Power-Law Graphs Have Minimal Scaling of Kemeny Constant for Random Walks

Wanyue Xu
Fudan University
Shanghai, China
xuwy@fudan.edu.cn

Yibin Sheng
Fudan University
Shanghai, China
14210240021@fudan.edu.cn

Zuobai Zhang
Fudan University
Shanghai, China
17300240035@fudan.edu.cn

Haibin Kan
Fudan University
Shanghai, China
hbkan@fudan.edu.cn

Zhongzhi Zhang
Fudan University
Shanghai, China
zhangzz@fudan.edu.cn

ABSTRACT

The mean hitting time from a node i to a node j selected randomly according to the stationary distribution of random walks is called the Kemeny constant, which has found various applications. It was proved that over all graphs with N vertices, complete graphs have the exact minimum Kemeny constant, growing linearly with N. Here we study numerically or analytically the Kemeny constant on many sparse real-world and model networks with scale-free small-world topology, and show that their Kemeny constant also behaves linearly with N. Thus, sparse networks with scale-free and small-world topology are favorable architectures with optimal scaling of Kemeny constant. We then present a theoretically guaranteed estimation algorithm, which approximates the Kemeny constant for a graph in nearly linear time with respect to the number of edges. Extensive numerical experiments on model and real networks show that our approximation algorithm is both efficient and accurate.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Networks** → **Network algorithms**; • **Theory of computation** → **Graph algorithms analysis**.

KEYWORDS

Random walk, Hitting time, Kemeny constant, Spectral graph theory, Random projection, Linear system solver, Normalized Laplacian matrix

ACM Reference Format:
Wanyue Xu, Yibin Sheng, Zuobai Zhang, Haibin Kan, and Zhongzhi Zhang. 2020. Power-Law Graphs Have Minimal Scaling of Kemeny Constant for Random Walks. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380093

Zhongzhi Zhang is the corresponding author. All the authors are with Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433. Haibin Kan and Zhongzhi Zhang are also with Shanghai Blockchain Engineering Research Center.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380093

1 INTRODUCTION

As a powerful theory and analysis tool, random walks have received considerable attention from the scientific community [12, 25, 36]. The fundamental quantities related to random walks include stationary distribution [43], hitting time [13, 38], mixing time [33], and cover time [11], all of which have found wide applications in various fields [20, 34, 46]. In the aspect of hitting time for a random walk on a graph from one vertex i to another vertex j, it is the expected time for the walker starting from i to visit j for the first time. The hitting time can be used to gauge transmission costs in wireless networks [17, 35], to design clustering algorithm [1, 9], and to measure the importance or centrality of a vertex [47].

In addition to the intrinsic interest of hitting time itself, many other interesting quantities associated with random walks (for example, Kemeny constant) are encoded in or expressed in terms of this fundamental quantity. The Kemeny constant is defined as the expected time for a walker from a vertex to a second vertex chosen randomly from the network according to the stationary distribution of the random walk. The Kemeny constant has many applications in several areas [22]. For example, it can be used to measure the efficiency of user navigation through the World Wide Web [32], where it can be accounted for the mean number of edges the random surfer needs to follow before arriving at the final destination. Again for instance, the Kemeny constant is related to the mixing rate of an irreducible Markov chain [33], by looking upon it as the expected time to mixing of the Markov chain [21, 28]. Moreover, the Kemeny constant is one of the widely used criticality [14, 18, 31] or connectivity [6] measures for a graph. Finally, in recent work the Kemeny constant was applied to gauge the efficiency of robotic surveillance in network environments [2, 40] and characterize the performance of a class of noisy formation control protocols [24].

It is well-established that [27] the Kemeny constant of a graph is determined by all the eigenvalues of the normalized Laplacian matrix of the graph, which is encoded in the topology of the underlying graph. In view of its wide range of applications, the Kemeny constant has been extensively studied [22]. In particular, the Kemeny constant in various networks with different topologies has received considerable interest. For example, previous work has studied the Kemeny constant for various trees, including Cayley

trees [26], tree-like polymer network [54], fractal trees [55], as well as the path graph [39]. In addition, the Kemeny constant for some networks with cycles was also studied, such as weighted Koch networks [41] and extended Sierpiński graphs [42]. Finally, Palacios and Renom [39] proved that among all N-vertex graphs, the minimum of the Kemeny constant is $1 + (N-1)^2/N$, which can be achieved only in the complete graphs. These works show that in different networks the behavior of Kemeny constant is also quite different.

It is well known that the random walk process is strongly affected by the topological properties of the underlying networks [36]. Then, an interesting question arises: of all connected networks, which are the optimal or almost optimal, having the smallest or almost smallest Kemeny constant? It is of practical significance, since it is particularly useful for designing networks with particular performance, such as best robustness and best efficiency of navigation or surveillance performance. It has been verified that among all undirected graphs the Kemeny constant is the least for complete graphs [39], which increases linearly with the network size (number of vertices). Nevertheless, most real-life networks are sparse with constant average degree [37], and simultaneously exhibit the striking scale-free [5] small-world [45] topologies, which have a strong effect on various dynamics on networks, e.g., noisy consensus [51–53] and disease spreading [8, 44]. Until now, the role of scale-free and small-world [45] structure on the Kemeny constant of random walks is still not well understood. On the other hand, direct computation of the Kemeny constant by calculating the eigenvalues of the normalized Laplacian matrix is time-consuming, which is infeasible for large networks, especially those with millions of vertices. Then, another question arises: Is there a fast algorithm for computing the Kemeny constant of a general graph?

In this paper, we study the Kemeny constant for scale-free small-world real-world and model networks. Our main work and contributions are as follows. First, we consider the Kemeny constant of sparse real networks with scale-free small-world properties, where the ratio of the Kemeny constant to the number of nodes approaches a constant. Second, we address the Kemeny constant for two sparse deterministic networks [16, 57] that display the remarkable scale-free small-world properties as observed in real networks [37]. Through the decimation approach, we derive recursive expressions for all eigenvalues of the transition matrices for both networks, and further obtain closed-form formulas for their Kemeny constant, which also grows as a linear function of the network size, displaying an identical scaling as that of complete graphs. Third, we study the Kemeny constant on the Barabási-Albert network [5], and show that the ratio of the Kemeny constant to the node number also tends to constants. Thus, for sparse networks with the scale-free and small-world structural properties, their Kemeny constant is almost smallest. Finally, we present an algorithm to approximately compute the Kemeny constant of a generic graph, the running time of which is nearly linear with respect to the number of edges. We experimentally demonstrate the effectiveness and efficiency of this algorithm on some real networks and the two studied deterministic networks [16, 57].

2 PRELIMINARY

In this section, we introduce some basic concepts in graph theory, random walks on a graph, and some related work for the problem to be studied.

2.1 Graph and Matrix Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with vertex/node set $\mathcal{V}$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, the numbers of vertices and edges in which are $N = |\mathcal{V}|$ and $E = |\mathcal{E}|$, respectively. Then, the total degree of all vertices is $2E$, and the average degree is $\langle d \rangle = (2E)/N$. A graph is said to be simple if it has no loops and parallel edges. Throughout this paper, all graphs considered are finite simple connected graphs, and the terms graph and network are used indistinctly. For a vertex $i \in \mathcal{V}$, let $\mathcal{N}_i = \{x|(x,i) \in \mathcal{E}\}$ denote the set of its neighborhood vertices and let $d_i = |\mathcal{N}_i|$ denote the degree of i.

The N vertices in graph $\mathcal{G}$ are labeled by $1, 2, 3, \ldots, N$, respectively. The adjacency relation between the N vertices is encoded in the adjacency matrix $A = (a_{ij})_{N\times N}$ of $\mathcal{G}$, where $a_{ij} = 1$ if vertices i and j are directly connected by an edge in $\mathcal{G}$, and $a_{ij} = 0$ otherwise. Then, the degree of vertex i is $d_i = \sum_{j=1}^{N} a_{ij}$. Let D denote the diagonal degree matrix of $\mathcal{G}$. The ith diagonal entry of D is d_i, while all other entries are zeros. Then, the Laplacian matrix $\mathcal{L}$ of $\mathcal{G}$ is defined to be $\mathcal{L} = D - A$.

The Laplacian matrix $\mathcal{L}$ of a graph $\mathcal{G}$ can also be expressed in terms of its node-edge incidence matrix $B \in \mathbb{R}^{E\times N}$, which is a signed matrix. The entry $b_{e,v}$, $e \in \mathcal{E}$ and $v \in \mathcal{V}$, of B is defined as follows: $b_{e,v} = 1$ if node v is the head of edge e, $b_{e,v} = -1$ if node v is the tail of edge e, and $b_{e,v} = 0$ otherwise. Let e_i denote the i-th standard basis vector. For an edge $e \in \mathcal{E}$ linking two nodes i and j, its corresponding row vector can be written as $b_{ij} \triangleq b_e = e_i - e_j$. Then, the Laplacian matrix can be written as $\mathcal{L} = B^\top B = \sum_{e\in\mathcal{E}} b_e b_e^\top$.

2.2 Random Walks on a Graph

For a connected undirected network $\mathcal{G}$, we can define an unbiased discrete time random walk taking place on it. At any time step, the walker starting from its current location moves to any of its neighboring vertex with the same probability. Such a stochastic process is described by a Markov chain [27], characterized by the transition matrix $T = D^{-1}A$, with the ij-th entry $t_{ij} = a_{ij}/d_i$ representing the probability of jumping to j from i in one time step. If $\mathcal{G}$ is a finite non-bipartite graph, the random walk is an ergodic Markov chain [27], which has a unique stationary distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_N)^\top$ satisfying the following three conditions: $\pi_i = d_i/(2E)$, $\sum_{i=1}^{N} \pi_i = 1$, and $\pi^\top T = \pi^\top$.

