2010] and second-order noisy consensus dynamics without leaders [Bamieh *et al.*, 2012].

[2] http://danspielman.github.io/Laplacians.jl/latest/

References

[Achlioptas, 2001] D. Achlioptas. Database-friendly random projections. In *PODS*, pages 274–281. ACM, 2001.

[Bader *et al.*, 2007] D. A. Bader, S. Kintali, K. Madduri, and M. Mihail. Approximating betweenness centrality. In *WAW*, volume 4863, pages 124–137. Springer, 2007.

[Bamieh *et al.*, 2012] B. Bamieh, M. R. Jovanovic, P. Mitra, and S. Patterson. Coherence in large-scale networks: Dimension-dependent limitations of local feedback. *IEEE Transactions on Automatic Control*, 57(9):2235–2249, 2012.

[Bergamini *et al.*, 2016] E. Bergamini, M. Wegner, D. Lukarski, and H. Meyerhenke. Estimating current-flow closeness centrality with a multigrid Laplacian solver. In *CSC*, pages 1–12. SIAM, 2016.

[Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):28–37, 2001.

[Bienstock *et al.*, 2014] D. Bienstock, M. Chertkov, and S. Harnett. Chance-constrained optimal power flow: Risk-aware network control under uncertainty. *SIAM Review*, 56(3):461–495, 2014.

[Boldi and Vigna, 2014] P. Boldi and S. Vigna. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262, 2014.

[Brandes and Fleischer, 2005] U. Brandes and D. Fleischer. Centrality measures based on current flow. In *STACS*, volume 3404, pages 533–544. Springer, 2005.

[Brandes and Pich, 2007] U. Brandes and C. Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, 17(07):2303–2318, 2007.

[Brandes, 2001] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[Ding *et al.*, 2011] L. Ding, D. Steil, B. Dixon, A. Parrish, and D. Brown. A relation context oriented approach to identify strong ties in social networks. *Knowledge-Based Systems*, 24(8):1187–1195, 2011.

[Geisberger *et al.*, 2008] R. Geisberger, P. Sanders, and D. Schultes. Better approximation of betweenness centrality. In *ALENEX*, pages 90–100. ACM, 2008.

[Girvan and Newman, 2002] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[Hayashi *et al.*, 2016] T. Hayashi, T. Akiba, and Y. Yoshida. Efficient algorithms for spanning tree centrality. In *IJCAI*, pages 3733–3739, 2016.

[Klein and Randić, 1993] D. J. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1):81–95, 1993.

[Knuth, 1993] D. E. Knuth. *The Stanford GraphBase: a Platform for Combinatorial Computing*, volume 37. Addison-Wesley Reading, 1993.

[Kunegis, 2013] J. Kunegis. Konect: The koblenz network collection. In *WWW*, pages 1343–1350. ACM, 2013.

[Kyng and Sachdeva, 2016] R. Kyng and S. Sachdeva. Approximate gaussian elimination for Laplacians - fast, sparse, and simple. In *FOCS*, pages 573–582. IEEE, 2016.

[Li and Zhang, 2018] H. Li and Z. Zhang. Kirchhoff index as a measure of edge centrality in weighted networks: Nearly linear time algorithms. In *SODA*, pages 2377–2396. SIAM, 2018.

[Lipman *et al.*, 2010] Y. Lipman, R. M Rustamov, and T. A Funkhouser. Biharmonic distance. *ACM Transactions on Graphics*, 29(3):27, 2010.

[Lusseau *et al.*, 2003] D. Lusseau, K. Schneider, O. J Boisseau, P. Haase, E. Slooten, and S. M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[Mavroforakis *et al.*, 2015] C. Mavroforakis, R. Garcia-Lebron, I. Koutis, and E. Terzi. Spanning edge centrality: Large-scale computation and applications. In *WWW*, pages 732–742. ACM, 2015.

[Newman, 2001] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[Newman, 2006] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[Patterson and Bamieh, 2014] S. Patterson and B. Bamieh. Consensus and coherence in fractal networks. *IEEE Transactions on Control of Network Systems*, 1(4):338–348, 2014.

[Spielman and Teng, 2004] D. A. Spielman and S. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*, pages 81–90. ACM, 2004.

[Teixeira *et al.*, 2013] A. S. Teixeira, P. T Monteiro, J. A. Carriço, M. Ramirez, and A. P Francisco. Spanning edge betweenness. In *MLG*, volume 24, pages 27–31. ACM, 2013.

[Tizghadam and Leon-Garcia, 2010] A. Tizghadam and A. Leon-Garcia. Autonomic traffic engineering for network robustness. *IEEE Journal on Selected Areas in Communications*, 28(1):39–50, 2010.

[Watts and Strogatz, 1998] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[White and Smyth, 2003] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *KDD*, pages 266–275. ACM, 2003.

[You *et al.*, 2017] K. You, R. Tempo, and L. Qiu. Distributed algorithms for computation of centrality measures in complex networks. *IEEE Transactions on Automatic Control*, 62(5):2080–2094, 2017.