Searching for Key Cycles in a Complex Network

Siyang Jiang,[1] Jin Zhou,[1,2,*] Michael Small,[3,4] Jun-an Lu,[1] and Yanqi Zhang[1]
[1]School of Mathematics and Statistics, Wuhan University, Hubei 430072, China
[2]Hubei Key Laboratory of Computational Science, Wuhan University, Hubei 430072, China
[3]The Complex Systems Group, Department of Mathematics and Statistics, University of Western Australia, Crawley, Western Australia 6009, Australia
[4]Mineral Resources, CSIRO, Kensington 6151, Western Australia

(Received 13 October 2022; revised 7 April 2023; accepted 13 April 2023; published 2 May 2023)

Searching for key nodes and edges in a network is a long-standing problem. Recently cycle structure in a network has received more attention. Is it possible to propose a ranking algorithm for cycle importance? We address the problem of identifying the key cycles of a network. First, we provide a more concrete definition of importance—in terms of Fiedler value (the second smallest Laplacian eigenvalue). Key cycles are those that contribute most substantially to the dynamical behavior of the network. Second, by comparing the sensitivity of Fiedler value to different cycles, a neat index for ranking cycles is provided. Numerical examples are given to show the effectiveness of this method.

DOI: 10.1103/PhysRevLett.130.187402

In the field of complex networks, two topics have recently attracted substantial attention. One is the structure of a network, including nodes [1,2], edges [3,4], cycles [5–7], simplicial complexes [8,9], and other higher order structures. The other is the dynamics on a network through phenomena such as synchronization [10–12], control [13], spread [14], games [15] and multi-intelligence consistency [16].

The structure of a network influences the dynamics on that network. It is vital to have a clear understanding of the basic network structure—the most basic structures of a network are nodes and edges. For practical purposes different ranking methods of node importance and edge importance have been given in, for example [1–4]. Identifying key nodes and edges provides a new mechanisms with which to understand the structure and dynamical behavior of a network. A cycle in a network is simply defined as a closed (nonrepeating) path with the same starting and ending node. Cycle structure is an important component both for networks as well as higher-order interaction. The number of cycles has been computed in scale-free networks [5] and various real networks [6]. Meanwhile, cycle structure promotes network function in many ways. Node centrality defined by the cycle structure performs well in spreading and control processes [7]. Cycles are the dominant contributors to information storage capability [17]. Networks with cycle structure have optimal synchronizability [18]. In addition, cycle structure plays an significant role in higher-order networks. The triangle (the simplest cycle) is the most basic structure of a high-order network structures [8].

The Laplacian matrix is a powerful tool in the research of dynamics—in particular, the second smallest Laplacian eigenvalue (also known as algebraic connectivity or Fiedler value [19]) plays an important role. Synchronization of a complex network is a collective dynamic of all nodes in the network. Synchronizability of a network can be measured by the Fiedler value [20]. The larger the value, the stronger the synchronizability. Diffusion of a network refers to the phenomenon that the concentration of the substance reaches uniform distribution within the network. The larger the Fiedler value, the faster the rate of diffusion [21].

In this Letter, we show that the cycle structure can increase the Fiedler value. Furthermore, we propose a cycle importance ranking method—the key cycles are those that contribute most substantially to dynamical behavior. As a result, all the cycle ranking indices in a network can be calculated by the Fiedler vector [22], which is the unit eigenvector corresponding to the second smallest Laplacian eigenvalue.

Consider an undirected graph G with uniform initial edge weight of 1. Let $\lambda_2(G)$ and $\mathbf{x}_2(G)$ be the second smallest Laplacian eigenvalue (Fiedler value) and the unit eigenvector corresponding to the eigenvalue (Fiedler vector) respectively.

First, the contribution of cycles to the Fiedler value is explored. Let G be a connected graph. Introduce the edge adding operation $G^e = G + e$ and node hanging operation: $G^{ev} = G + ev$, which we abbreviate as the adding and hanging operations, respectively. An edge is included in an adding operation. Hanging operation adds a node and connects it to an existing node. The operations are depicted in Fig. 1.

