Finite-Time Distributed Topology Design for Optimal Network Resilience

Dong Xue[a,*], Sandra Hirche[a]

[a]*Chair of Information-Oriented Control, Technical University of Munich, Munich, Germany*

Abstract

The process of enhancing the ability of a complex network against various malicious attacks through link addition/rewiring has been the subject of extensive interest and research. The performance of existing methods often highly depends on full knowledge about the network topology. In this article, we devote ourselves to developing new distributed strategies to perform link manipulation sequentially using only local accessible topology information. This strategy is concerned with a matrix-perturbation-based approximation of the network-based optimization problems and a distributed algorithm to compute eigenvectors and eigenvalues of graph matrices. In addition, the development of a distributed stopping criterion, which provides the desired accuracy on the distributed estimation algorithm, enables us to solve the link-operation problem in a finite-time manner. Finally, all results are illustrated and validated using numerical demonstrations and examples.

1. Introduction

Recent years have witnessed a growing interest in the performance analysis of complex networks across a broad range of disciplines including mathematics, biology, physics, computer science, sociology, systems and control theory, and so on [1, 2, 3].

*Corresponding author
Email address: `dong.xue@ecust.edu.cn` (Dong Xue)

Preprint submitted to IET Control Theory & Applications *July 8, 2020*

Among other properties, the resilience of networks is of key importance in complex network analysis. Network resilience suggests the ability of a network to promptly recover thereafter to a stable state in the presence of external perturbations or structural damages. This phenomenon manifests itself in diverse domains. For example, in supply chain networks [4], network resilience lies in its possibility to maintain connectedness and operations under the loss of some ingredients or functions. Internet (communication networks) [5] is vulnerable to a wide range of challenges including software and hardware faults, human mistakes, and large-scale natural disasters. Other malicious attacks in realistic networks include terrorist attacks on transportation networks [6], parasitic species invasion in ecosystems [7], and cascade failures in power supply system [8]. Developing tools for the analysis and designing strategy to strength network resilience is necessary, not only from the engineering point of view but also for effective infrastructure construction and policy design [9].

The resilience of a network is close to the connectivity of networks: an intensively connected network is fundamentally resilient. Highly resilient networks as connectivity structure in real-life applications, for instance, help to accelerate the spread of information over sensor networks and to improve synchronization phenomenon in biological and engineering systems [10]. Two preventive approaches to enhance network resilience are to add new edges to or to rewire the existing edges in the network. The major challenge to determine between which pairs of nodes the new connections should be made. Numerous efforts have been directed towards this line of research [11, 12]. In general, the link operation problems can be formulated as an NP-hard optimization problem. This class of optimization problems is more likely to be solved by brute-force searching all possibility and selecting the best solutions. However, the computation complexity increases rapidly as the size of networks grows and quickly becomes unfeasible even for some moderate networks. About the computational complexity of problems in systems and control, please refer to the tutorial introduction [13]. As a result, extensive papers instead look for convex relaxation and heuristic algorithms to approximate the optimal solution. For example, a relaxation method based on semi-definite programming (SDP) can be applied to tackle this problem [14], whereas this method does not scale to massive networks. Moreover, various heuristics are proposed to tackled link operation problem in the literature [15, 16].

The aforementioned methods, however, performance rely a great deal on full information of network topology, which is difficult to accumulate in real large-scale networks due to, e.g., geographical constraints or privacy concerns. In an attempt to eliminate the dependence on high-level knowledge about the network, distributed methods using local computation and nearby communication appear in the recent literature. The authors in [11] propose a distributed algorithm for link addition by connecting the node of the minimum degree to a random another node. Despite needing only local information, i.e. node degree, this method gives the solution that is not necessarily optimal and especially, loses its effect in sparse networks. A criterion that adding links between nodes with the maximal deviation of the eigenvector components of graph Laplacian is proposed in [17]. Nevertheless, to obviate the network-wide computation, numerous distributed algorithms to estimate the eigenvectors of network matrix emerge in recent publications. For instance, a decentralized orthogonal iteration approach is proposed in [18] but with a centralized initialization. In the work [19], a decentralized power iteration (PI) algorithm is introduced to estimate the eigenvectors in a continuous-time setting, suffering from the challenging application in practice. Recent contributions that reside in developing distributed PI methods with nested loops (for decentralized intermediate normalization) can be found, e.g., in [20, 21]. Yet, this nesting design by means of consensus averaging (CA) algorithm severely affects the efficiency of these distributed methods in terms of communication limitation and convergence speed. This is due to the fact that reaching consensus for CA techniques requires an infinite amount of lower-level iterations. In addition to the aforecited paper, most of the (distributed) PI methods appearing in, e.g., [22, 23] converge in an asymptotic manner. For applicability to real scenarios, however, it is desirable that the convergence of link operation must do so in finite time rather than merely asymptotically.

The main contribution of this article is to develop a distributed strategy to provide a near-optimal solution to link operation problem under the inaccessibility of global network topology. Central in this framework is to operate a budget of links one after another instead of all at once. The rationale behind the sequentially adding/rewiring edges is the mimicking of the gradient-based approach. With the aid of matrix perturbation analysis, the primary optimization problem is approximated by a maximization problem involving the Fiedler vector of networks, thus avoiding NP-hardness. Since the approximation requires access to the entries of a network-wide vector, we propose a distributed algorithm, based on distributed power iteration and maximum-consensus to estimate the Fiedler vector. In this developed distributed algorithm, each node commits to estimate a single entry of the Fiedler vector and undertakes local computation using only limited knowledge of network topology. More fundamentally, we develop a distributed stopping criterion for the distributed computation mechanism, providing an explicit bound on the accuracy of the estimation algorithm. As such, the strategy of link operation for the enhancement of network resilience performs in a fully distributed and finite-time manner.

The remainder of this article is organized as follows. After introducing basic notions from graph theory and matrix algebra, the problem of interest is formulated in Section 2. The main results of this article including gradient-based approximation, distributed power iteration method, and stopping criteria for distributed computation are presented in Section 3. The developed strategies are demonstrated and its performance is evaluated via some numerical examples in 4. Finally, section 5 concludes this article and all proofs are given in the appendices.

2. Preliminaries and Problem Formulation

2.1. Basic Notations and Concepts

Let $\mathbb{R}$ ($\mathbb{R}_{>0}$) and $\mathbb{Z}$ ($\mathbb{Z}_{>0}$) be set of (positive) real numbers and (positive) integers, respectively. $\mathbf{1}$ ($\mathbf{0}$) denotes the column vector of all ones (zeros) with appropriate dimension. An identity matrix with dimensions inferred from context is given by $\boldsymbol{I}$. For a given set $\mathcal{C}$, $|\mathcal{C}|$ denotes the cardinality of this set. For a symmetric matrix $\boldsymbol{C}$, $\lambda_i(\boldsymbol{C})$ denotes its i-th maximal eigenvalues sorted in the increasing order $\lambda_1(\boldsymbol{C}) \leq \ldots \leq \lambda_n(\boldsymbol{C})$ and $\mathsf{sr}(\boldsymbol{C}) = \mathsf{max}_i |\lambda_i(\boldsymbol{C})|$ represents the spectral radius of matrix $\boldsymbol{C}$.