A fundamental quantity for random walks is hitting time, also called first-passage time [13]. The hitting time from vertex i to vertex j, denoted by F_{ij}, is the expected time for a walker starting from i to visit j for the first time. Many other interesting quantities related to random walks can be expressed in terms of, or be encoded in, hitting times. For example, the Kemeny constant K of random walks on $\mathcal{G}$ is a weighted average of hitting times [27]: $K = \sum_{j=1}^{N} \pi_j F_{ij}$ defined as the expected time for a walker starting from a vertex i to another vertex j selected randomly from the vertex set $\mathcal{V}$, according to the stationary distribution π.

Many important structural and dynamical properties of a network are related to or determined by the spectra of its transition

matrix. However, except for regular graphs, the transition matrix T of a network is not symmetric. So, we introduce another matrix P similar to T, which is defined by

$$P = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = D^{\frac{1}{2}} T D^{-\frac{1}{2}}, \tag{1}$$

where $D^{-\frac{1}{2}}$ is a diagonal matrix with its ith diagonal entry being $1/\sqrt{d_i}$. By definition, P is real and symmetric and thus has an identical set of eigenvalues as T. Hereafter, 0 denotes the number zero, the zero matrix or zero vector of appropriate dimensions, and I stands for the identity matrix of appropriate dimensions. Then, the normalized Laplacian matrix [10, 49, 50] of a network is $L = I - P$.

Let $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_N$ be the N eigenvalues of transition matrix T, which obey relation $\lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_N = 0$. By construction, all eigenvalues λ_i $(i = 1, 2, \ldots, N)$ are real, which can be rearranged in a decreasing order as $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_N \geq -1$. Let $\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_N$ be the N eigenvalues of normalized Laplacian matrix L, which can be ranked in an increasing order as $0 = \sigma_1 < \sigma_2 \leq \sigma_3 \leq \cdots \leq \sigma_N \leq 2$. It has been proved [27] that

$$K = \sum_{j=1}^{N} \pi_j F_{ij} = 1 + \sum_{k=2}^{N} \frac{1}{1-\lambda_k} = 1 + \sum_{k=2}^{N} \frac{1}{\sigma_k}. \tag{2}$$

Thus, the Kemeny constant is a global spectral characteristic of a network. From (2), the Kemeny constant K is independent of the starting vertex i, it is then also referred to as eigentime identity.

2.3 Related Work

Due to the broad range of applications, the Kemeny constant has received considerable attention [22]. Particularly, in order to uncover the effect of disparate topological properties on the behavior of Kemeny constant, many groups have made concerted efforts to this key quantity for networks with distinct topological properties. It was shown that in different networks with size N but distinct structure, the Kemeny constant K often behaves differently with N. For example, in Koch networks [41, 48] and small-world trees [26, 54] such as Cayley trees, $K \sim N \ln N$; in extended Sierpiński graphs [42] and fractal trees [26, 55], K varies superlinearly as $K \sim N^{\theta}$ with $1 < \theta < 2$; and in the path graph, $K \sim N^2$. In addition, it was proved that among all graphs with N nodes, the minimum possible value of the Kemeny constant is $1 + (N-1)^2/N$, which can be uniquely attained in the complete graphs [39] .

Previous work implies that the linear growth with network size is the possible minimal scaling for the Kemeny constant. A network is called optimal if this linear scaling for the Kemeny constant can be reached. In this sense, the complete graph is an absolutely optimal graph. However, complete graphs are dense and cannot describe real networked systems, which are sparse and exhibit simultaneously the striking scale-free [5] and small-world [45] properties. The scale-free property means that the degree distribution $P(k)$ of nodes follows a power-law form $P(k) \sim k^{-\gamma}$ with power exponent lying between 2 and 3, while the small-world property implies that the average distance over all pairs of nodes scales at most logarithmically with the number of nodes. Thus, it is of theoretical and practical interest to design or find optimal sparse graphs with scale-free small-world features, where the minimal scaling for the Kemeny constant can be achieved. Moreover, since computing Kemeny constant via evaluating the eigenvalues of graph Laplacian is computationally expensive for large-scale networks, it is of great interest to develop an efficient and fast algorithm for estimating the Kemeny constant of an arbitrary graph.

Table 1: Statistics of some datasets and their Kemeny constant K. For a network with N nodes and E edges, we denote the number of nodes and edges in its largest connected component by N' and E', respectively.

Network	N	E	N'	E'	γ	K/N'
Hamsterster friendships	1,858	12,534	1,788	12,476	2.461	1.193
Protein	1,870	2,203	1,458	1,948	2.879	2.601
Hamster full	2,426	16,631	2,000	16,098	2.421	1.380
Human protein (Vidal)	3,133	6,149	2,783	6,007	2.132	1.517
Route views	6,474	12,572	6,474	12,572	2.462	1.246
arXiv astro-ph	18,771	198,050	17,903	196,972	2.861	1.281
CAIDA	26,475	53,381	26,475	53,381	2.509	1.206
Internet topology	34,761	107,720	34,761	107,720	2.233	1.146
Brightkite	58,228	214,078	56,739	212,945	2.481	1.426

In the following sections, we will study the Kemeny constant for scale-free small-world sparse networks and design a fast randomized algorithm for approximating the Kemeny constant of a general graph. We first study the Kemeny constant for some real scale-free networks and show that the ratio of the Kemeny constant to the number of vertices is constant. Then we determine the Kemeny constant for two deterministic scale-free small-world sparse networks [16, 57] and the Barabási-Albert network [5], and show that their Kemeny constants behave linearly with the network size. Thus, scale-free networks are optimal with minimal scaling of Kemeny constant. Finally, we present a fast algorithm to approximate the Kemeny constant for a graph, whose complexity scales nearly linearly with the number of edges in the graph. Also, we test our algorithm on many real networks, as well as the two considered deterministic networks [16, 57].

3 THE KEMENY CONSTANT IN REALISTIC SCALE-FREE NETWORKS

In this section, we study the Kemeny constant of some real-life networks having a power-law degree distribution $P(k) \sim k^{-\gamma}$ with γ in the interval of $(2, 3)$. We use a large collection of networks chosen from different domains.

In Table 1, we report the Kemeny constant of some real-world scale-free networks. All data sets are taken from the Koblenz Network Collection [29]. The considered real-life networks are representative, including social networks, information networks, technological networks, and metabolic networks. For those networks that are disconnected originally, we compute the Kemeny constant for their largest connected components (LCC). Related information for the studied real networks and their LCC is shown in Table 1, where the networks are listed in an increasing order of the number of nodes. The smallest network includes about 2×10^3 vertices, while the largest network contains approximately 6×10^4 vertices.

Table 1 shows that for the considered realistic scale-free networks with power exponent $2 < \gamma \leqslant 3$, their Kemeny constant is very small. Moreover, the ratio of the Kemeny constant to the number of nodes is constant, which is a little larger than 1 for most

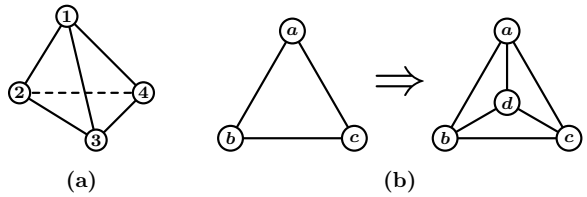

Figure 1: (a) Initial construction of the Apollonian network. (b) Iterative construction method of the Apollonian network. One can obtain the next iteration the Apollonian network by performing the operation on the right-hand side of the arrow for each active triangle.

of the networks. Therefore, for the studied scale-free realistic networks, their Kemeny constant grows linearly with the number of nodes, a phenomenon similar to that in the complete graphs.

In fact, the linear growth of the Kemeny constant found in real-world scale-free networks is universal. In the following three sections, we will study the Kemeny constant in three model scale-free networks, the Kemeny constant of which also scales linearly with the number of nodes.

4 THE KEMENY CONSTANT IN THE APOLLONIAN NETWORK

In this section, we study the Kemeny constant for random walks on the Apollonian network with scale-free small-world properties [16].

4.1 Network Construction and Properties

The Apollonian network was derived from the Apollonian packing and was proposed independently in [3] and in [16], with different initial constructions but similar structural and dynamical properties. Here we focus on the version in [16], which can be defined in an iterative manner [56]. Let $\mathcal{A}_g = (\mathcal{V}_g, \mathcal{E}_g)$ denote Apollonian network after g generation evolution. Initially ($g = 0$), $\mathcal{A}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ is a tetrahedron consisting of four faces or triangles, see Fig. 1 (a). Let $1, 2, 3, 4$ denote the four vertices in $\mathcal{A}_0$. Then, $\mathcal{V}_0 = \{1, 2, 3, 4\}$ and $\mathcal{E}_0 = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$. For three vertices a, b, and c in $\mathcal{A}_g$, if they form a triangle of $\mathcal{A}_g$ that does not contain any smaller triangles in it, we call it an active triangle of $\mathcal{A}_g$ and use a tuple (a, b, c) to denote this active triangle. Let $\mathcal{S}_g$ be the set of active triangles of $\mathcal{A}_g$. As we will show below, $\mathcal{S}_g$ includes $4 \cdot 3^g$ active triangles. By definition, $\mathcal{S}_0 = \{(1, 2, 3), (1, 2, 4), (1, 3, 4), (2, 3, 4)\}$. Let Δ_i be the ith ($i = 1, 2, \ldots, 4 \times 3^g$) active triangle in $\mathcal{S}_g$. Given that we have $\mathcal{A}_g$ ($g \geq 0$), $\mathcal{A}_{g+1}$ can be constructed from $\mathcal{A}_g$ as follows, see Fig. 1 (b). For every active triangle $\Delta_i = (a, b, c)$ containing three vertices a, b, c in $\mathcal{A}_g$, we add a new vertex d inside it and connect vertex d to the three vertices a, b, and c of the active triangle. Figure 2 illustrates the network $\mathcal{A}_2$.