Any connected graph has a spanning tree [23]. This tree can be generated by a sequence of hanging operations and the remaining edges are generated by the adding

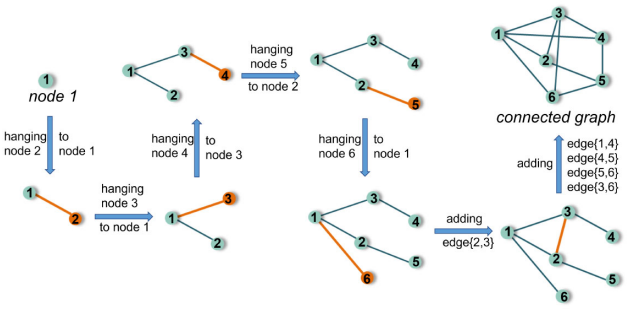


FIG. 1. The generation of a connected graph. Starting from node 1, hanging nodes 2, 3, 4, 5, and 6 in turn to get a spanning tree. Then adding other edges to get the connected graph.

operations. Hence any connected graph can be generated by a sequence of hanging and then adding operations. The generation of a connected graph is illustrated in Fig. 1. It is clear that cycles can be generated only by the adding operations.

To explain the importance of cycles, we analyze the variation of the Fiedler value caused by these two operations. Let G be a connected graph with n nodes. On the one hand, $\lambda_2(G) \le \lambda_2(G^e) \le \lambda_3(G)$ [24], where $G^e = G + e$. On the other hand, assume then that G^{ev} is obtained by adding an edge to G^v, where $G^v = G + v$ is an unconnected graph. Clearly, the eigenvalues of G^v are $0, 0, \lambda_2(G), \lambda_3(G), \ldots, \lambda_n(G)$. Therefore $\lambda_2(G) = \lambda_3(G^v)$, and G^{ev} can be generated by adding an edge to G^v. Then, one gets $\lambda_2(G^{ev}) \le \lambda_3(G^v) = \lambda_2(G)$. Hence, in a network, the hanging operation decreases the Fiedler value—while adding operation increases the Fiedler value.

For example, the Fiedler value of the chain network and the ring network with n nodes are $\lambda_2^{\text{chain}} = 4\sin^2(\pi/2n)$ and $\lambda_2^{\text{ring}} = 4\sin^2(\pi/n)$, respectively [25]. As a result, with the increase of n, λ_2^{chain} becomes smaller. Therefore, the hanging node operation decreases the Fiedler value. Only by adding an edge to the chain, which causes the generation of a ring (cycle), can the Fiedler value be increased. Further, $\lim_{n\to\infty}(\lambda_2^{\text{ring}}/\lambda_2^{\text{chain}}) = 4$. When the number of nodes is large enough, the second smallest Laplacian eigenvalue can be increased by a factor of 4.

Every time an edge is added in a connected graph, at least one cycle is introduced into the graph. The more edges that are added, the more cycles appear in a network, and the larger the Fiedler value. Therefore, the Fiedler value of a network gets greater with the increasing number of cycles.

Second, we search for the key cycles in a complex network. Let $C = \{c_i, i = 1, 2, \ldots, k\}$ be the set of all cycles in a network, where k is the number of cycles, c_i represents one of the cycles. A cycle c_i of order (length) l_i is defined as a closed loop composed of l_i edges and l_i nodes $n_1, n_2, n_3, \ldots, n_{l_i}$, denoted as $c_i = (n_1, n_2, n_3, \ldots, n_{l_i})$. It means cycle c_i composed of edge (n_1, n_2), edge $(n_2, n_3), \ldots$, edge (n_{l_i}, n_1). Denote the set of these edges as E_i. The graph after adding weight ϵ to a cycle c_i, namely,

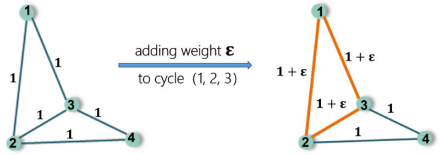


FIG. 2. Adding weight ϵ to cycle (1,2,3), namely adding weight ϵ to each edge of this cycle.