Consider a network of n nodes represented by a connected undirected graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ with a set of nodes $\mathbb{V} = \{1, \ldots, n\}$ and a set of edges $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$. In this article, we confine our attention to unweighted graphs for the sake of simplicity and the outcomes can be easily extended to weighted cases as discussed in [24]. If there exists an edge $(i, j) \in \mathbb{E}$ meaning $(j, i) \in \mathbb{E}$, then node i and j are neighboring. The complement graph of $\mathcal{G}$ is a graph $\overline{\mathcal{G}}$ with the same vertex set $\mathbb{V}$ as $\mathcal{G}$ and its edge set $\overline{\mathbb{E}}$ has an element $(i, j) \in \overline{\mathbb{E}}$ if and only if $(i, j) \notin \mathbb{E}$. The adjacency matrix of a graph is given by $\boldsymbol{A}(\mathcal{G}) \in \mathbb{R}^{n \times n}$ and its entries are defined by $[\boldsymbol{A}(\mathcal{G})]_{ij} = 1$ if $(j, i) \in \mathbb{E}$; otherwise 0. The *Laplacian* matrix of a graph is then given by $\boldsymbol{L}(\mathcal{G}) \triangleq \mathsf{diag}(\sum_{j=1}^{n} a_{1j}, \ldots, \sum_{j=1}^{n} a_{nj}) - \boldsymbol{A}(\mathcal{G})$ which has at least one eigenvalue at zero. A graph is connected if there is a path between any pair of distinct nodes. In particular, the second smallest eigenvalue, known as *algebraic connectivity*, assesses the connectedness of the graph, i.e., $\lambda_2(\boldsymbol{L}) > 0$ if $\mathcal{G}$ connected; otherwise disconnected. For the sake of convenience, we postulate that the spectrum of the connected graph Laplacian always satisfies $\lambda_2(\boldsymbol{L}) < \lambda_3(\boldsymbol{L})$ throughout this article. The normalized right eigenvector $\boldsymbol{\nu}_2(\boldsymbol{L}) := [\nu_2^1, \ldots, \nu_2^n]^{\mathsf{T}}$ corresponding to $\lambda_2(\boldsymbol{L})$, called *Fiedler vector*, is informative of the topological properties of networks. After assigning a label $l_{ij} \in \{1, \ldots, |\mathbb{E}|\}$ to the edge connecting node i and j, the Laplacian matrix $\boldsymbol{L}$ can be factorized by $\boldsymbol{L} = \sum_{l=1}^{|\mathbb{E}|} \boldsymbol{e}_l \boldsymbol{e}_l^{\mathsf{T}}$ where $\boldsymbol{e}_l = [e_l^1, \ldots, e_l^n]^{\mathsf{T}} \in \mathbb{R}^n$ is edge vector and for $l_{ij} \sim (j,i) \in \mathbb{E}$, its elements are defined as

$$\boldsymbol{e}_l^k = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } k = \{1, \dots, n\}.$$

Note that relabeling the edges does not change the analysis, so we sometimes drop the subscript in l_{ij} if there is no risk of confusion.

Furthermore, we say an $n \times n$ matrix $\boldsymbol{C} = [c_{ij}]$ is *compatible* with a graph $\mathcal{G}$ if $c_{ij} = 0$ iff $(j,i) \notin \mathbb{E}$ and $j \neq i$. Matrix $\boldsymbol{C}$ is nonnegative, i.e. $\boldsymbol{C} \geq 0$, if all its elements are nonnegative. A nonnegative symmetric matrix $\boldsymbol{C}$ is *irreducible* if and only if its compatible graph is connected. A symmetric matrix $\boldsymbol{C}$ is *primitive* if it is irreducible and has at least one positive diagonal element. A primitive symmetric matrix $\boldsymbol{C}$ has a simple largest eigenvalue and thus one has $\mathsf{sr}(\boldsymbol{C}) = \lambda_n(\boldsymbol{C})$.

In the article, the minimum/maximum-consensus algorithm serves as a key tool to design link-operation strategy, allowing all nodes to compute distributively the maximum and minimum of locally computable quantities, respectively. Specifically, individual node follows the update rule by

$$x_i(t+1) = \max_{j \in \{k | (k,i) \in \mathbb{E}\} \cup \{i\}} x_j(t), \tag{1}$$

$$x_i(t+1) = \min_{j \in \{k | (k,i) \in \mathbb{E}\} \cup \{i\}} x_j(t). \tag{2}$$

Compared with average consensus whose exact equilibrium can only be reached after an infinite amount of iterations, the iterative algorithms (1) and (2) converge in no more than n steps [25].

2.2. Problem Formulation

Given a connected undirected graph $\mathcal{G}_0 = (\mathbb{V}, \mathbb{E}_0)$ with the Laplacain $\boldsymbol{L}_0$, our goal is to add $m_a \geq 1$ number of links formulated by the set $\Delta\mathbb{E}^+ \subseteq \overline{\mathbb{E}}_0$ with $|\Delta\mathbb{E}^+| = m_a$, such that the algebraic connectivity of the resulting graph $\mathcal{G}_{m_a}^+ = (\mathbb{V}, \mathbb{E}_0 \cup \Delta\mathbb{E}^+)$ is maximized. As a motivating instantiation, the algebraic connectivity is of great significance to reflect the resilience of a network against random failures and targeted attacks. Furthermore, increasing the convergence rate of the consensus algorithm is another paradigmatic application of the theoretical treatments in this article [12]. Hence, one approach to enhance network resilience is by constructing additional interconnection in the network to increase its algebraic connectivity, which can be mathematically cast as

$$\begin{aligned} \max_{\Delta\mathbb{E}^+ \subseteq \overline{\mathbb{E}}_0} \quad & \lambda_2(\boldsymbol{L}^+_{m_a}) \\ \text{s.t.} \quad & |\Delta\mathbb{E}^+| = m_a, \end{aligned} \tag{P1a}$$

where $\boldsymbol{L}^+_{m_a}$ is the Laplacian associated to $\mathcal{G}^+_{m_a}$. With the edge labeling $\bar{l}_i j \sim (j,i) \in \overline{\mathbb{E}}_0$ on the complement graph $\overline{\mathcal{G}}_0$ and Laplacian factorization, the optimization problem (P1a) can be recast as

$$\begin{aligned} \max_{\bar{\boldsymbol{y}} \in \{0,1\}^{|\overline{\mathbb{E}}_0|}} \quad & \lambda_2(\boldsymbol{L}_0 + \Delta\boldsymbol{L}^+_{m_a}) \\ \text{s.t.} \quad & \Delta\boldsymbol{L}^+_{m_a} = \sum_{(j,i)\in\overline{\mathbb{E}}_0} \bar{y}_{\bar{l}_{ij}} \bar{\boldsymbol{e}}_{\bar{l}_{ij}} \bar{\boldsymbol{e}}^{\mathsf{T}}_{\bar{l}_{ij}}, \mathbf{1}^{\mathsf{T}}\bar{\boldsymbol{y}} = m_a, \end{aligned} \tag{P1b}$$

where $\bar{\boldsymbol{e}}_{\bar{l}_{ij}}$ is the edge vector adjunct to $\overline{\mathcal{G}}_0$ and the Boolean vector $\bar{\boldsymbol{y}} = [\bar{y}_1, \ldots, \bar{y}_{|\overline{\mathbb{E}}_0|}]^{\mathsf{T}}$ amounts to that $\bar{y}_{\bar{l}_{ij}} = 1$ means the edge $(j,i) \in \overline{\mathbb{E}}_0$ is selected to add into the graph $\mathcal{G}_0$.