Let $N_g = |\mathcal{V}_g|$ and $E_g = |\mathcal{E}_g|$ denote, respectively, the numbers of vertices and edges in $\mathcal{A}_g$. Let $\mathcal{W}_{g+1} = \mathcal{V}_{g+1} \backslash \mathcal{V}_g$ denote the set of new vertices introduced at iteration $g+1$, and let $W_g = |\mathcal{W}_g|$ denote the number of these newly introduced vertices. Let $S_g = |\mathcal{S}_g|$ be the number of active triangles of $\mathcal{A}_g$. Then, we have the following

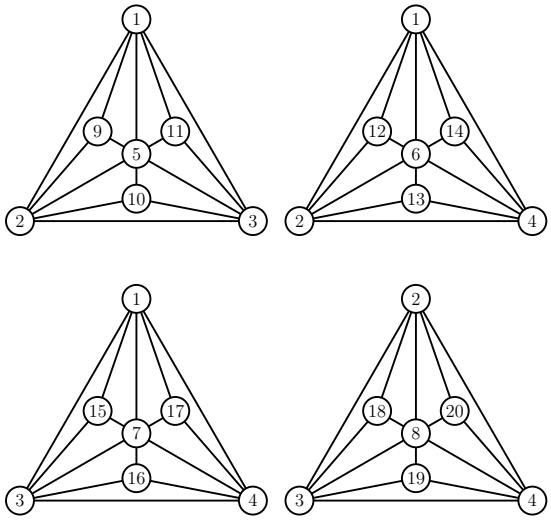

Figure 2: The Apollonian network $\mathcal{A}_2$ and its vertex labelings.

relations:

$$\mathcal{S}_{g+1} = \bigcup_{\Delta_i=(a,b,c)\in\mathcal{S}_g} \{(a, b, N_g + i), (a, c, N_g + i), (b, c, N_g + i)\},$$

$$\mathcal{V}_{g+1} = \mathcal{V}_g \bigcup \{N_g + 1, N_g + 2, \ldots, N_g + S_g\},$$

$$\mathcal{E}_{g+1} = \mathcal{E}_g \bigcup_{\Delta_i=(a,b,c)\in\mathcal{S}_g} \{(a, d = N_g + i), (b, d), (c, d)\}.$$

By construction, each active triangle of $\mathcal{A}_g$ generates three active triangles of $\mathcal{A}_{g+1}$, which means $S_{g+1} = 3 \cdot S_g = 12 \cdot 3^g$. Since each active triangle of $\mathcal{A}_g$ gives rise to one new vertex and three new edges at the $(g + 1)$th iteration, we have $W_{g+1} = S_g = 4 \cdot 3^g$, $N_{g+1} = N_g + W_{g+1}$, and $E_{g+1} = E_g + 3S_g$, which lead to

$$N_g = 2 \cdot 3^g + 2 \tag{3}$$

and

$$E_g = 6 \cdot 3^g . \tag{4}$$

Therefore, the average degree of all vertices in $\mathcal{A}_g$ is $2E_g/N_g$, which tends to 6 for large networks, indicating that the Apollonian network is sparse. Let $d_i(g)$ be the degree of vertex i in $\mathcal{A}_g$, which was generated at iteration g_i ($g_i \geq 0$). Then, $d_i(g + 1) = 2\, d_i(g) = 3 \times 2^{g-g_i}$.

The Apollonian network displays the typical features of various real-life networks [16]. It is scale-free with its degree distribution $P(k)$ having a power-law form $P(k) \sim k^{-(1+\ln 3/\ln 2)}$. Moreover, it is small-world with its diameter increasing as a logarithmic function of the network size [56].

4.2 Recursive Relations for Matrices

After introducing the construction and properties of the Apollonian network, we present a recursive formulation to calculate the eigenvalues of the network which subsequently can be used to calculate the Kemeny constant.

Let A_g and D_g denote, respectively, the adjacency matrix and diagonal degree matrix of $\mathcal{A}_g$. The element $A_g(i,j)$ at row i and column j of A_g is defined as: $A_g(i,j) = 1$ if vertices i and j are adjacent in $\mathcal{A}_g$, $A_g(i,j) = 0$ otherwise. Then, the transition matrix of $\mathcal{A}_g$, denoted by T_g, is defined by $T_g = (D_g)^{-1}A_g$, the ijth entry of which is $T_g(i,j) = A_g(i,j)/d_i(g)$. The normalized Laplacian matrix of $\mathcal{A}_g$, denoted by L_g, is $L_g = I - (D_g)^{\frac{1}{2}}T_g(D_g)^{-\frac{1}{2}}$. We next determine the relations between the three matrices A_g, D_g, and T_g.

For the Apollonian network $\mathcal{A}_{g+1}$, let α denote the set of old vertices already existing at generation g, and β the set of new vertices belonging to $\mathcal{W}_{g+1}$. Then, A_{g+1} can be written in the following block form

$$A_{g+1} = \begin{pmatrix} A_{g+1}^{\alpha,\alpha} & A_{g+1}^{\alpha,\beta} \\ A_{g+1}^{\beta,\alpha} & A_{g+1}^{\beta,\beta} \end{pmatrix} = \begin{pmatrix} A_g & A_{g+1}^{\alpha,\beta} \\ A_{g+1}^{\beta,\alpha} & 0 \end{pmatrix},$$

where $A_{g+1}^{\alpha,\alpha} = A_g$, $A_{g+1}^{\beta,\beta}$ is the zero matrix with order $W_{g+1} \times W_{g+1}$, and $A_{g+1}^{\alpha,\beta} = (A_{g+1}^{\beta,\alpha})^\top$. The diagonal matrix D_g obeys relation

$$D_{g+1} = \begin{pmatrix} D_{g+1}^{\alpha,\alpha} & 0 \\ 0 & D_{g+1}^{\beta,\beta} \end{pmatrix} = \begin{pmatrix} 2D_g & 0 \\ 0 & 3I \end{pmatrix},$$

which is based on the fact that during the network evolution from iteration g to iteration $g+1$, the degree of vertices in α doubles, and the degree of all vertices in β is 3. And the transition matrix T_g evolves as

$$T_{g+1} = D_{g+1}^{-1}A_{g+1} = \begin{pmatrix} \frac{1}{2}T_g & \frac{1}{2}D_g^{-1}A_{g+1}^{\alpha,\beta} \\ \frac{1}{3}A_{g+1}^{\beta,\alpha} & 0 \end{pmatrix}.$$

In this way, we have obtained recursive relations for related matrices.

4.3 Eigenvalues of Related Matrices

In order to determine all eigenvalues of the transition matrix T_g, we need some lemmas.

Lemma 4.1. *For the Apollonian network $\mathcal{A}_{g+1}$ after $g+1$ ($g \geq 0$) iterations,*

$$A_{g+1}^{\alpha,\beta}A_{g+1}^{\beta,\alpha} = D_g + 2A_g. \tag{5}$$

Lemma 4.2. *Suppose M is an $N \times N$ matrix with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_N$, and $f_1(x)$ and $f_2(x)$ are two polynomials in x. Then*

$$\det(f_1(x)I - f_2(x)M) = \prod_{i=1}^{N}(f_1(x) - f_2(x)\lambda_i). \tag{6}$$

Let $P_g(x) = \det(xI - T_g)$ denote the characteristic polynomial of matrix T_g. The following lemma provides an expression for the characteristic polynomial $P_g(x)$.

Lemma 4.3. *For $g \geq 0$,*

$$P_{g+1}(x) = x^{W_{g+1}-N_g}\det\left(\left(x^2 - \frac{1}{6}\right)I - \left(\frac{1}{2}x + \frac{1}{3}\right)T_g\right).$$

Let $\lambda_1^{(g)}, \lambda_2^{(g)}, \cdots, \lambda_{N_g}^{(g)}$ be the N_g eigenvalues of transition matrix T_g, and let $\Lambda(T_g)$ be the set of these N_g eigenvalues, that is, $\Lambda(T_g) = \left\{\lambda_1^{(g)}, \lambda_2^{(g)}, \cdots, \lambda_{N_g}^{(g)}\right\}$. By definition, $\lambda_i^{(g)}$ ($i = 1, 2, \ldots, N_g$) are the roots of $P_g(x) = \det(xI - T_g) = 0$. Define two functions $g_1(x)$ and $g_2(x)$:

$$g_1(x) = \frac{x}{4} + \frac{1}{4}\sqrt{x^2 + \frac{16}{3}x + \frac{8}{3}} \tag{7}$$

and

$$g_2(x) = \frac{x}{4} - \frac{1}{4}\sqrt{x^2 + \frac{16}{3}x + \frac{8}{3}}. \tag{8}$$

The following theorem shows that all the eigenvalues of matrix T_{g+1} can be obtained from those of T_g.