adding weight ϵ to each edge of this cycle, is denoted as $G_{c_i}^{\epsilon}$. For example, consider $c_1 = (1, 2, 3)$ in Fig. 2. The Laplacian matrix after adding weight ϵ is

$$L(G_{c_1}^{\epsilon}) = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} + \epsilon \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Denote the variation of the Fiedler value by $\delta\lambda_{c_i}^{\epsilon} = \lambda_2(G_{c_i}^{\epsilon}) - \lambda_2(G)$. We illustrate $\delta\lambda_{c_i}^{\epsilon} \ge 0$ when $\epsilon > 0$. According to Courant's theorem [19], the second smallest Laplacian eigenvalue is represented as $\lambda_2(G_{c_i}^{\epsilon}) = \min_{\mathbf{x}\perp\mathbf{1},\|\mathbf{x}\|=1} \mathbf{x}^T L(G_{c_i}^{\epsilon})\mathbf{x}$, where $\mathbf{1}$ is the column vector with all components being 1. Then,

$$\begin{aligned} \lambda_2(G_{c_i}^{\epsilon}) &= \min_{\mathbf{x}\perp\mathbf{1},\|\mathbf{x}\|=1} \mathbf{x}^T[L(G) + L(G_{c_i}^{\epsilon}) - L(G)]\mathbf{x} \\ &\ge \min_{\mathbf{x}\perp\mathbf{1},\|\mathbf{x}\|=1} \mathbf{x}^T L(G)\mathbf{x} \\ &\quad + \min_{\mathbf{x}\perp\mathbf{1},\|\mathbf{x}\|=1} \mathbf{x}^T[L(G_{c_i}^{\epsilon}) - L(G)]\mathbf{x}, \\ &= \lambda_2(G) + \min_{\mathbf{x}\perp\mathbf{1},\|\mathbf{x}\|=1} \mathbf{x}^T[L(G_{c_i}^{\epsilon}) - L(G)]\mathbf{x}. \end{aligned}$$

Since $L(G_{c_i}^{\epsilon}) - L(G)$ is a semi-positive definite matrix, it is easy to obtain $\delta\lambda_{c_i}^{\epsilon} \ge 0$ with $\epsilon > 0$. Similarly, $\delta\lambda_{c_i}^{\epsilon} \le 0$ with $\epsilon < 0$. Without loss of generality, we only discuss $\epsilon > 0$ hereafter.

Based on the variation of the Fiedler value, a new cycle ranking method is presented. The larger the $\delta\lambda_{c_i}^{\epsilon}$ ($\epsilon \ll 1$), and consequently the more key is the cycle c_i.

Now we use perturbation theory to approximate $\delta\lambda_{c_i}^{\epsilon}$. With $\epsilon \ll 1$, $\lambda_2(G_{c_i}^{\epsilon})$ and $\mathbf{x}_2(G_{c_i}^{\epsilon})$ are expressed as convergent power series [26]

$$\lambda_2(G_{c_i}^{\epsilon}) = \lambda_2(G) + k_1\epsilon + k_2\epsilon^2 + k_3\epsilon^3 + \cdots \tag{1}$$

and

$$\begin{aligned} \mathbf{x}_2(G_{c_i}^{\epsilon}) &= \mathbf{x}_2(G) + (t_{11}\epsilon + t_{12}\epsilon^2 + \cdots)\mathbf{x}_1(G) \\ &\quad + (t_{31}\epsilon + t_{32}\epsilon^2 + \cdots)\mathbf{x}_3(G) + \cdots \\ &\quad + (t_{n1}\epsilon + t_{n2}\epsilon^2 + \cdots)\mathbf{x}_n(G), \end{aligned} \tag{2}$$

respectively, where $\mathbf{x}_\alpha(G)$ are the unit eigenvectors corresponding to $\lambda_\alpha(G)$, $\alpha = 1, 2, \ldots, n$, k_β and $t_{\gamma\beta}$ are some constants, $\beta = 1, 2, \ldots, \gamma = 1, 3, \ldots, n$. The definitions of eigenvalue and eigenvector are

$$L(G^\epsilon_{c_i})\mathbf{x}_2(G^\epsilon_{c_i}) = \lambda_2(G^\epsilon_{c_i})\mathbf{x}_2(G^\epsilon_{c_i}). \tag{3}$$

Substituting Eq. (1) and Eq. (2) into Eq. (3), one obtains

$$\begin{aligned}&[L(G) + L(G^\epsilon_{c_i}) - L(G)][\mathbf{x}_2(G) + (t_{11}\epsilon \\ &\quad + t_{12}\epsilon^2 + \cdots)\mathbf{x}_1(G) + \cdots + (t_{n1}\epsilon + t_{n2}\epsilon^2 + \cdots)\mathbf{x}_n(G)] \\ &= (\lambda_2(G) + k_1\epsilon + k_2\epsilon^2 + \cdots)[\mathbf{x}_2(G) + (t_{11}\epsilon \\ &\quad + t_{12}\epsilon^2 + \cdots)\mathbf{x}_1(G) + \cdots + (t_{n1}\epsilon + t_{n2}\epsilon^2 + \cdots)\mathbf{x}_n(G)].\end{aligned}$$