Moreover, swapping a portion of existing links is another way to improve the network resilience [6]. Here, we reroute $m_r \geq 1$ number of existing edges in $\mathbb{E}_0$ to maximize the algebraic connectivity of the resulting graph $\widehat{\mathcal{G}}_{m_r} = (\mathbb{V}, \widehat{\mathbb{E}}_{m_r})$ where $|\widehat{\mathbb{E}}_{m_r}| = |\mathbb{E}_0|$. Analogously, one has the following optimization problem

$$\begin{aligned} \max_{\widehat{\mathbb{E}}_{m_r}} \quad & \lambda_2(\widehat{\boldsymbol{L}}_{m_r}) > 0 \\ \text{s.t.} \quad & |\widehat{\mathbb{E}}_{m_r} \cap \mathbb{E}_0| = |\mathbb{E}_0| - m_r, \ |\widehat{\mathbb{E}}_{m_r}| = |\mathbb{E}_0|, \end{aligned} \tag{P2a}$$

where $\widehat{\boldsymbol{L}}_{m_r}$ is the Laplacian of rewired graph $\widehat{\mathcal{G}}_{m_r}$. Likewise, we restate the optimization problem (P2a) as

$$
\begin{aligned}
\max_{\bar{\boldsymbol{y}},\boldsymbol{y}} \quad & \lambda_2(\boldsymbol{L}_0 + \Delta \boldsymbol{L}^+_{m_r} - \Delta \boldsymbol{L}^-_{m_r}) > 0 \\
\text{s.t.} \quad & \Delta \boldsymbol{L}^+_{m_a} = \sum_{(j,i)\in \overline{\mathbb{E}}_0} \bar{y}_{\bar{l}_{ij}} \bar{\boldsymbol{e}}_{\bar{l}_{ij}} \bar{\boldsymbol{e}}^{\mathsf{T}}_{\bar{l}_{ij}}, \qquad \boldsymbol{1}^{\mathsf{T}} \bar{\boldsymbol{y}} = m_r \\
& \Delta \boldsymbol{L}^-_{m_r} = \sum_{(j,i)\in \mathbb{E}^+_{m_r}} y_{l_{ij}} \boldsymbol{e}_{l_{ij}} \boldsymbol{e}^{\mathsf{T}}_{l_{ij}}, \quad \boldsymbol{1}^{\mathsf{T}} \boldsymbol{y} = m_r,
\end{aligned}
\tag{P2b}
$$

where $\bar{l}_{ij} \sim (j,i) \in \overline{\mathbb{E}}_0$, $\bar{\boldsymbol{y}} = [\bar{y}_1, \ldots, \bar{y}_{|\overline{\mathbb{E}}_0|}]^{\mathsf{T}} \in \{0,1\}^{|\overline{\mathbb{E}}_0|}$, $\boldsymbol{y} = [y_1, \ldots, y_{|\mathbb{E}_0|+m_r}]^{\mathsf{T}} \in \{0,1\}^{|\mathbb{E}_0|+m_r}$, and $l_{ij} \sim (j,i) \in \mathbb{E}^+_{m_r}$.

The optimization (P1b) and (P2b) are NP-hard whose global solution can be acquired by exhaustively searching all possibility or the suboptimal solutions based on heuristic [26]. However, those algorithms often depend fairly on prior knowledge of the entire network topology and also the existence of a central coordinator. Often, those prerequisites are unsubstantial in practice for several reasons, such as computational restrictions and privacy concerns. Hence, the remainder of this article focuses on developing a distributed computation algorithm to solve the optimization problem (P1b) and (P2b) individually by nodes based on the gather local information.

3. Main Results

In this section, we propose a distributed algorithm to (sub-)optimally solve the optimization problem (P1b)/(P2b). Especially, the resultant topology manipulation process is performed by individual nodes in the network and depends only on the neighboring information.

3.1. A Greedy Heuristic for Maximization

To circumvent the NP-hard nature, the matrix perturbation analysis [27] is first used to provide an approximation of the original problems. This eigenvalue sensitivity analysis shows the variation of matrix eigenvalues when the matrix is perturbed. For link addition, the matrix $\Delta \boldsymbol{L}^+_{m_a}$ can be treated as a perturbation imposing on $\boldsymbol{L}_0$. Then, a suboptimal solution to the optimization problem (P1b) can be obtained by solving the following approximation.

Lemma 1 (Link addition)**.** *Optimization problem (P1b) can be approximated as*

$$\begin{aligned} \max_{\bar{\boldsymbol{y}}} \quad & \sum_{(j,i)\in\overline{\mathbb{E}}_0} \bar{y}_{\bar{l}_{ij}} (\nu_2^i - \nu_2^j)^2 \\ & \mathbf{1}^\mathsf{T}\bar{\boldsymbol{y}} = m_a, \quad \bar{\boldsymbol{y}} \in \{0,1\}^{|\overline{\mathbb{E}}_0|} \end{aligned} \tag{P1c}$$

where $\bar{l}_{ij} \sim (i,j) \in \overline{\mathbb{E}}_0$ *and* $\boldsymbol{\nu}_2(\boldsymbol{L}_0) = [\nu_2^1, \ldots, \nu_2^n]^\mathsf{T}$ *is the eigenvector corresponding to the eigenvalue* $\lambda_2(\boldsymbol{L}_0)$*.*

Lemma 1 means to preferentially select links $\sim (j,i) \in \overline{\mathbb{E}}_0$ associted with large deviation $|\nu_2^i - \nu_2^j|$. Noteworthily, the aggregated squared difference of Fiedler vector elements, i.e., $\sum_{j=1}^n a_{ij}(\nu_2^i - \nu_2^j)^2$, appears as a metric to assess the criticality of individual node [28].

The approximation of the optimization problem (P2b) is in some way analogous to the link addition problem. More importantly, we postulate explicitly for the rewired graph being connected in the following approximated problem.

Lemma 2 (Link rewiring)**.** *The optimization (P2b) can be approximated as follows*

$$\begin{aligned} \max_{\bar{\boldsymbol{y}},\boldsymbol{y}} \quad & \sum_{(j,i)\in\overline{\mathbb{E}}_0} \bar{y}_{\bar{l}_{ij}} (\nu_2^i - \nu_2^j)^2 - \sum_{\in\mathbb{E}_{m_r}^+} y_{l_{ij}} (\mu_2^i - \mu_2^j)^2 \\ \text{s.t.} \quad & \mathbf{1}^\mathsf{T}\bar{\boldsymbol{y}} = \mathbf{1}^\mathsf{T}\boldsymbol{y} = m_r, \quad \lambda_2(\widehat{\boldsymbol{L}}_{m_r}) > 0, \end{aligned} \tag{P2c}$$

where $\bar{l}_{ij} \sim (j,i) \in \overline{\mathbb{E}}_0$, $l_{ij} \sim (j,i) \in \mathbb{E}_{m_r}^+$; $\boldsymbol{\nu}_2(\boldsymbol{L}_0) = [\nu_2^1, \ldots, \nu_2^n]^\mathsf{T}$ *and* $\boldsymbol{\mu}_2(\boldsymbol{L}_{m_r}^+) = [\mu_2^1, \ldots, \mu_2^n]^\mathsf{T}$ *specify the eigenvector corresponding to the second smallest eigenvalue of Laplacian matrix of graph* $\mathcal{G}_0$ *and graph* $\mathcal{G}_{m_r}^+$*, respectively.*

Thus far, the original combinatorial problems (P1b) and (P2b) are approximated respectively by (P1c) and (P2c) which are concerned with the eigenvector of interests. Nevertheless, the complete topological knowledge is indispensable for computing those eigenvectors in question, thus lying the barrier to performing the topology manipulation process in a distributed fashion.

3.2. Distributed Estimation of Eigenvectors

Power iteration (PI) method [29] has been widely used to estimate the simple largest eigenvalue and its associated eigenvector of a symmetric matrix $\boldsymbol{Q}$ in terms of

$$\tilde{\boldsymbol{z}}_n(t+1) = \frac{\boldsymbol{Q}\tilde{\boldsymbol{z}}_n(t)}{\|\boldsymbol{Q}\tilde{\boldsymbol{z}}_n(t)\|}, \tag{3}$$

where $\tilde{\boldsymbol{z}}_n(t)$ is the variable of PI estimator at time t with a non-zero initial vector $\tilde{z}_n(0)$. As time evolves, the sequence $\{\tilde{\boldsymbol{z}}_n(t)\}_{t\in\mathbb{R}_{\geq 0}}$ approaches asymptotically to the eigenvector corresponding to the principal eigenvalue $\lambda_n(\boldsymbol{Q})$ at the rate $|\lambda_{n-1}(\boldsymbol{Q})/\lambda_n(\boldsymbol{Q})|$. A comprehensive description and some refined variants of power iteration can be found in [30, 29] and the references therein.