Theorem 4.4. *The eigenvalue set $\Lambda(T_{g+1})$ of matrix T_{g+1} consists of two subsets $\Lambda_1(T_{g+1})$ and $\Lambda_2(T_{g+1})$, satisfying $\Lambda(T_{g+1}) = \Lambda_1(T_{g+1}) \cup \Lambda_2(T_{g+1})$, where $\Lambda_1(T_{g+1})$ includes only eigenvalue 0 with multiplicity $N_{g+1} - 2N_g = 2 \cdot 3^g - 2$, $\Lambda_2(T_{g+1})$ contains the remaining $2N_g = 2(2 \cdot 3^g + 2)$ eigenvalues $\lambda_{i,1}^{(g+1)}$ and $\lambda_{i,2}^{(g+1)}$ generated by $\lambda_i^{(g)}$ ($i = 1, 2, \ldots, N_g$) in the following way*

$$\lambda_{i,1}^{(g+1)} = g_1(\lambda_i^{(g)}) = \frac{\lambda_i^{(g)}}{4} + \frac{1}{4}\sqrt{(\lambda_i^{(g)})^2 + \frac{16}{3}\lambda_i^{(g)} + \frac{8}{3}}, \tag{9}$$

$$\lambda_{i,2}^{(g+1)} = g_2(\lambda_i^{(g)}) = \frac{\lambda_i^{(g)}}{4} - \frac{1}{4}\sqrt{(\lambda_i^{(g)})^2 + \frac{16}{3}\lambda_i^{(g)} + \frac{8}{3}}. \tag{10}$$

Proof. According to Lemmas 4.2 and 4.3,

$$\begin{aligned} P_{g+1}(x) &= x^{W_{g+1}-N_g}\det\left(\left(x^2 - \frac{1}{6}\right)I - \left(\frac{1}{2}x + \frac{1}{3}\right)T_g\right) \\ &= x^{W_{g+1}-N_g}\prod_{i=1}^{N_g}\left(\left(x^2 - \frac{1}{6}\right) - \left(\frac{1}{2}x + \frac{1}{3}\right)\lambda_i^{(g)}\right). \end{aligned} \tag{11}$$

By definition, the N_{g+1} eigenvalues of T_{g+1} are the N_{g+1} roots of $P_{g+1}(x) = 0$. (11) indicates that 0 is an eigenvalue of T_{g+1} with multiplicity $W_{g+1} - N_g = N_{g+1} - 2N_g = 2 \cdot 3^g - 2$. These eigenvalues 0 form the subset $\Lambda_1(T_{g+1})$.

Except the $2 \cdot 3^g - 2$ eigenvalues 0, the other $2N_g$ eigenvalues of matrix T_{g+1} can be determined by equation

$$\prod_{i=1}^{N_g}\left(\left(x^2 - \frac{1}{6}\right) - \left(\frac{1}{2}x + \frac{1}{3}\right)\lambda_i^{(g)}\right) = 0, \tag{12}$$

which is equivalent to

$$\xi_i(x) \triangleq x^2 - \frac{\lambda_i^{(g)}}{2}x - \left(\frac{\lambda_i^{(g)}}{3} + \frac{1}{6}\right) = 0, \tag{13}$$

$i = 1, 2, \ldots, N_g$. From (13), we obtain that for an arbitrary element $\lambda_i^{(g)}$ in set $\Lambda(T_g)$, both solutions of $\xi_i(x) = 0$, denoted by $\lambda_{i,1}^{(g+1)}$ and $\lambda_{i,2}^{(g+1)}$, are in $\Lambda_2(T_{g+1})$, which are given by (9) and (10), respectively. Thus, each eigenvalue $\lambda_i^{(g)}$ generates two eigenvalues of matrix $\Lambda(T_{g+1})$, and all the N_g eigenvalues of matrix $\Lambda(T_g)$ give rise to $2N_g = 2(2 \cdot 3^g + 2)$ eigenvalues, which constitute subset $\Lambda_2(T_{g+1})$. □

Theorem 4.4 provides a recursive expression for eigenvalues of the transition matrix for the Apollonian network $\mathcal{A}_g$, as well as their multiplicity. Actually, using a similar approach, we can also determine the eigenvalues corresponding to the normalized Laplacian matrix L_g.

Let $\sigma_1^{(g)}, \sigma_2^{(g)}, \cdots, \sigma_{N_g}^{(g)}$ be the N_g eigenvalues of the normalized Laplacian matrix L_g, and let $\Omega(L_g)$ be the set of these N_g eigenvalues, that is, $\Omega(L_g) = \left\{\sigma_1^{(g)}, \sigma_2^{(g)}, \cdots, \sigma_{N_g}^{(g)}\right\}$. Then, the relation $\sigma_i^{(g)} = 1 - \lambda_i^{(g)}$ holds for all $i = 1, 2, \ldots, N_g$.

Corollary 4.5. *The eigenvalue set $\Omega(L_{g+1})$ of matrix L_{g+1} consists of two subsets $\Omega_1(L_{g+1})$ and $\Omega_2(L_{g+1})$, satisfying $\Omega(L_{g+1}) = \Omega_1(L_{g+1}) \cup \Omega_2(L_{g+1})$, where $\Omega_1(L_{g+1})$ includes only eigenvalue 1 with multiplicity $N_{g+1} - 2N_g = 2 \cdot 3^g - 2$, $\Omega_2(L_{g+1})$ contains the remaining $2N_g = 2(2 \cdot 3^g + 2)$ eigenvalues $\sigma_{i,1}^{(g+1)}$ and $\sigma_{i,2}^{(g+1)}$ generated by $\sigma_i^{(g)}$ $(i = 1, 2, \ldots, N_g)$ in the following way*

$$\sigma_{i,1}^{(g+1)} = \frac{3}{4} + \frac{\sigma_i^{(g)}}{4} - \frac{1}{4}\sqrt{\left(\sigma_i^{(g)}\right)^2 - \frac{22}{3}\sigma_i^{(g)} + 9}, \tag{14}$$

$$\sigma_{i,2}^{(g+1)} = \frac{3}{4} + \frac{\sigma_i^{(g)}}{4} + \frac{1}{4}\sqrt{\left(\sigma_i^{(g)}\right)^2 - \frac{22}{3}\sigma_i^{(g)} + 9}\,. \tag{15}$$

The proof is similar to that of Theorem 4.4, we here omit the proof detail.

4.4 The Kemeny Constant

We are now in position to use the obtained eigenvalues to determine the Kemeny constant for the Apollonian network $\mathcal{A}_g$, denoted by K_g.

Theorem 4.6. *For $g \geq 0$, the closed-form expression for the Kemeny constant K_g of the Apollonian network $\mathcal{A}_g$ is*

$$K_g = 1 + \frac{1}{12}\left(32 \times 3^g - 16\left(\frac{9}{5}\right)^g + 11\right). \tag{16}$$

For $g \to \infty$,

$$K_g \sim \frac{4}{3}N_g. \tag{17}$$

Proof. According to (2),

$$K_g = 1 + \sum_{i=2}^{N_g} \frac{1}{\sigma_i^{(g)}}\,. \tag{18}$$

Let R_g denote the sum of reciprocals of nonzero eigenvalues for matrix L_g. That is, $R_g = \sum_{i=2}^{N_g} \frac{1}{\sigma_i^{(g)}}$, which can be evaluated as

$$R_g = \sum_{\sigma_i^{(g)} \in \Omega_1(L_g)} \frac{1}{\sigma_i^{(g)}} + \sum_{\sigma_i^{(g)} \in \Omega_2(L_g)\setminus\{0\}} \frac{1}{\sigma_i^{(g)}}. \tag{19}$$

By Corollary 4.5,

$$R_g = N_g - 2N_{g-1} + \frac{1}{\sigma_{1,2}^{(g)}} + \sum_{i=2}^{N_{g-1}}\left(\frac{1}{\sigma_{i,1}^{(g)}} + \frac{1}{\sigma_{i,2}^{(g)}}\right). \tag{20}$$

Considering $\sigma_1^{(g-1)} = 0$, (14) and (15), we have $\sigma_{1,1}^{(g)} = 0$, $\sigma_{1,2}^{(g)} = 3/2$, and

$$\frac{1}{\sigma_{i,1}^{(g)}} + \frac{1}{\sigma_{i,2}^{(g)}} = \frac{3}{5} + \frac{9}{5\sigma_i^{(g-1)}} \tag{21}$$

for all $i = 2, 3, \ldots, N_{g-1}$. Then,

$$\begin{aligned} R_g &= N_g - 2N_{g-1} + \frac{2}{3} + \frac{3}{5}(N_{g-1} - 1) + \frac{9}{5}\sum_{i=2}^{N_{g-1}} \frac{1}{\sigma_i^{(g-1)}} \\ &= \frac{9}{5}R_{g-1} + \frac{16}{5}3^{g-1} - \frac{11}{15}. \end{aligned} \tag{22}$$

Using the initial condition $R_0 = 9/4$, (22) is solved to obtain

$$R_g = \frac{1}{12}\left(32 \times 3^g - 16\left(\frac{9}{5}\right)^g + 11\right), \tag{23}$$

inserting which into (18) giving (16).

We now express K_g as a function of the network size N_g. From $N_g = 2 \times 3^g + 2$, we have $3^g = (N_g - 2)/2$ and $g = (\ln(N_g - 2) - \ln 2)/\ln 3$. Hence, the Kemeny constant K_g in (16) can be expressed in terms of network size N_g as

$$K_g = 1 + \frac{1}{12}\left(16N_g - 16\left(\frac{N_g}{2} - 1\right)^{2-\ln 5/\ln 3} - 21\right). \tag{24}$$

When $g \to \infty$, one obtains (17). □

Thus, for large N_g, the Kemeny constant for the sparse Apollonian network behaves linearly with N_g, which is similar to that for dense complete graphs. Then, the Apollonian network is an optimal scale-free small-world network in the sense that it has the least scaling of Kemeny constant.

5 THE KEMENY CONSTANT IN EXTENDED PSEUDOFRACTAL NETWORKS

In this section, we study the Kemeny Constant in extended pseudofractal networks with the scale-free small-world features [57]. We will show that their Kemeny Constant also increases linearly with the network size.