Since both ends of the equation are convergent power series, the coefficients of ϵ are equal [27]. Then one gets

$$\begin{aligned}&L(G)\left[\sum_{\alpha=1,3\ldots,n} t_{\alpha 1}\mathbf{x}_\alpha(G)\right] + \frac{1}{\epsilon}[L(G^\epsilon_{c_i}) - L(G)]\mathbf{x}_2(G) \\ &\quad = \lambda_2(G)\left[\sum_{\alpha=1,3\ldots,n} t_{\alpha 1}\mathbf{x}_\alpha(G)\right] + k_1\mathbf{x}_2(G).\end{aligned} \tag{4}$$

Since $\mathbf{x}_2(G)$ is orthogonal to the other eigenvectors, left multiply Eq. (4) by $\mathbf{x}_2^T(G)$ and get

$$k_1 = \frac{1}{\epsilon}\mathbf{x}_2^T(G)[L(G^\epsilon_{c_i}) - L(G)]\mathbf{x}_2(G). \tag{5}$$

Substituting Eq. (5) into Eq. (1) and ignoring $O(\epsilon^2)$, one gets

$$\begin{aligned}\delta\lambda^\epsilon_{c_i} &= \mathbf{x}_2^T(G)[L(G^\epsilon_{c_i}) - L(G)]\mathbf{x}_2(G) \\ &= \epsilon\sum_{(p,q)\in E_i}(x_p - x_q)^2,\end{aligned}$$

where E_i is the set of all edges of cycle c_i, x_p is the pth component of $\mathbf{x}_2(G)$.

We see that $\epsilon\sum_{(p,q)\in E_i}(x_p - x_q)^2$ is an approximation of $\delta\lambda^\epsilon_{c_i}$ when $\epsilon \ll 1$. The larger the $\epsilon\sum_{(p,q)\in E_i}(x_p - x_q)^2$, the more influential the cycle. The number of terms in $\epsilon\sum_{(p,q)\in E_i}(x_p - x_q)^2$ is equal to the order l_i of a cycle. For example, select cycle (1,2,3) and cycle (1,2,4,3) in Fig. 2. The result are $\epsilon[(x_1-x_2)^2+(x_2-x_3)^2+(x_3-x_1)^2]$ and $\epsilon[(x_1-x_2)^2+(x_2-x_4)^2+(x_4-x_3)^2+(x_3-x_1)^2]$, respectively, where x_p is the pth component of $\mathbf{x}_2(G)$. It is clear that the value of ϵ does not affect the ranking result—for the purpose of ranking, ϵ can be ignored.

Therefore, we propose a new cycle ranking index $I_{c_i} = \sum_{(p,q)\in E_i}(x_p - x_q)^2$ to measure the importance of a cycle, which is actually the slope with $\epsilon = 0$. That is to say, the variation of Fiedler value is measured by Fiedler vector. Accordingly, a ranking algorithm of cycle importance is proposed.

Algorithm: Cycle Rank.

1. Given a connected graph G.
2. Calculate Fiedler vector $\mathbf{x}_2(G)$.
3. Select a cycle c_i.
4. Calculate index $I_{c_i} = \sum_{(p,q)\in E_i}(x_p - x_q)^2$.
5. Back to 3. Until all the cycles have been ranked.

The larger the I_{c_i}, the more key the cycle.

Another intuitive knowledge of I_{c_i} is the contribution of cycle c_i to Fiedler value. Because $\lambda_2(G) = \mathbf{x}_2^T(G)L(G)\mathbf{x}_2(G) = \sum_{(p,q)\in\bar{E}}(x_p - x_q)^2$, where $\bar{E}$ denotes the set of all edges in G, the sum of the contribution of all the edges is $\lambda_2(G)$; while the sum of the contribution of the edges in a cycle is I_{c_i}.

To illustrate our results, a computationally generated network and an experimental network are shown to verify the effectiveness of the ranking index.