One of the major challenges that arise in the application of PI approach for sizable networks is the intermediate normalization in each iteration step. Yet, this intermediate operation is necessary to settle the overflow problem, so that the estimates are unlikely to grow to infinity when $\lambda_n(\boldsymbol{Q}) > 1$ or shrink to zero when $\lambda_n(\boldsymbol{Q}) < 1$.

In the following, we develop a completely distributed PI algorithm. When introducing the general theory, we will drop the subscript of graphs and simply write $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ with the Laplacian $\boldsymbol{L}$. Before jumping into the details, we first define Perron matrix $\boldsymbol{P}$ of a graph by $\boldsymbol{P} = \boldsymbol{I}_n - \beta\boldsymbol{L}$, where the scaling factor satisfies $0 < \beta < 1/\tilde{n}$, ensuring the power iteration on Perron matrix converges to the non-negative dominant eigenvalue. Here, $\tilde{n}$ amounts to an estimate of the size of the network $\mathcal{G}$ or an upper bound on it. It is already known that $\tilde{n}$ can be computed in a distributed manner [31].

Since we aim to estimate the spectrum associated with the second smallest eigenvalue of Laplacian, a matrix deflation is conducted on Perron matrix as $\boldsymbol{Q} = \boldsymbol{P} - \boldsymbol{1}\boldsymbol{1}^\mathsf{T}/n$ whose dominant eigenvector coincides with the one associated to $\lambda_2(\boldsymbol{L})$. Besides symmetry and positive semi-definiteness, the deflated Perron matrix entails row and column sums equal to zero. The eigenstructures of Laplacian matrix, Perron matrix, and the deflated Perron matrix are summarized in Table 1 wherein the eigenvalue (e-value) λ_i and eigenvector (e-vector)

Table 1: Spectrum of matrices

	$\boldsymbol{L}$		$\boldsymbol{P}$		$\boldsymbol{Q}$	
i	e-value	e-vector	e-value	e-vector	e-value	e-vector
1	0	$\mathbf{1}/\sqrt{n}$	$1-\beta\lambda_n$	$\boldsymbol{\nu}_n$	0	$\mathbf{1}/\sqrt{n}$
2	λ_2	$\boldsymbol{\nu}_2$	$1-\beta\lambda_{n-1}$	$\boldsymbol{\nu}_{n-1}$	$1-\beta\lambda_n$	$\boldsymbol{\nu}_n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	λ_n	$\boldsymbol{\nu}_n$	1	$\mathbf{1}/\sqrt{n}$	$1-\beta\lambda_2$	$\boldsymbol{\nu}_2$

$\boldsymbol{\nu}_i$ for $i \in \mathbb{V}$ are in reference to Laplacian matrix $\boldsymbol{L}$.

Let $\tilde{\boldsymbol{\nu}}_2(t) = [\tilde{\nu}_2^1(t), \ldots, \tilde{\nu}_2^n(t)]^\mathsf{T}$ be the estimation variable of the Fiedler vector $\boldsymbol{\nu}_2$ of the graph $\mathcal{G}$ at iteration step t. After initializing with a random non-zero vector, each node updates its estimate by following the iterative principle,

$$\begin{aligned} &\tilde{\nu}_2^i(t+1) = h_i(t)/\alpha(t), \\ &h_i(t) = \tilde{\nu}_2^i(t) - \beta \sum_{j=1}^{n} [\boldsymbol{A}]_{ij} \left(\tilde{\nu}_2^i(t) - \tilde{\nu}_2^j(t) \right) - \frac{1}{n} \mathbf{1}^\mathsf{T} \tilde{\boldsymbol{\nu}}_2(t), \end{aligned} \tag{4}$$

where $\alpha(t) = \mathsf{max}_{i \in \mathbb{V}} |\sum_{j=1}^{n} [\boldsymbol{Q}]_{ij} \tilde{\nu}_2^i(t)|$ implies the adoption of the infinity norm to the iteration method (3). In the computation algorithm (4), the normalization factor α and the matrix deflation require that each node has access to some network-wide vectors. To obviate this requirement, one can first notice that

$$\mathbf{1}^\mathsf{T} \tilde{\boldsymbol{\nu}}_2(t) = \prod_{k=0}^{t} \frac{1}{\alpha(t-k)} \mathbf{1}^\mathsf{T} \boldsymbol{Q}^t \tilde{\boldsymbol{\nu}}_2(0) = 0, \quad \forall\, t \in \mathbb{Z}_{>0},$$

due to the fact that $\mathbf{1}$ is the left eigenvector of $\boldsymbol{Q}$ corresponding to the eigenvalue 0. We adjust slightly the initial condition to

$$\tilde{\boldsymbol{\nu}}_2(0) = \boldsymbol{L}\boldsymbol{p}, \quad \tilde{\nu}_2^i(0) = \sum_{j=1}^{n} [\boldsymbol{A}]_{ij} (p_i - p_j), \tag{5}$$

where $\boldsymbol{p} = [p_1, \ldots, p_n]^\mathsf{T}$ is a random non-zero vector, so one has $\mathbf{1}^\mathsf{T} \tilde{\boldsymbol{\nu}}_2(0) = 0$. The modified initialization can also be achieved in a distributed realization. Since the graph compatible with matrix $\boldsymbol{Q}$ is connected, the factor $\alpha(t)$ can be computed distributively by the max-consensus algorithm (1) with the initial condition

$$x_i(0) = \Big| \sum_{j=1}^{n} [\boldsymbol{Q}]_{ij} \tilde{\nu}_2^i(t) \Big|.$$

Thus, the update rule (4) can be implemented in a fully distributed fashion with the aid of the initialization (5) and max-consensus algorithm (1). As an inheritance of centralized PI method (3), the estimate $\tilde{\boldsymbol{\nu}}_2(t)$ converges exponentially to the desired eigenvector $\boldsymbol{\nu}_2(\boldsymbol{L})$ at the rate $(1-\beta\lambda_3(\boldsymbol{L}))/(1-\beta\lambda_2(\boldsymbol{L}))$.

3.3. Stopping Criteria for Distributed Power Method

In the application to real-life networks, however, the previously proposed distributed estimation technique, due to its asymptotic convergence nature, may be too slow to respond to e.g., the abruptness of cascade failures in power grids. In what follows, we endeavor to advance the applicability of the distributed algorithm (4) by providing a stopping criterion, while guaranteeing an ideal error tolerance of estimation. The textbook [30] provides a bound on the accuracy of the PI-based approximation of eigenvalues and eigenvectors for self-adjoint matrices. Therefore, we start off by tailing this stopping condition to accommodate the case of the positive semi-definite and symmetric matrices.

Proposition 1 (Centralized Stopping Criterion)**.** *Consider a positive semi-definite, irreducible, and symmetric matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and the PI method* (3) *with the initialization* (5)*. Given a scalar $\epsilon \in [0,1)$, if $\|\boldsymbol{\gamma}(t)\|_2 \le \epsilon |\tilde{\lambda}_2(t)|$, then*

$$\|\boldsymbol{\nu}_2 - \tilde{\boldsymbol{\nu}}_2(t)\|_2 \le 2\epsilon \left(1 + \frac{\epsilon}{1-\epsilon}\right) \frac{\lambda_n(\boldsymbol{Q})}{\lambda_n(\boldsymbol{Q}) - \lambda_{n-1}(\boldsymbol{Q})},$$

where $\tilde{\lambda}_2(t) := \tilde{\boldsymbol{\nu}}_2^{\mathsf{T}}(t)\boldsymbol{Q}\tilde{\boldsymbol{\nu}}_2(t)/\|\tilde{\boldsymbol{\nu}}_2^{\mathsf{T}}(t)\tilde{\boldsymbol{\nu}}_2(t)\|_2^2$ is an estimation of the algebraic connectivity of the graph $\mathcal{G}$ at step t and $\boldsymbol{\gamma}(t) := \tilde{\lambda}_2(t)\tilde{\boldsymbol{\nu}}_2(t) - \boldsymbol{Q}\tilde{\boldsymbol{\nu}}_2(t)$ denotes the residual of $\tilde{\boldsymbol{\nu}}_2(t)$ w.r.t. $\boldsymbol{Q}$.