The extended pseudofractal networks are an extension of pseudofractal web [15], which are also built in an iterative way. Let $\mathcal{F}_g = (\mathcal{V}_g, \mathcal{E}_g)$ denote the network family after g $(g \geq 0)$ iterations. For $g = 0$, $\mathcal{F}_0$ is a triangle. For $g \geq 1$, $\mathcal{F}_g$ is obtained from $\mathcal{F}_{g-1}$ by performing the operation in Fig. 3 (a): every edge in $\mathcal{E}_{g-1}$ generates m (a positive integer) additional vertices, which are attached to both end vertices of this edge. Figure 3 (b) illustrates the network $\mathcal{F}_2$ for $m = 2$.

In what follows, we use the same notations as those for the Apollonian network studied in the previous section. By construction, we have the following relations: $E_{g+1} = (2m+1)E_g$, $W_{g+1} = mE_g$, and $N_{g+1} = N_g + W_{g+1}$. Thus, for all $g \geq 0$, $E_g = 3(2m+1)^g$, $W_{g+1} = 3m(2m+1)^g$, and $N_g = 3((2m+1)^g + 1)/2$. In addition, for a vertex i generated at iteration g_i, its degree evolves as $d_i(g+1) = (m+1)d_i(g) = 2(m+1)^{g-g_i}$.

The extended pseudofractal networks are sparse with an average degree 4. They also display the remarkable scale-free and small-world features observed in many real-life networks [57]. Their degree distribution $P(k)$ follows a power-law behavior $P(k) \sim k^{-\gamma_m}$ with $\gamma_m = 1 + \ln(2m+1)/\ln(m+1)$. And their diameter grows logarithmically with the network size [57].

In a similar way to that of Appolonian network, the explicit formula and the leading behavior for the Kemeny constant K_g of the extended pseudofractal networks $\mathcal{F}_g$ are summarized in the following theorem.

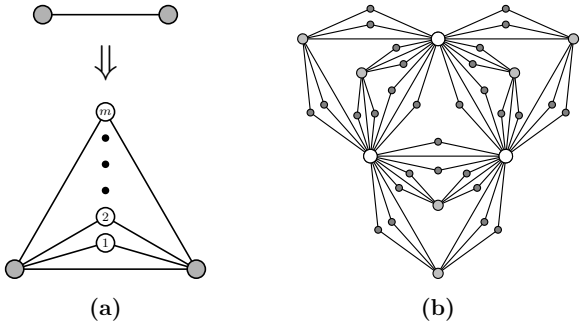

Figure 3: (a) Construction approach for the extended pseudofractal networks. The next iteration is obtained by performing the operation on the bottom of the arrow for each existing edge. (b) Illustration for an extended pseudofractal network $\mathcal{F}_2$ corresponding to a particular case $m = 2$.

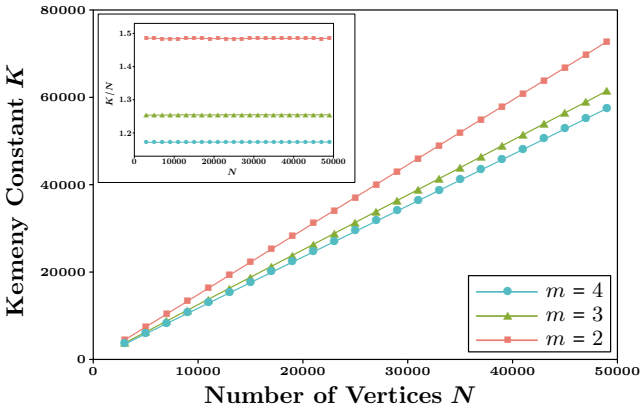

Figure 4: Kemeny constant on the Barabási-Albert network.

Theorem 5.1. *For $g \geq 0$, the exact expression for the Kemeny constant K_g of the extended pseudofractal networks $\mathcal{F}_g$ is*

$$K_g = 1 + \frac{1}{30m(2m+1)}\left(\left(8 + 25m + 18m^2\right) + (135m + 90m^2)(1+2m)^g - (28m^2 + 120m + 8)\left(\frac{2+4m}{2+m}\right)^g\right). \quad (25)$$

When $g \to \infty$,

$$K_g \sim \frac{2m+3}{2m+1} N_g. \quad (26)$$

Theorem 5.1 shows that the Kemeny constant of the extended pseudofractal networks also scales linearly with the network size.

6 THE KEMENY CONSTANT IN THE BARABÁSI-ALBERT NETWORK

In the preceding sections, we have studied the Kemeny constant in some realistic scale-free networks and two deterministic scale-free small-world networks, and presented that the leading scaling of their Kemeny constant grows linearly with the network size. To further investigate the universality of this linear scaling about Kemeny constant for random walks in scale-free networks, we also study the Kemeny constant in the popular Barabási-Albert network [5], and observe a linear behavior for the Kemeny constant.

As a classic scale-free network model, the Barabási-Albert network [5] is generated by applying the following algorithm. Initially, we have a connected graph with a small number $m_0 \geq m$ vertices, with $m \geq 1$. At every time step, we generate a new vertex with m links, and connect it to m different old nodes, with the probability that the new vertex is linked to an old vertex i being proportional to the degree of i. After performing the operations of growth and preferential attachment a sufficient number of times, we obtain a Barabási-Albert scale-free network with a power-law degree distribution $P(k) \sim k^{-3}$ and average degree $2m$.

We study the Kemeny constant for random walks on various Barabási-Albert networks with different network size and average degree. In Fig. 4, we present the numerical results for Kemeny constant on Barabási-Albert networks, which grows linearly with the number of vertices. Thus, the linear scaling of Kemeny constant appears to be universal for the Barabási-Albert networks.

7 STRUCTURAL REASONS FOR THE OBSERVED MINIMAL SCALING OF KEMENY CONSTANT

In the four preceding sections, we have investigated the Kemeny constant in some real-world scale-free networks and three model scale-free graphs. We showed that in all studied networks, their Kemeny constant behaves linearly with N, the number of vertices. The behavior is identical to that of complete graphs, the Kemeny constant of which is the smallest among all graphs with the same number of vertices. Therefore, all considered networks are almost optimal in the sense that they have the minimal scaling for the Kemeny constant. Because the Kemeny constant of a graph is fully determined by the non-zero eigenvalues of its normalized Laplacian matrix, which are in turn affected by the graph topology, we argue that the observed linear scaling for Kemeny constant of the studied networks lies in the scale-free and small-world structure, as well as the presence of cycles of various length. The following heuristic arguments are helpful to deepen our understanding.

In a scale-free graph, there exist large-degree vertices directly connected to many other vertices. Moreover, for those real and model scale-free graphs with different cycles of various lengths, their average geodesic distance is very low, which scales at most logarithmically with the vertex number N [37]. The aggregation of these structural properties considerably influences various dynamics on graphs. For example, for random walks on scale-free small-world loopy graphs, the hitting time to a large-degree hub vertex scales sublinearly with N [36]. In contrast, the hitting time to a small vertex is higher, behaving linearly with N. By definition in (2), the Kemeny constant $K = \sum_{j=1}^{N} \pi_j F_{ij}$ is a weighted average of hitting times. Although putting more weight on the hitting time to a hub vertex, K is a linear function of N, which is due to the fact that hub vertices are much less, in comparison with the low-degree vertices.

As we claimed above, the linear scaling of the Kemeny constant is the result of the synergy of scale-free, small-world, and loopy properties. Since scale-free behavior and presence of cycles with different lengths can often result in the small-world phenomenon, below, we will illustrate that either cycles or scale-free behavior alone cannot guarantee the linear scaling of the Kemeny constant.

First, according to [58], for a graph $\mathcal{G}$ with the minimal degree $d_{\min}$, the Kemeny constant K satisfies $K \geq \Omega(\mathcal{G})d_{\min}/N$, where $\Omega(\mathcal{G})$ is the Kirchhoff index [19] of $\mathcal{G}$. When $\mathcal{G}$ is the Farey graphs that have similar structural properties as those of the Watt-Strogatz small-world model [45], $K \geq N \ln N$, since in this case, $d_{\min} = 2$ and $\Omega(\mathcal{G}) \sim N^2 \ln N$ [51]. The scaling $N \ln N$ of K for the Farey graphs are much larger than the linear scaling for extended pseudofractal networks. We notice that both Farey graphs and extended pseudofractal networks are small-world and possess various cycles at different scales. The main reason for the difference between their Kemeny constant is the power-law property of the extended pseudofractal networks, which does not exist in Farey graphs.

We continue to show that only scale-free small-world properties cannot necessarily lead to $K \sim N$. For this purpose, we consider Kemeny constant of the Koch networks [41, 48], for which $K \geq N \ln N$ in spite of the fact that the Koch networks are simultaneously scale-free and small-world. The reason that the Kemeny constant of the Koch networks is greater than linear scaling is as follows: there exist only triangles in Koch networks, lacking cycles of other lengths.

8 FAST ALGORITHM FOR COMPUTING KEMENY CONSTANT

As we know, the Kemeny constant K can be expressed in terms of the normalized Laplacian as $K = 1 + \sum_{k=2}^{N} \frac{1}{\sigma_k} = 1 + \mathrm{Tr}\left(L^{\dagger}\right)$, where $L^{\dagger}$ is the Moore-Penrose inverse of the normalized Laplacian matrix and $\sigma_2, \sigma_3, \cdots, \sigma_N$ are the nonzero eigenvalues of L. A straightforward way to calculate Kemeny constant K of a graph involves computing either the eigenvalues or the pseudoinverse $L^{\dagger}$ of the normalized Laplacian matrix L, both of which have a complexity of $O(N^3)$ and are intractable to huge networks. In this section, we introduce a randomized algorithm to compute an approximation of K for a general graph in nearly linear time with respect to the number of edges.