Figure 3 shows a network with 6 cycles. Different cycles are marked with different colors, and denoted as $c_1, c_2, ..., c_6$. According to the algorithm, we use the index I_{c_i} to rank the cycles, $I_{c_1}, I_{c_2}, ..., I_{c_6}$ are presented on the right part of Fig. 4. The index decreases from c_1 to c_6. We conduct sensitivity simulation to examine the accuracy of ranking results in Fig. 4. The curves represent $\delta\lambda^\epsilon_{c_i}$ with the variation of ϵ. It shows that the ranking results with $\delta\lambda^\epsilon_{c_i}$ and I_{c_i} are consistent when ϵ is small, which can be seen from the initial slopes of Fig. 4. As ϵ continues to increase, the curves intersect occasionally. Overall, the index I_{c_i} is effective even with large ϵ. Furthermore, it is interesting to see that a cycle with a high order is frequently more important than a cycle with a low order; while some low order cycles, such as the better-positioned cycle c_3 of order 3, is more significant than c_4 of order 5.

To verify the effectiveness of the proposed cycle-ranking index, the edge-based pinning synchronization control [28] by using Chua's circuits as the node dynamics is applied in the network in Fig. 3. The synchronization times by, respectively, controlling the edges in c_1, c_3, and c_6 are

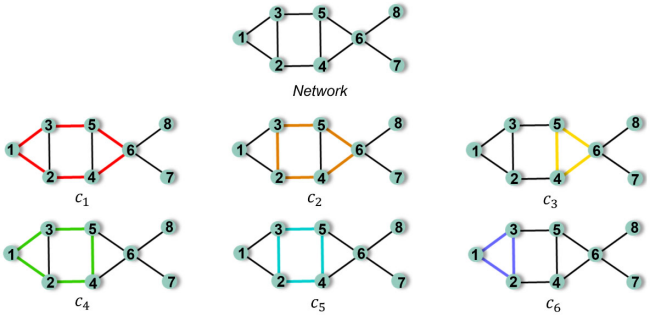


FIG. 3. A sample network with 6 cycles. Cycles are identified as $c_1, c_2, \ldots, c_6$ which are depicted by distinct colors.

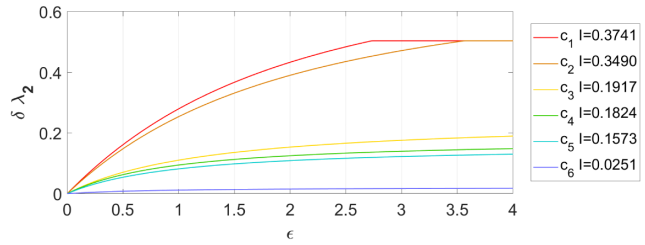


FIG. 4. The variation trend of $\delta\lambda_{c_i}^{\epsilon}$ versus the variation of $\epsilon \in [0, 4]$ in the sample network.

shown in Fig. 5. According to the ranking results, the importance of c_1, c_3, and c_6 decreases progressively one by one, which is in agreement with the time sequence of achieving synchronization.

Clearly, the position and the length of a cycle are two key factors in I_{c_i} to measure its importance. The position of c_i depends on the components of Fiedler vector, while the length of c_i depends on the number of summation terms. In addition, another normalized length index $\bar{I}_{c_i} = I_{c_i}/l_i$ can be considered, which takes the position into account. By $\bar{I}_{c_i}$, the priority levels of the cycles in Fig. 3 are c_2, c_3, c_1, c_5, c_4, and c_6 in sequence. It provides a different way to consider the cycle significance in a network, and may be applied to different potential applications.

The *C. elegans* metabolic network [29] is employed to rank the importance of the triangles. There are 453 nodes, 2025 edges, and 3284 triangles in the network. According to our algorithm, the five most important triangles and the five least important triangles are listed in Fig. 6(a). The locations of these triangles are shown in Fig. 6(b).

We find that node 56 appears frequently in key triangles, which results from the fact that the different value between the 56th component of the Fiedler vector and other components is large. Clearly, node 56 is significant.