In the implementation of this stopping criterion, the desideratum of the network-wide information $\tilde{\lambda}_2(t)$ hinders the distributed computation. The remainder of this subsection is devoted to explaining how this barrier can be circumvented.

At the outset, we introduce an auxiliary variable by

$$r_i(t) := \frac{\tilde{\nu}_2^i(t) h_i(t)}{\mathsf{max}(\delta, (\tilde{\nu}_2^i(t))^2)} \tag{6}$$

where the sufficiently small constant $0 < \delta << 1$ guarantees that the function $r_i(t)$ is well defined and entails an asymptotic convergence to the eigenvalue $\lambda_n(\boldsymbol{Q})$.

Lemma 3. *Consider the dynamics* (4) *evolving on a connected graph with the initialization* (5). *The auxiliary states converge asymptotically to a consensus configuration* $r_1(\infty) = \ldots = r_n(\infty) = r^*$ *and the consensus state* r^* *is equal to* $1 - \beta\lambda_2(\boldsymbol{L})$.

To this end, we get ready to present a distributed stopping criterion for the distributed power iteration, providing a bound on the efficiency of the estimation of the Fiedler vector.

Theorem 1. *Consider a connected graph* $\mathcal{G}$ *with the deflated Perron matrix* $\boldsymbol{Q}$ *and the distributed in-network computation* (3) *with the initialization* (5). *For a given threshold* $\epsilon \in [0, 1)$, *if*

$$\max_{i \in \mathbb{V}} |\rho_i| \leq \frac{\epsilon}{\sqrt{\tilde{n}}} \Big(\min_{i \in \mathbb{V}} |r_i(t)| \Big), \tag{7}$$

where ρ_i *is the* i*-th entry of the matrix-vector product*

$$\boldsymbol{\rho}(t) = ((\max_{i \in \mathbb{V}}(|r_i(t)| - 1))\boldsymbol{I}_n + \beta\boldsymbol{L} + \frac{\boldsymbol{1}\boldsymbol{1}^\mathsf{T}}{n})\tilde{\boldsymbol{\nu}}_2(t), \tag{8}$$

then the estimation error of $\tilde{\boldsymbol{\nu}}_2(t)$ *satisfies the following condition*

$$\|\boldsymbol{\nu}_2 - \tilde{\boldsymbol{\nu}}_2(t)\|_2 \leq 2\epsilon \left(1 + \frac{\epsilon}{1 - \epsilon}\right) \frac{1 - \beta\lambda_2(\boldsymbol{L})}{\beta(\lambda_3(\boldsymbol{L}) - \lambda_2(\boldsymbol{L}))}. \tag{9}$$

It is noted that $\boldsymbol{\rho}(t)$ in (8) reads to a modified residual of $\tilde{\boldsymbol{\nu}}_2$ w.r.t. the deflated Perron matrix $\boldsymbol{Q}$ and can be computed in a distributed fashion. With the help of max / min-consensus protocol -a fully distributed content- plays an important role in the in-network computation. Moreover, Lemma 3 exposes that $\lim_{t\to\infty} r_i(t) = 1 - \beta\lambda_2(\boldsymbol{L})$ for all $i \in \mathbb{V}$, facilitating agents to estimate algebraic connectivity in a distributed and finite-time manner. Thus, our distributed estimation algorithm together with the stopping criterion has a wider range of applications such as for the distributed connectivity detection and preservation in flocking control of multi-agent systems [32], and for the distributed link removal problem aiming at controlling the epidemic spreading [33].

3.4. The Complete Distributed Strategies for Link Operation

This subsection recapitulates the main results of this article. In a nutshell, the suboptimal solution to the optimization problem (P1b) can be computed distributively under the stopping criterion (7) using the distributed power method (4) and the initialization mechanism (5).

More importantly, we consider the problem of improving network resilience by sequentially adding/rewiring edges. The rationale behind the idea has been demonstrated in [33]. In simple terms, the number of links to be operated can be treated as the step-size in the gradient-based approach and a smaller step-size gives rise to a better approximation quality.

Proposition 2. *A sub-optimal solution to the link-addition problem (P1b) can be derived from solving the following distributed maximization problem along the sequence* $s \in \{1, \dots, m_a\}$

$$\begin{aligned} \max_{\bar{\boldsymbol{y}}(s)} \quad & \sum_{(j,i)\in\overline{\mathbb{E}}^{+}_{s-1}} \bar{y}_{\bar{l}_{ij}}(s)|\tilde{\nu}^{i}_{2}(s) - \tilde{\nu}^{j}_{2}(s)| \\ \text{s.t.} \quad & \mathbf{1}^{\mathsf{T}}\bar{\boldsymbol{y}}(s) = 1, \end{aligned} \tag{P1d}$$

where $\bar{\boldsymbol{y}}(s) = [\bar{y}_1(s), \dots, \bar{y}_{|\overline{\mathbb{E}}^{+}_{s-1}|}(s)]^{\mathsf{T}} \in \{0,1\}^{|\overline{\mathbb{E}}^{+}_{s-1}|}$*,* $\bar{l}_{ij}(s) \sim (j,i) \in \overline{\mathbb{E}}^{+}_{s-1}$*, and* $\tilde{\boldsymbol{\nu}}_2(s) = [\tilde{\nu}^1_2(s), \dots, \tilde{\nu}^n_2(s)]^{\mathsf{T}}$ *is the estimated Fiedler vector of the graph Laplacian* $\boldsymbol{L}^{+}_{s-1}$ *via the distributed PI computation* (4) *with the stopping criterion* (7)*.*

To retain the essential simplicity of what is going on, we omit the dependence of $\tilde{\boldsymbol{\nu}}_2(s)$ on the time argument t in Proposition 2. Thus, the problem (P1d) is a distributed convex optimization problem with a separable cost function along a sequence of issues (successive link-addition). Algorithm 1 presents the pseudo-code for the addition of a single link to the current network $\mathcal{G}_{s-1}^+$ at issue instant s. To sum up, the distributed strategy for multiple links addition invokes Algorithm 1 successively until m_a links are added.

Algorithm 1 Distributively adding one link to $\mathcal{G}_{s-1}^+$

Require: a connected graph $\mathcal{G}_{s-1}^+$, a scalar $\epsilon \in [0, 1)$, a random non-zero vector $\boldsymbol{p} \in \mathbb{R}^n$, and $t \leftarrow 0$