8.1 Approximation Algorithm and its Theoretical Performance

Our method is to approximate the trace of $L^{\dagger}$ based on Hutchinson's Monte-Carlo method [23]. For this purpose, we generate M independent vectors $x_1, x_2, \cdots, x_M \in \mathbb{R}^n$, with the entry of each vector being 1 or -1 with identical probability. Then, for an n-dimensional positive semi-definite matrix A, $\frac{1}{M}\sum_{i=1}^{M} x_i^{\top} A x_i$ can be used to estimate the trace $\mathrm{Tr}(A)$ of A. Since $\mathbb{E}\left[x_i^{\top} A x_i\right] = \mathrm{Tr}(A)$, by the law of large numbers, $\frac{1}{M}\sum_{i=1}^{M} x_i^{\top} A x_i$ should be close to $\mathrm{Tr}(A)$ when M is large. The following lemma [4] provides the performance of $\frac{1}{M}\sum_{i=1}^{M} x_i^{\top} A x_i$ as an estimation of $\mathrm{Tr}(A)$.

Lemma 8.1. *Let A be a positive semidefinite matrix with rank $\mathrm{rank}(A)$. Let $x_1, \ldots, x_M$ be M independent vectors, for each of which their entries are 1 or -1 with the same probability. Let ϵ, δ be scalars such that $0 < \epsilon \leq 1/2$ and $0 < \delta < 1$. Then, for any $M \geq 24\epsilon^{-2}\ln(2\mathrm{rank}(A)/\delta)$, the following statement holds with probability at least $1 - \delta$:*

$$(1-\epsilon)\mathrm{Tr}(A) \leq \frac{1}{M}\sum_{i=1}^{M} x_i^{\top} A x_i \leq (1+\epsilon)\mathrm{Tr}(A).$$

Using Lemma 8.1, the estimation of the Kemeny constant K can be reduced to evaluating the quadratic forms of $L^{\dagger}$. However, if we directly compute the quadratic forms, we must first evaluate $L^{\dagger}$, the time cost for which is high. To avoid the inverse operation of a matrix, we will utilize the nearly linear time solver for Laplacian systems from [30], the performance of which is characterized in the following lemma, where the notation $\widetilde{O}(\cdot)$ hides $\mathrm{poly}(\log N)$ factors.

Lemma 8.2. *The algorithm $z = \textsc{LaplSolve}(\mathcal{L}, y, \epsilon)$ takes a Laplacian matrix $\mathcal{L}$ of a graph $\mathcal{G}$ with N nodes and E edges, a vector $y \in \mathbb{R}^N$ and a scalar $\epsilon > 0$ as input, and returns a vector $z \in \mathbb{R}^N$ such that with probability $1 - 1/\mathrm{poly}(N)$ the following statement holds:*

$$\left\| z - \mathcal{L}^{\dagger} y \right\|_{\mathcal{L}} \leq \epsilon \left\| \mathcal{L}^{\dagger} y \right\|_{\mathcal{L}},$$

where $\|x\|_{\mathcal{L}} = \sqrt{x^{\top}\mathcal{L}x}$. The algorithm runs in expected time $\widetilde{O}(E)$.

However, having only Lemmas 8.1 and 8.2, we still cannot evaluate the quadratic forms of $L^{\dagger}$. Fortunately, this can be solved by using the connection between $L^{\dagger}$ and $\mathcal{L}^{\dagger}$. Let $\mathbf{1}$ be a column vector of approximate dimensions, whose entries are all ones. By definition, we have $\mathcal{L} = D^{\frac{1}{2}} L D^{\frac{1}{2}}$, using this relation we can establish a connection between the Moore-Penrose inverse of L and $\mathcal{L}$ as given in the following lemma [7].

Lemma 8.3. *Given a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes and E edges, with Laplacian matrix $\mathcal{L}$ and normalized Laplacian matrix L, let $\mathcal{L}^{\dagger}$ and $L^{\dagger}$ be the Moore-Penrose inverse of $\mathcal{L}$ and L, respectively. Then,*

$$L^{\dagger} = (I - \frac{1}{2E} D^{\frac{1}{2}} \mathbf{1}\mathbf{1}^{\top} D^{\frac{1}{2}}) D^{\frac{1}{2}} \mathcal{L}^{\dagger} D^{\frac{1}{2}} (I - \frac{1}{2E} D^{\frac{1}{2}} \mathbf{1}\mathbf{1}^{\top} D^{\frac{1}{2}}). \quad (27)$$

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

Input : A graph G with N nodes and E edges; a real number $0 \leq \epsilon \leq 1/2$
Output : the approximation of Kemeny constant $\tilde{K}$

$M = \lceil 48\epsilon^{-2}\ln(2N) \rceil$
for $i = 1$ *to* M **do**
  Generate a vector x_i with each entry being randomly ± 1
  $y_i \leftarrow D^{\frac{1}{2}}(I - \frac{1}{2E} D^{\frac{1}{2}} \mathbf{1}\mathbf{1}^{\top} D^{\frac{1}{2}}) x_i$
  $z_i \leftarrow \textsc{LaplSolve}(\mathcal{L}, y_i, \frac{\epsilon}{3\sqrt{2}} N^{-2.5})$
  Compute $t_i \overset{\text{def}}{=} \|B z_i\|^2$
Compute $\tilde{K} = \frac{1}{M}\sum_{i=1}^{M} t_i$
return $\tilde{K}$

Using Lemmas 8.1, 8.2, and 8.3, we propose an approximation algorithm $\textsc{ApproxKemeny}(\mathcal{G}, \epsilon)$ for computing the Kemeny constant of an arbitrary graph $\mathcal{G}$, as depicted in Algorithm 1, which has a good approximation guarantee. Before giving the approximation factor of our algorithm, we provide the following lemma.

Lemma 8.4. *Let $\mathcal{G}$ be a connected graph with N nodes, and let $\mathcal{L}$ be its Laplacian matrix. Let y be a vector in $\mathbb{R}^N$, and let ϵ be a real number obeying $0 < \epsilon \leq 1/2$. Suppose z is a vector such that*

$$\left\| z - \mathcal{L}^\dagger y \right\|_{\mathcal{L}} \leq \delta \left\| \mathcal{L}^\dagger y \right\|_{\mathcal{L}}, \tag{28}$$

where

$$\delta \leq \frac{\epsilon}{3\sqrt{2}} N^{-2.5}. \tag{29}$$

Then, we have

$$(1-\epsilon)^2 y^\top \mathcal{L}^\dagger y \leq \|Bz\|^2 \leq (1+\epsilon)^2 y^\top \mathcal{L}^\dagger y. \tag{30}$$

The following theorem gives the approximation guarantee of Algorithm 1.

Theorem 8.5. *Given a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes, E edges, and scalar $0 < \epsilon \leq 1/2$, the algorithm ApproxKemeny$(\mathcal{G}, \epsilon)$ returns $\tilde{K}$ as an approximation of the Kemeny constant. With high probability, the following statement holds :*

$$(1-\epsilon)^3 K \leq \tilde{K} \leq (1+\epsilon)^3 K. \tag{31}$$

Proof. Since $M = \lceil 48\epsilon^{-2} \ln(2N) \rceil \geq 48\epsilon^{-2} \ln(2N)$, by Lemma 8.1, we have

$$(1-\epsilon)\mathrm{Tr}\left(L^\dagger\right) \leq \frac{1}{M} \sum_{i=1}^{M} x_i^\top L^\dagger x_i \leq (1+\epsilon)\mathrm{Tr}\left(L^\dagger\right). \tag{32}$$

In addition, by Lemma 8.4,

$$(1-\epsilon)^2 x_i^\top L^\dagger x_i \leq \|Bz_i\|^2 \leq (1+\epsilon)^2 x_i^\top L^\dagger x_i \tag{33}$$

holds with probability $1 - \frac{1}{N}$. Combining (32) and the sum of (33) over i, we obtain

$$(1-\epsilon)^3 \mathrm{Tr}\left(L^\dagger\right) \leq \frac{1}{M} \sum_{i=1}^{M} \|Bz_i\|^2 \leq (1+\epsilon)^3 \mathrm{Tr}\left(L^\dagger\right),$$

which implies (31). □

In addition to the high accuracy, Algorithm 1 is also efficient, as summarized in the following theorem.

Theorem 8.6. *The time cost of Algorithm 1 is $\widetilde{O}(E\epsilon^{-2})$. The space cost of Algorithm 1 is $O(E)$.*

Proof. We first prove the time complexity of Algorithm 1. Line 3 takes $O(N)$ time. For Line 4, to compute y_i fast, we can first evaluate $\mathbf{1}^\top D^{\frac{1}{2}} x_i$, which takes $O(N)$ time, since D is a diagonal matrix. Then, we compute $D^{\frac{1}{2}} x_i$ and $D\mathbf{1}\mathbf{1}^\top D^{\frac{1}{2}} x_i$ and their difference, which also takes $O(N)$ time. So Line 4 takes $O(N)$ time. Line 5 takes $\widetilde{O}(E)$ time. And using sparse matrix multiplication, line 6 takes $O(E)$ time, since B has $2E$ non-zero entries. So lines 2-6 take $\widetilde{O}(M \times E) = \widetilde{O}(E\epsilon^{-2})$. Line 7 takes $O(M) = \widetilde{O}(\epsilon^{-2})$. Therefore, the overall time cost is $\widetilde{O}(E\epsilon^{-2})$.