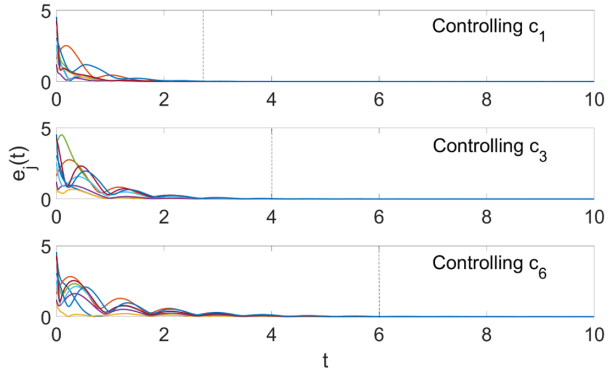
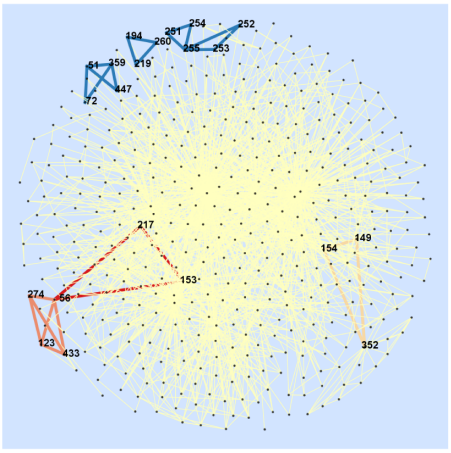


FIG. 5. The norm of error $e_j(t) = ||\xi_j(t) - s(t)||_2$, $1 \le j \le 8$ versus time t by, respectively, controlling cycles c_1, c_3, and c_6 in the network in Fig. 3, where $\xi_j(t)$ and $s(t)$ are the node state and the synchronization state respectively. The average error is obtained after 50 simulations by using initial state value range of $[-25, 25]$.

Triangle	I_{c_i}	Rank
$c_1 = (56, 153, 217)$	0.1536	1
$c_2 = (56, 123, 274)$	0.0506	2
$c_3 = (56, 274, 433)$	0.0506	2
$c_4 = (56, 123, 433)$	0.0506	2
$c_5 = (149, 154, 352)$	0.0040	5
...	...	...
$c_{3280} = (251, 254, 255)$	0	3284
$c_{3281} = (51, 72, 359)$	0	3284
$c_{3282} = (51, 359, 447)$	0	3284
$c_{3283} = (194, 219, 260)$	0	3284
$c_{3284} = (252, 253, 255)$	0	3284

(a) Ranking result



(b) Location

FIG. 6. The ranking result and the corresponding locations of triangles in the *C. elegans* metabolic network.

There may be a deep relationship between the key nodes and the key cycles, which will be discussed in our future work. Furthermore, the unimportant triangles tend to be among the nodes with low degrees.

Next, we put forward two strategies to speed up the network diffusion rate. We choose some triangles to simulate the variation of $\delta\lambda_{c_i}^{\epsilon}$ with the variation of ϵ in Fig. 7. It is found that c_1, c_2, c_3, and c_4 are obviously more important than other triangles. Weighting the triangles $c_5, c_6, ..., c_{3284}$, the Fiedler value is almost unchanged. Even though c_1, c_2, c_3, and c_4 are influential, they only

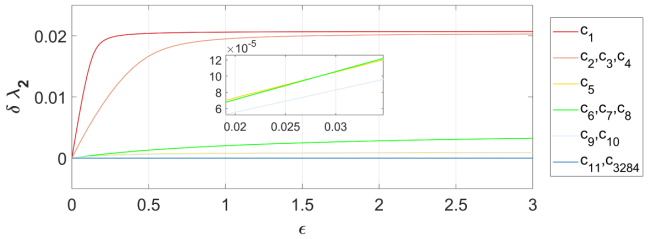


FIG. 7. The variation trend of $\delta\lambda_{c_i}^{\epsilon}$ versus the variation of $\epsilon \in [0, 3]$ in the *C. elegans* metabolic network. The $c_1, \ldots, c_{11}, c_{3284}$ are shown. The ranking results with $\delta\lambda_{c_i}^{\epsilon}$ and I_{c_i} are consistent when $\epsilon < 0.025$.

increase the original Fiedler value 0.258 by 8% at most. That is to say, if we want to more significantly increase the Fiedler value, it is not possible by changing only weighting. As a result, in the *C. elegans* metabolic network, to slightly speed up the diffusion rate, an effective way is to increase the weight of c_1, c_2, c_3, or c_4; to greatly speed up the diffusion rate, one should add new edges to generate new triangles.