1: execute (5) with $\boldsymbol{p}$ to produce an initial condition $\tilde{\boldsymbol{\nu}}_2(s, 0)$
2: **while** $\max_{i\in\mathbb{V}} |\rho_i(s,t)| > \frac{\epsilon}{\sqrt{\tilde{n}}}\Big(\min_{i\in\mathbb{V}} |r_i(s,t)|\Big)$ **do**
3:     $t \leftarrow t + 1$
4:     $\tilde{\nu}_2^i(s,t) \leftarrow h_i(s, t-1)/\alpha(s, t-1)$
5:     $h_i(s,t) \leftarrow \tilde{\nu}_2^i(s,t) - \beta \sum_j [\boldsymbol{A}(s-1)]_{ij} \Big(\tilde{\nu}_2^i(s,t) - \tilde{\nu}_2^j(s,t)\Big) - \frac{\sum_j \tilde{\nu}_2^j(s,t)}{n}$
6:     $\alpha(s,t) \leftarrow \max_{i\in\mathbb{V}} |\sum_{j=1}^n [\boldsymbol{Q}(s-1)]_{ij} \tilde{\nu}_2^i(s,t)|$
7: **end while**
8: node assigns $\boldsymbol{\pi}_i(s) \leftarrow \tilde{\nu}_2^i(s)\boldsymbol{e}_i$ and transmits to neighbors
9: **if** $[\boldsymbol{\pi}_i(s)]_j \neq 0,\ \forall i, j \in \mathbb{V}$ **then**
10:     **break**
11: **else**
12:     **for** $j \in \{k | (i,k) \in \mathbb{E}_{s-1}^+\}$ **do**
13:         **for** $k = 1 : 1 : n$ **do**
14:             **if** $[\boldsymbol{\pi}_i(s)]_k == 0\ \&\ [\boldsymbol{\pi}_j(s)]_k \neq 0$ **then**
15:                 $[\boldsymbol{\pi}_i(s)]_k \leftarrow [\boldsymbol{\pi}_j(s)]_k$
16:             **end if**
17:         **end for**
18:     **end for**
19:     back to step 9
20: **end if**
21: $\bar{l}_{ij^*}(s) \leftarrow \arg\max_{j\in\{k|(i,k)\notin\mathbb{E}_{s-1}^+\}} |[\boldsymbol{\pi}_i]_i(s) - [\boldsymbol{\pi}_i]_j(s)|$
22: compute $\bar{l}_{i^*j^*}(s) \leftarrow \arg\max |[\boldsymbol{\pi}_i(s)]_i - [\boldsymbol{\pi}_i(s)]_{j^*}|$ for (i, j^*) using max-consensus (1) with $x_i(0) \leftarrow |[\boldsymbol{\pi}_i(s)]_i - [\boldsymbol{\pi}_i(s)]_{j^*}|$
23: output $\mathcal{G}_s^+ \leftarrow (\mathbb{V}, \mathbb{E}_{s-1}^+ \cup \{(i^*, j^*)\})$

In analogy to link-addition procedure, we can develop a distributed strategy to rewire the existing links in the network.

Proposition 3. *Along the sequence $s \in \{1, \dots, m_r\}$, solving the following optimization problem sequentially and distributively*

$$\begin{aligned} \max_{\bar{\boldsymbol{y}}(s), \boldsymbol{y}(s)} \quad & \sum_{(j,i) \in \widehat{\mathbb{E}}_{s-1}} \bar{y}_{\bar{l}_{ij}}(s) |\tilde{\nu}_2^i(s) - \tilde{\nu}_2^j(s)| \\ & \qquad - \sum_{(j,i) \in \widehat{\mathbb{E}}_s^+} y_{l_{ij}}(s) |\tilde{\mu}_2^i(s) - \tilde{\mu}_2^j(s)| \\ \text{s.t.} \quad & \mathbf{1}^\mathsf{T} \bar{\boldsymbol{y}}(s) = \mathbf{1}^\mathsf{T} \boldsymbol{y}(s) = 1, \quad \lambda_2(\widehat{\boldsymbol{L}}(s)) > 0, \end{aligned} \tag{P2d}$$

provides a suboptimal solution to the link-rewiring problem (P2b). Here, $\tilde{\boldsymbol{\nu}}_2(s) = [\tilde{\nu}_2^1(s), \dots, \tilde{\nu}_2^n(s)]^\mathsf{T}$ and $\tilde{\boldsymbol{\mu}}_2(s) = [\tilde{\mu}_2^1(s), \dots, \tilde{\mu}_2^n(s)]^\mathsf{T}$ are the estimated Fiedler vector of graph $\mathcal{G}_{s-1}$ and $\mathcal{G}_s^+$ as a result of distributed algorithm (4), (7), *respectively.*

The pseudo-code for distributed single-link-rewiring is shown in Algorithm 2. In contrast with link-addition, one important issue for rewiring links is how to preserve the connectedness of the resulting network after removing the existing links. The lines 8-11 respond to the distributed verification of the connectedness of the graph $\mathcal{G}_s^+$ in the event of removing link (i^*, j^*). To this end, the distributed link-rewiring strategy with preserving-connectedness relocates sequentially the existing links by implementing m_r rounds of Algorithm 2.

Thus far, the developed strategy for link-addition (resp. rewiring) provides a sub-optimal solution to the optimization problem (1a) (resp. (2a)) as also shown by the next simulation section. The optimality gap between the solution to (P1d) and the global one to (P1b) involves two aspects: the approximation error of the matrix perturbation and the estimation accuracy of the distributed PI. According to matrix perturbation theory [27], the approximation error depends on all eigenvalues together with their associated eigenvectors of the Laplacian matrix. In particular, the graph with a large algebraic connectivity and a wide gap between $\lambda_3(\boldsymbol{L}) - \lambda_2(\boldsymbol{L})$ leads to a small error, as well as a tight upper bound on the accuracy of the distributed PI method in reference to (9).

Algorithm 2 Distributively rewiring one link in $\mathcal{G}_0$

Require: a connected graph $\mathcal{G}_{s-1}$, node i associates with a neighborhood set $\mathbb{N}^i_{s-1}$, a scalar $\epsilon \in [0,1)$, and a random non-zero vector $\boldsymbol{p} \in \mathbb{R}^n$

1: use Algorithm 1 to add one link and establish $\mathcal{G}_s^+$
2: execute (5) with $\boldsymbol{p}$ to produce an initial condition $\tilde{\boldsymbol{\mu}}_2(s,0)$
3: execute line 2-7 to distributively estimate Fiedler vector of the graph $\mathcal{G}_s^+$ by $\tilde{\boldsymbol{\mu}}_2(s,0)$
4: store $\mathbb{E}_{\text{temp}} \leftarrow \mathbb{E}_s^+$
5: node computes $l_{ij*}(s) = \mathsf{arg\,min}\, |\tilde{\mu}_i(s) - \tilde{\mu}_j(s)|$ over graph $(\mathbb{V}, \mathbb{E}_{\text{temp}})$ and transmits to neighbors
6: compute $l_{i^*j^*}(s) = \mathsf{arg\,min}\, |\tilde{\mu}_i(s) - \tilde{\mu}_j(s)|$ over graph $\mathcal{G}_s^+$ using min-consensus algorithm (2) with $x_i(0) \leftarrow |\tilde{\mu}_i(s) - \tilde{\mu}_{j^*}(s)|$
7: execute max-consensus algorithm (1) over graph $(\mathbb{V}, \mathbb{E}_s^+ \setminus \{(i^*, j^*)\})$ with $x_{i^*} = 1$ and $x_j = 0$ for all $j \neq i$
8: **if** $x_{i^*} \neq x_{j^*}$ **then**
9: $\quad \mathbb{E}_{\text{temp}} \leftarrow \mathbb{E}_{\text{temp}} \setminus \{(i^*, j^*)\})$
10: $\quad$ go back to step 5
11: **end if**
12: output $\widehat{\mathcal{G}}_s \leftarrow (\mathbb{V}, \mathbb{E}_s^+ \setminus \{(i^*, j^*))$

4. Numerical Simulations

In this section, the proposed distributed algorithms to yield sub-optimal solutions for link adding and rewiring problems are validated and evaluated via several numerical examples.