We continue to prove the space cost. We need $O(E)$ to store the original graph $\mathcal{G}$. It takes $O(N)$ to store matrix D and $O(E)$ to store matrix B. Lines 3-5 take $O(N)$ space to store vectors x_i, y_i and z_i, respectively. Line 6 takes at most $O(M) = \widetilde{O}(\epsilon^{-2})$ space. Therefore, the overall space cost is $O(E)$, with the smaller terms being omitted. This completes the proof of the space complexity. □

8.2 Experimental Results

To demonstrate the performance of Algorithm 1, we use it to compute the Kemeny constant for some real and model networks. We perform all experiments on a machine with 4-core 4.2GHz Intel i7-7700K CPU and with 32GB of RAM. The approximation algorithm was implemented in *Julia v0.6.0*, where the `LaplSolve` is from [30].

8.2.1 Results for Real Networks. We first demonstrate the efficiency and scalability of Algorithm 1 for approximating the Kemeny constant of some real networks, by comparing with the exact algorithm given in (2) through directly computing eigenvalues of the normalized Laplacian matrix. Related information about the studied real networks and their LCC is shown in Tables 1 and 2. Table 3 reports the computational time of Algorithm 1 and the exact algorithm. From Table 3, we observe that for all chosen parameter ϵ, the running time for Algorithm 1 is much less than that for the exact algorithm, particularly for those relatively large tested networks. Note that for the large networks in Table 2, due to the limits of memory and time we cannot run the exact algorithm on the machine. In contrast, we can obtain the approximation of the Kemeny constant for those networks by using Algorithm 1, indicating the efficiency and scalability of our proposed algorithm.

We then demonstrate the accuracy of Algorithm 1 for approximating the Kemeny constant of real networks. To this end, we

Table 2: Statistics of partial datasets used in our experiments. For a network with N vertices and E edges, we denote the number of vertices and edges in its largest connected component by N' and E', respectively.

Network	N	E	N'	E'
Livemocha	104,103	2,193,083	104,103	2,193,083
WordNet	146,005	656,999	145,145	656,230
Gowalla	196,591	950,327	196,591	950,327
com-DBLP	317,080	1,049,866	317,080	1,049,866
Amazon	334,863	925,872	334,863	925,872
Pennsylvania	1,088,092	1,541,898	1,087,562	1,541,514
roadNet-TX	1,379,917	1,921,660	1,351,137	1,879,201

Table 3: The running time (seconds, s) of Algorithm 1 and the exact algorithm (Exact) with various ϵ on real networks.

Network	Exact	Algorithm 1 (s) with various ϵ					
	(s)	0.3	0.25	0.2	0.15	0.1	0.05
Hamsterster friendships	0.103	0.211	0.231	0.354	0.636	1.402	5.268
Protein	0.074	1.311	0.086	0.129	0.225	0.493	1.919
Hamster full	0.124	0.271	0.311	0.481	0.867	1.735	7.233
Human protein (Vidal)	0.311	0.152	0.223	0.318	0.524	1.233	4.728
Route views	2.874	0.277	0.370	0.574	1.009	2.198	8.639
arXiv astro-ph	78.90	6.287	8.128	11.16	20.08	51.03	191.5
CAIDA	252.8	1.999	2.429	3.973	7.450	16.40	68.01
Internet topology	564.8	4.245	4.438	7.233	11.75	28.30	104.5
Brightkite	2729	11.07	13.70	21.08	37.72	83.10	296.3
Livemocha	–	86.63	125.0	183.4	313.6	748.8	2742
WordNet	–	37.81	54.19	83.16	161.1	325.3	1198
Gowalla	–	62.81	80.39	140.2	249.1	495.8	2163
com-DBLP	–	111.4	163.1	228.2	464.9	983.8	3839
Amazon	–	153.7	221.3	361.2	652.4	1400	5667
Pennsylvania	–	581.2	810.8	1355	2363	5178	20550
roadNet-TX	–	850.4	1215	1852	3268	7811	29118

compare the approximation results $\tilde{K}$ obtained by Algorithm 1 with the exact results K obtaind by (2). Table 4 reports the relative error $\rho = (K - \tilde{K})/K$ of Algorithm 1. One can see that for all ϵ and all networks, the actual relative errors are significantly small. Thus, except for the efficiency, Algorithm 1 also gives good approximation $\tilde{K}$ for the Kemeny constant K of real networks.

Table 4: Relative error ρ of Algorithm 1 ($\times 10^{-4}$).

Network	Relative error for various ϵ			
	0.3	0.2	0.1	0.05
Hamsterster friendships	26.0	14.7	5.25	1.03
Protein	26.9	18.6	54.0	4.90
Hamster full	36.2	17.4	3.35	7.82
Human protein (Vidal)	10.2	15.7	15.5	0.68
Route views	8.23	9.73	1.32	0.77
arXiv astro-ph	8.49	18.7	0.45	0.98
CAIDA	4.34	0.77	1.78	0.91
Internet topology	0.95	1.43	1.31	0.48
Brightkite	3.66	2.80	0.48	0.38

8.2.2 Results for Model Networks. We continue to evaluate the performance of Algorithm 1 by using it to compute the Kemeny constant for model networks, including the Apollonian network and extended pseudofractal networks with $m = 1$ and $m = 2$. The numerical results are reported in Table 5. For the three considered networks, the numbers of nodes are more than one million, with the number of nodes in the third network being almost three million. For each of the three networks, the computation time is lower than 46 minutes, with the relative error less than 0.001, which indicates that the approximation algorithm works effectively for all considered networks. This again demonstrates the advantage of our algorithm for large-scale networks.

9 CONCLUSIONS

As a fundamental quantity for random walks on networks, the Kemeny constant has found broad applications in different areas. For example, it was proved to be a useful indicator measuring network criticality, the efficiency of stochastic robotic surveillance strategies in network environments, as well as the efficiency of navigation on the Web, with the low Kemeny constant representing high efficiency. In these contexts, constructing or finding networks with optimal Kemeny constant is of both theoretical and practical interest. It is known that among all networks with the same network size, the complete graph is the unique optimal network having the least Kemeny constant, the leading scaling of which is a linear function of the network size, with a slope being exactly 1. However, complete graphs cannot describe typical real-life networked systems, most of which are sparse and simultaneously scale-free and small-world, exhibiting power-law degree distribution and small distance.

In order to explore the behavior of the Kemeny constant on networks with scale-free small-world structure, in this paper we presented an extensive study of the Kemeny constant in some real-life scale-free networks, two sparse deterministic scale-free small-world networks, and the Barabási-Albert network. For all the studied networks, their Kemeny constants are low, which display a linear growth with the network size. Particularly, for each network, the ratio of the Kemeny constant to the number of nodes is constant, only a little greater than 1. Thus, minimal scaling for Kemeny constant is similar to that of complete graphs can be achieved by sparse scale-free networks with constant average vertex degree. In addition, we developed a randomized algorithm that approximately computes the Kemeny constant for any connected graph in nearly linear time with respect to the number of edges. We experimentally demonstrated the accuracy and efficiency of our algorithm. Our work sheds light on the structure design of networks with small Kemeny constant, as well as fast and accurate computation of the Kemeny constant.

Table 5: Exact Kemeny constant K, their approximation $\tilde{K}$, relative error $\rho = (K - \tilde{K})/K$, and running time (seconds) for $\tilde{K}$ on networks the Apollonian network $\mathcal{A}_{12}$, extended pseudofractal network $\mathcal{F}_{13}$ with $m = 1$, and extended pseudofractal network $\mathcal{F}_9$ with $m = 2$, denoted by $\mathcal{F}_9'$. K is obtained via (16) and (25), while $\tilde{K}$ is obtained through Algorithm 1 with $\epsilon = 0.1$.

Network	Vertices	Edges	K	$\hat{K}$	Error ρ	Time
$\mathcal{A}_{12}$	1,062,884	3,188,646	1,415,634	1,415,576	0.000041	1736
$\mathcal{F}_{13}$	2,391,486	4,782,969	3,971,608	3,972,037	0.00011	1482
$\mathcal{F}_9'$	2,929,689	5,859,375	4,100,948	4,096,858	0.00099	2754

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No. 61872093, 61803248, U19A2066 and 61672166), the National Key R & D Program of China (No. 2018YFB1305104 and 2019YFB2101703), Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01) and ZJLab.

REFERENCES

[1] Ahmad Ali Abin. 2018. A Random Walk Approach to Query Informative Constraints for Clustering. *IEEE Trans. Cybern.* 48, 8 (2018), 2272–2283.
[2] Pushkarini Agharkar, Rushabh Patel, and Francesco Bullo. Dec. 2014. Robotic surveillance and Markov chains with minimal first passage time. In *Proc. IEEE Conf. Decision Control*. IEEE, 6603–6608.
[3] José S Andrade Jr, Hans J Herrmann, Roberto FS Andrade, and Luciano R Da Silva. 2005. Apollonian networks: simultaneously scale-free, small world, Euclidean, space filling, and with matching graphs. *Phys. Rev. Lett.* 94, 1 (2005), 018702.
[4] Haim Avron and Sivan Toledo. 2011. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM* 58, 2 (2011), 8.
[5] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.
[6] Joost Berkhout and Bernd F Heidergott. 2019. Analysis of Markov influence graphs. *Oper. Res.* 67, 3 (2019), 892–904.
[7] Enrico Bozzo. 2013. The Moore–Penrose inverse of the normalized graph Laplacian. *Linear Algebra Appl.* 439, 10 (2013), 3038–3043.
[8] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. 2008. Epidemic thresholds in real networks. *ACM Trans. Inform. Syst. Secur.* 10, 4 (2008), 13.
[9] M. Chen, J. Z. Liu, and X. Tang. 2008. Clustering via Random Walk Hitting Time on Directed Graphs. In *Proc. AAAI Conf. Artificial Intelligence*. 616–621.
[10] Fan RK Chung. 1997. *Spectral Graph Theory*. American Mathematical Society, Providence, RI.
[11] Marie Chupeau, Olivier Bénichou, and Raphaël Voituriez. 2015. Cover times of random searches. *Nat. Phys.* 11, 10 (2015), 844–847.