In summary, a new cycle ranking approach has been proposed, which is obtained by comparing the sensitivity of Fiedler value to different cycles. Our results indicate that the importance of all the cycles in a network are calculated by Fiedler vector. The proposed ranking index is simple and effective, which can be extended to the importance of any structure in weighted or high-order networks such as edge or tetrahedron, and is expected to be applied in the areas of consensus, diffusion, synchronization, and control.

This work is supported by the National Natural Science Foundation of China under Grants No. 62173254 and No. 62176099.

*Corresponding author.
jzhou@whu.edu.cn

[1] L. Y. Lü, D. B. Chen, X. L. Ren, Q. M. Zhang, Y. C. Zhang, and T. Zhou, Phys. Rep. **650**, 1 (2016).
[2] H. Liu, X. H. Xu, J. A. Lu, G. R. Chen, and Z. G. Zeng, IEEE Trans. Syst. Man Cybern. **51**, 786 (2021).
[3] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).
[4] T. Bröhl and K. Lehnertz, Chaos **29 3**, 033115 (2019).
[5] G. Bianconi and A. Capocci, Phys. Rev. Lett. **90**, 078701 (2003).
[6] H. J. Kim and J. M. Kim, Phys. Rev. E **72**, 036109 (2005).
[7] T. L. Fan, L. Y. Lü, D. H. Shi, and T. Zhou, Commun. Phys. **4**, 1 (2021).
[8] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, Phys. Rep. **874**, 1 (2020).
[9] D. H. Shi, Z. F. Chen, X. Sun, Q. H. Chen, C. Ma, Y. Lou, and G. R. Chen, Commun. Phys. **4**, 1 (2021).
[10] P. S. Skardal, D. Taylor, and J. Sun, Phys. Rev. Lett. **113**, 144101 (2014).
[11] S. B. Zhu, J. Zhou, J. H. Lü, and J. A. Lu, IEEE Trans. Autom. Control **66**, 3550 (2021).
[12] S. B. Zhu, J. Zhou, X. H. Yu, and J. A. Lu, IEEE Trans. Syst. Man Cybern. **52**, 3342 (2022).
[13] J. Zhou, J. Chen, J. A. Lu, and J. H. Lü, IEEE Trans. Autom. Control **62**, 3468 (2017).
[14] A. Nematzadeh, E. Ferrara, A. Flammini, and Y.-Y. Ahn, Phys. Rev. Lett. **113**, 088701 (2014).
[15] F. C. Santos and J. M. Pacheco, Phys. Rev. Lett. **95**, 098104 (2005).
[16] J. Zhou, X. Q. Wu, W. W. Yu, M. Small, and J. A. Lu, Systems and Control Letters **61**, 195 (2012).
[17] J. T. Lizier, F. M. Atay, and J. Jost, Phys. Rev. E **86**, 026110 (2012).
[18] G. R. Chen, IEEE-CAA J. Autom. **9**, 573 (2022).
[19] M. Fiedler, Czech. Math. J. **23**, 298 (1973).
[20] X. F. Wang and G. R. Chen, Int. J. Bifurcation Chaos Appl. Sci. Eng. **12**, 187 (2002).
[21] S. Gómez, A. Díaz-Guilera, J. Gómez-Gardeñes, C. J. Pérez-Vicente, Y. Moreno, and A. Arenas, Phys. Rev. Lett. **110**, 028701 (2013).
[22] M. Fiedler, Czech. Math. J. **25**, 619 (1975).
[23] R. J. Wilson, *Introduction to Graph Theory* (Longman Press, London, 1979).
[24] R. Grone, R. Merris, and V. S. Sunder, SIAM J. Matrix Anal. Appl. **11**, 218 (1990).
[25] P. V. Mieghem, *Graph Spectra for Complex Networks* (Cambridge University Press, Cambridge, England, 2010).
[26] J. H. Wilkinson, *The Algebraic Eigenvalue Problem* (Clarendon Press, Oxford, 1965).
[27] W. Rudin, *Principles of Mathematical Analysis* (McGraw-Hill Press, New York, 1976).
[28] W. W. Yu, P. DeLellis, G. R. Chen, M. di Bernardo, and J. Kurths, IEEE Trans. Autom. Control **57**, 2153 (2012).
[29] R. A. Rossi and N. K. Ahmed, Proc. AAAI Conf. Artif. Intell. **29** (2015), 10.1609/aaai.v29i1.9277.