As the interconnection structure is shown in Fig. 1, a network of 10 nodes is taken into account in the first case. The small size of such a network makes possible the comparison with global optimal solution which is accessible via exhaustive search approach. We start off by demonstrating the results via a Monte-Carlo simulation of the distributed estimation algorithm (4) for the Fiedler vector with a sufficiently large iteration length. By observing Fig. 2(a),

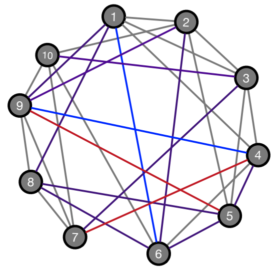
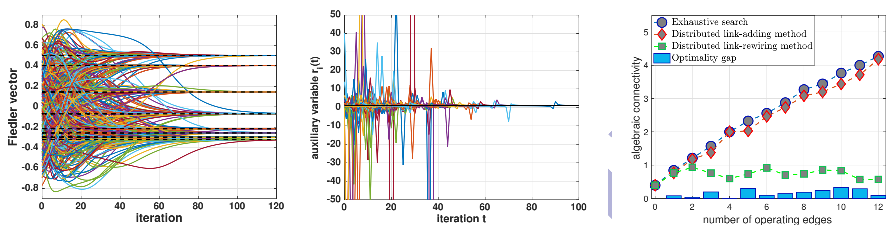


Figure 1: Adding 12 new links (colored lines) into a network consisting 10 nodes and 15 edges (gray lines).

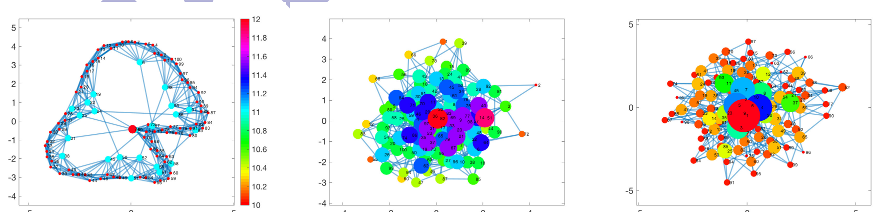


(a) distributed computation of $\boldsymbol{\nu}_2(\boldsymbol{L})$

(b) distributed computation of $\lambda_2(\boldsymbol{L})$

(c) a network example

Figure 2: Monte Carlo trial of distributed estimation algorithm and link-operation over a small-size network: (a) $\tilde{\nu}_2^i$ (solid colored lines) converge asymptotically to ν_2^i (dashed black line) for all $i \in \mathbb{V}$; (b) r_i (solid colored lines) converge asymptotically to $\lambda_2(\boldsymbol{L})$ (solid black line) for all $i \in \mathbb{V}$; (c) comparison of distributed link-addition strategy, the distributed link-rewiring strategy and brute-force search.



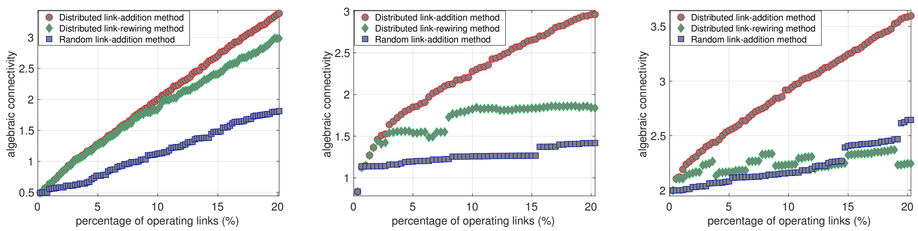
(a) Watts-Strogatz small world network

(b) Erdös Rényi stochastic network

(c) Barabási-Albert scale free network

Figure 3: Random network models of 100 nodes

the estimate states converge asymptotically to their corresponding true values independent of initial conditions. Fig. 2(b) provides the convergence behavior



(a) Watts-Strogatz small world network (b) Erdös Rényi stochastic network (c) Barabási-Albert scale free network

Figure 4: Performance comparison over random networks

of variables $r_i(t)$ ($i \in \mathbb{V}$), supporting the claim in Lemma 3. Namely, utilization of the auxiliary variable given in (6) allows us to distributively estimate the algebraic connectivity of a graph.

With the emphasis on finite-time computation in this article, we examine the effect of the distributed stopping criterion (7). The results are summarized in Table 2 in which a smaller threshold ϵ leads to a higher accuracy of approximations of the Fiedler vector but a longer iteration time. Since it is not necessary to compute the exact values of every entry of $\boldsymbol{\nu}_2$ in link operation problem, let the threshold ϵ be equal to 0.01 in the small-size case.

To evaluate the developed distributed strategy for link addition/rewiring, we vary the number m_a from 1 to 12. The resultant variation of algebraic connectivity after adding (rerouting) m_a (m_r) links into (of) the network is shown in Fig. 2(c) and those new links are drawn by blue and purple solid lines in Fig. 1. As shown in Fig. 2(c), the sub-optimal solution derived from the successive application of Algorithm 1 is very close to the global optimizer by a brute-force search. In addition, it should not be surprising that strengthening network resilience by adding new edges into the network outperforms by reallocating the existing links.

Next, the developed results are evaluated on three random networks: 1). *Watts-Strogatz* (WS) small-world model ($|\mathbb{V}| = 100|$, $|\mathbb{E}| = 308$), 2). *Erdös Rényi* (ER) random model ($|\mathbb{V}| = 100$, $|\mathbb{E}| = 268$), and 3). *Barabási-Albert*

Table 2: Stopping criterion with different thresholds

threshold ϵ	iteration steps	$\|\boldsymbol{\nu}_2 - \tilde{\boldsymbol{\nu}}_2(t)\|_2$	$\lambda_2(\boldsymbol{L}_1^+)$
0.085	1	1.4059	0.5101
0.08	25	0.2786	0.7935
0.05	28	0.1935	0.7935
0.01	40	0.0553	0.7935
0.005	47	0.0299	0.7935
0.001	66	0.0063	0.7935
0.0005	75	0.0031	0.7935
0.0001	95	0.0006	0.7935
0.00005	103	0.0003	0.7935
0.00001	123	0.0001	0.7935
0.000005	132	0.00003	0.7935

(BA) scale-free model ($|\mathbb{V}| = 100$, $|\mathbb{E}| = 281$). We first apply the proposed link addition strategy to construct $m_a = |\mathbb{E}_0|/10$ percent new interconnection of the amount of the existing links for each network and then employ the link-rewiring strategy to reconstruct $m_r = |\mathbb{E}^+_{m_a}|/10$ of the existing links. Here, we adopt a random link-adding method as an alternative solution of the global optimizer which is in general intractable in such large-scale networks. It is evident from Fig. 4 that the distributed link-addition strategy over all three random networks provide much more compelling performance than randomly adding links when it comes to network resilience improvement. Moreover, the distributed link-rewiring strategy seems to perform relatively worse than the distributed link-addition strategy in all three networks. As can be seen in Fig. 4(c), the distributed link-rewiring strategy loses its effect in BA networks and does not provide a better improvement of network resilience. This is mainly due to the higher heterogeneity of BA networks as compared to WS and ER ones. Here, heterogeneity of the network is concerned with its properties such as degree, betweenness, closeness, and centrality.

5. Conclusion

This article provides a distributed strategy to solve the problem of network topology manipulation when the global structure information is unavailable. Specifically, we consider the problem of link addition/rewiring in a network to enhance the network resilience against malicious attacks. Due to the combinatorial nature and NP-hardness, an approximation scheme based on eigenvalue sensitivity analysis is applied to these problems, providing a sub-optimal solution to primary optimization. The approximated problem involves the informative eigenvectors associated with the eigenvalues of interests. The development of the distributed algorithm to estimate the eigenvectors and the distributed stopping criteria to support finite-time computation facilitates us to fulfill link operation in a distributed manner without complete information of network topology. So far, the distributed-topology design problems are focused on an undirected and unweighted graph. Possible future direction would be the extension of current results to edge-consensus problems in line graphs [34], multiple time-scales multi-agent cooperation control [35, 36], and networks with time-varying [10] and antagonistic interactions [37].

Acknowledgements

This work was supported in part by the German Research Foundation within the Priority Program SPP 1914 “Cyber-Physical Networking”, and in part by the joint Sino-German Research Project “Control and optimization for event-triggered networked autonomous multi-agent systems (COVEMAS)”, which is funded through the German Research Foundation (DFG) and the National Science Foundation China (NSFC).

References

[1] H. Su, H. Wu, X. Chen, M. Chen, Positive edge consensus of complex networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (2017) 2242–2250.