[12] Michael B Cohen, Jonathan Kelner, John Peebles, Richard Peng, Aaron Sidford, and Adrian Vladu. 2016. Faster algorithms for computing the stationary distribution, simulating random walks, and more. In *Proc. 57th Annual Symposium on Foundations of Computer Science*. IEEE, 583–592.

[13] S Condamin, O Bénichou, V Tejedor, R Voituriez, and J Klafter. 2007. First-passage times in complex scale-invariant media. *Nature* 450, 7166 (2007), 77–80.

[14] Pasquale De Meo, Fabrizio Messina, Domenico Rosaci, Giuseppe ML Sarné, and Athanasios V Vasilakos. 2017. Estimating graph robustness through the Randic index. *IEEE Trans. Cybern.* 48, 11 (2017), 3232–3242.

[15] Sergey N Dorogovtsev, Alexander V Goltsev, and José Ferreira F Mendes. 2002. Pseudofractal scale-free web. *Phys. Rev. E* 65, 6 (2002), 066122.

[16] Jonathan PK Doye and Claire P Massen. 2005. Self-similar disk packings as model spatial scale-free networks. *Phys. Rev. E* 71, 1 (2005), 016128.

[17] Abbas El Gamal, James Mammen, Balaji Prabhakar, and Devavrat Shah. 2006. Optimal throughput-delay scaling in wireless networks-part I: The fluid model. *IEEE Trans. Inf. Theory* 52, 6 (2006), 2568–2592.

[18] François Fouss, Marco Saerens, and Masashi Shimbo. 2016. *Algorithms and Models for Network Data and Link Analysis.* Cambridge University Press.

[19] Arpita Ghosh, Stephen Boyd, and Amin Saberi. 2008. Minimizing effective resistance of a graph. *SIAM Rev.* 50, 1 (2008), 37–66.

[20] Kai-Lung Hua, Hong-Cyuan Wang, Chih-Hsiang Yeh, Wen-Huang Cheng, and Yu-Chi Lai. 2018. Background extraction using random walk image fusion. *IEEE Trans. Cybern.* 48, 1 (2018), 423–435.

[21] Jeffrey J Hunter. 2006. Mixing times with applications to perturbed Markov chains. *Linear Algebra Appl.* 417, 1 (2006), 108–123.

[22] Jeffrey J Hunter. 2014. The role of Kemeny's constant in properties of Markov chains. *Commun. Stat. — Theor. Methods* 43, 7 (2014), 1309–1321.

[23] MF Hutchinson. 1989. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun. Stat.-Simul. Comput.* 18, 3 (1989), 1059–1076.

[24] Ali Jadbabaie and Alex Olshevsky. 2019. Scaling Laws for Consensus Protocols Subject to Noise. *IEEE Trans. Autom. Control* 64, 4 (2019), 1389–1402.

[25] Wen Jiang, De-Shuang Huang, and Shenghong Li. 2016. Random walk-based solution to triple level stochastic point location problem. *IEEE Trans. Cybern.* 46, 6 (2016), 1438–1451.

[26] Alafate Julaiti, Bin Wu, and Zhongzhi Zhang. 2013. Eigenvalues of normalized Laplacian matrices of fractal trees and dendrimers: Analytical results and applications. *J. Chem. Phys.* 138, 20 (2013), 204116.

[27] John G Kemeny and James Laurie Snell. 1976. *Finite Markov Chains.* Springer, New York.

[28] Steve Kirkland. 2010. Fastest expected time to mixing for a Markov chain on a directed graph. *Linear Algebra Appl.* 433, 11 (2010), 1988–1996.

[29] Jérôme Kunegis. 2013. KONECT: The Koblenz Network Collection. In *Proc. 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil). ACM, New York, USA, 1343–1350. https://doi.org/10.1145/2487788.2488173

[30] Rasmus Kyng and Sushant Sachdeva. 2016. Approximate Gaussian elimination for Laplacians-fast, sparse, and simple. In *Proc. IEEE 57th Annual Symposium on Foundations of Computer Science*. IEEE, 573–582.

[31] Bertrand Lebichot and Marco Saerens. 2018. A bag-of-paths node criticality measure. *Neurocomputing* 275 (2018), 224–236.

[32] Mark Levene and George Loizou. 2002. Kemeny's constant and the random surfer. *Am. Math. Mon.* 109, 8 (2002), 741–745.

[33] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. 2009. *Markov Chains and Mixing Times.* American Mathematical Society, Providence, RI.

[34] Xiaoli Li, Zhifeng Han, Lijun Wang, and Huchuan Lu. 2016. Visual tracking via random walks on graph model. *IEEE Trans. Cybern.* 46, 9 (2016), 2144–2155.

[35] Y. H. Li and Z.-L. Zhang. 2013. Random walks and Green's function on digraphs: A framework for estimating wireless transmission costs. *IEEE/ACM Trans. Netw.* 21, 1 (2013), 135–148.

[36] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. 2017. Random walks and diffusion on networks. *Phys. Rep.* 716-717 (2017), 1–58.

[37] M. E. J. Newman. 2003. The Structure and Function of Complex Networks. *SIAM Rev.* 45, 2 (2003), 167–256. https://doi.org/10.1137/S003614450342480

[38] Jae Dong Noh and Heiko Rieger. 2004. Random walks on complex networks. *Phys. Rev. Lett.* 92, 11 (2004), 118701.

[39] José Luis Palacios and José Miguel Renom. 2010. Bounds for the Kirchhoff index of regular graphs via the spectra of their random walks. *Int. J. Quantum Chem.* 110, 9 (2010), 1637–1641.

[40] Rushabh Patel, Pushkarini Agharkar, and Francesco Bullo. 2015. Robotic surveillance and Markov chains with minimal weighted Kemeny constant. *IEEE Trans. Autom. Control* 60, 12 (2015), 3156–3167.

[41] Yi Qi, Huan Li, and Zhongzhi Zhang. 2018. Extended corona product as an exactly tractable model for weighted heterogeneous networks. *Comput. J.* 61, 5 (2018), 745–760.

[42] Yi Qi and Zhongzhi Zhang. 2019. Spectral properties of extended Sierpiński graphs and their applications. *IEEE Trans. Netw. Sci. Eng.* 6, 3 (2019), 512–522.

[43] C Vallejos, A Reinaldo, V Martínez, and M José. 2014. A Fast Transformation of Markov Chains and Their Respective Steady-State Probability Distributions. *Comput. J.* 57, 1 (2014), 1–11.

[44] Piet Van Mieghem, Jasmina Omic, and Robert Kooij. 2009. Virus spread in networks. *IEEE/ACM Trans. Netw.* 17, 1 (2009), 1–14.

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

[46] G H Weiss. 1994. *Aspects and Applications of the Random Walk.* North-Holland, Amsterdam.

[47] Scott White and Padhraic Smyth. 2003. Algorithms for estimating relative importance in networks. In *Proc. Int. Conf. Knowledge Discovery and Data Mining*. 266–275.

[48] Pinchen Xie, Yuan Lin, and Zhongzhi Zhang. 2015. Spectrum of walk matrix for Koch network and its application. *J. Chem. Phys.* 142, 22 (2015), 224106.

[49] Pinchen Xie, Zhongzhi Zhang, and Francesc Comellas. 2016. On the spectrum of the normalized Laplacian of iterated triangulations of graphs. *Appl. Math. Comput.* 273 (2016), 1123–1129.

[50] Pinchen Xie, Zhongzhi Zhang, and Francesc Comellas. 2016. The normalized Laplacian spectrum of subdivisions of a graph. *Appl. Math. Comput.* 286 (2016), 250–256.

[51] Yuhao Yi, Zhongzhi Zhang, Yuan Lin, and Guanrong Chen. 2015. Small-world topology can significantly improve the performance of noisy consensus in a complex network. *Comput. J.* 58, 12 (2015), 3242–3254.

[52] Yuhao Yi, Zhongzhi Zhang, and Stacy Patterson. 2020. Scale-free loopy structure is resistant to noise in consensus dynamics in power-law graphs. *IEEE Trans. Cybern.* 50, 1 (2020), 190–200.

[53] Yuhao Yi, Zhongzhi Zhang, Liren Shan, and Guanrong Chen. 2017. Robustness of first-and second-order consensus algorithms for a noisy scale-free small-world Koch network. *IEEE Trans. Control Syst. Technol.* 25, 1 (2017), 342–350.

[54] ZhongZhi Zhang, XiaoYe Guo, and Yuan Lin. 2014. Full eigenvalues of the Markov matrix for scale-free polymer networks. *Phys. Rev. E* 90, 2 (2014), 022816.

[55] Zhongzhi Zhang, Zhengyi Hu, Yibin Sheng, and Guanrong Chen. 2012. Exact eigenvalue spectrum of a class of fractal scale-free networks. *Europhys. Lett.* 99, 1 (2012), 10007.

[56] Zhongzhi Zhang, Lili Rong, and Shuigeng Zhou. 2006. Evolving Apollonian networks with small-world scale-free topologies. *Phy. Rev. E* 74, 4 (2006), 046105.

[57] Zhongzhi Zhang, Lili Rong, and Shuigeng Zhou. 2007. A general geometric growth model for pseudofractal scale-free web. *Physica A* 377, 1 (2007), 329–339.

[58] Bo Zhou and Nenad Trinajstić. 2009. On resistance-distance and Kirchhoff index. *J. Math. Chem.* 46, 1 (2009), 283–289.