[2] D. Wang, J. Zhou, Z. Wang, W. Wang, Random gradient-free optimization for multiagent systems with communication noise under a time-varying weight balanced digraph, IEEE Transactions on Systems, Man, and Cybernetics: Systems-Doi: 10.1109/TSMC.2017.2757265.

[3] T. Yang, Y. Wan, H. Wang, Z. Lin, Global optimal consensus for discrete-time multi-agent systems with bounded controls, Automatica 97 (2018) 182–185.

[4] K. Zhao, A. Kummar, T. Harrison, J. Yen, Analyzing the resilience of complex supply network topologies against random and targeted disruptions, IEEE Systems Journal 5 (1) (2011) 28–39.

[5] P. Smith, D. Hutchison, J. Sterbenz, M. Schoeller, A. Fessi, M. Karaliopoulos, C. Lac, B. Plattner, Network resilience: A systematic approach, IEEE Communications Magazine 49 (7) (2011) 88–97.

[6] C. Schneider, A. Moreira, J. J. Andrade, S. Havlin, H. Herrmann, Mitigation of malicious attacks on networks, PNAS 108 (10) (2011) 3838–3841.

[7] L. Gunderson, Ecological resilience - In theory and application, Aunnal Review of Ecology and Systematics 31 (2000) 425–439.

[8] J. Simpson-Porco, F. Dörfler, F. Bullo, Voltage collapse in complex power grids, Nature Communications 7 (2016) 1–8.

[9] M. Asllani, T. Carletti, Topological resilience in non-normal networked systems, Physical Review E 97.

[10] T. Yang, Z. Meng, G. Shi, Y. Hong, K. Johansson, Network synchronization with nonlinear dynamics and switching interactions, IEEE Transactions on Automatic Control 61 (10) (2016) 3103–3108.

[11] H. Wang, P. V. Mieghem, Algebraic connectivity optimization via link addition, Proceeding of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems.

[12] M. Jalili, X.-H. Yu, Enhancement of synchronizability in networks with community structure through adding efficient inter-community link, IEEE Transactions on Network Science and Engineering 3 (2) (2016) 106–116.

[13] V. Blondel, J. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica 36 (2000) 1249–1274.

[14] A. Bishop, I. Shames, Link operations for slowing the spread of disease in complex networks, Europhysics Letters 95 (18005).

[15] P. van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. van de Bovenkamp, D. Liu, H. Wang, Decreasing the spectral radius of a graph by link removals, Physical Review E 84 (016101).

[16] C. Schneider, T. Mihaljev, S. Havlin, H. Herrmann, Suppressing epidemics with a limited amount of immunization units, Physical Review E 84.

[17] A. Milanese, J. Sun, T. Nishikawa, Approximating spectral impact of structural perturbation in large networks, Physical Review E 81 (2010) 046112.

[18] D. Kempe, F. McSherry, A decentralized algorithm for spectral analysis, Journal of Computer and System Sciences 74(1) (2008) 70–83.

[19] P. Yang, R. Freeman, G. Gordon, K. Lynch, S. Srinivasa, R. Sukthankar, Decentralized estimation and control of graph connectivity for mobile sensor networks, Automatica 46(2) (2010) 390–396.

[20] D. Xue, A. Gusrialdi, S. Hirche, Robust distributed control design for interconnected systems under topology uncertainty, In: Proceeding of the 2013 American Control Conference (2013) 6556–6561.

[21] S. Liu, P. Chen, A. H. III, Acceleterat distributed dual averaging over evolving networks of growing connectivity, IEEE Transactions on Signal Processing 66 (7) (2017) 1845–1859.

[22] A. Bertrand, M. Moonen, Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks, Signal Processing 93 (5) (2013) 1106–1117.

[23] H. Liu, X. Cao, J. He, P. Cheng, C. Li, J. Chen, Y. Sun, Distributed identification of the most critical node for average consensus, IEEE Transactions on Signal Processing 63 (16) (2015) 4315–4328.

[24] D. Xue, A. Gusrialdi, S. Hirche, A distributed strategy for near-optimal network topology design, In: Proceeding of the 21st International Symposium on Mathematical Theory of Networks and Systems (2014) 7–14.

[25] B. Oreshkin, M. Coates, M. Rabbat, Optimization and analysis of distributed averaging with short node memory, IEEE Transactions on Signal Process 58 (5) (2014) 2850–2865.

[26] T. Summers, F. Cortesi, J. Lygeros, On submodularity and controllability in complex dynamical networks, IEEE Transactions on Control of Network Systems 3 (2016) 91–101.

[27] G. Stewart, J. Sun, Matrix perturbation theory, Boston: Academic Press, 1990.

[28] W. Asif, M. Lestas, H. Qureshi, M. Rajarajan, Optimization based spectral partitioning for node criticality assessment, Journal of Network and Computer Applications 75 (2016) 279–292.

[29] G. Golub, C. van Loan, Matrix computations, 4th Edition, Johns Hopkins University Press, 2012.

[30] S. Borm, C. Mehl, Numerical methods for eigenvalue problems, De Gruyter Textbook, 2012.

[31] S. Zhang, C. Tepedelenlioglu, A. Spanias, Distributed network center and size estimation, IEEE Sensors Journal 14 (15) (2018) 6033–6045.

[32] H.-S. Su, X.-F. Wan, G.-R. Chen, A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements, International Journal of Control 82 (2009) 1334–1343.

[33] D. Xue, S. Hirche, Distributed topology manipulation to control epidemic spreading over networks, IEEE Transactions on Signal ProcessingDoi: 10.1109/TSP.2018.2887211.

[34] H. Su, H. Wu, J. Lam, Positive edge-consensus for nodal networks via output feedback, IEEE Transactions on Automatic ControlDoi: 10.1109/TAC.2018.2845694. doi:DOI:10.1109/TAC.2018.2845694.

[35] M. Long, H. Su, B. Liu, Group controllability of two-time-scale multi-agent networks, Journal of Franklin Institute 355 (2018) 6045–6061.

[36] M. Long, H. Su, B. Liu, Second-order controllability of two-time-scale multi-agent systems, Applied Mathematics and Computation 343 (2019) 299–313.

[37] F. Liu, D. Xue, S. Hirche, M. Buss, Polarizability, consensusability and neutralizability of opinion dynamics on coopetitive networks, IEEE Transactions on Automatic ControlDoi: 10.1109/TAC.2018.2879599. doi:doi:10.1109/TAC.2018.2879599.

[38] A. Ghosh, S. Boyd, Growing well-connected graphs, In: Proceeding of the 45th IEEE Conference on Decision and Control (2006) 6605–6611.

[39] E. Ramirez-Llanos, S. Martinez, Distributed stopping criterion for the power iteration applied to virus mitigation, In: Proceedings of 49th Asilomar Conference on Signals, Systems and Computers (2015) 1328–1332.

6. Appendices

6.1. Proof of Lemma 1

It is already known that the algebraic connectivity of a connected graph is a non-decreasing function with respect to the edge addition [38]. According to matrix perturbation theory, the first order expansion of the second smallest eigenvalue of $\boldsymbol{L}_0 + \Delta \boldsymbol{L}_{m_a}^{+}$ can be computed as

$$\lambda_2(\boldsymbol{L}_0 + \Delta \boldsymbol{L}_{m_a}^{+}) = \lambda_2(\boldsymbol{L}_0) + \frac{\boldsymbol{\nu}_2^{\intercal} \Delta \boldsymbol{L}_{m_a}^{+} \boldsymbol{\nu}_2}{\boldsymbol{\nu}_2^{\intercal} \boldsymbol{\nu}_2} + \mathcal{O}(||\Delta \boldsymbol{L}_{m_a}^{+}||^2),$$

which enables us to approximate the objective function in (P1b) by

$$\max_{\tilde{\boldsymbol{y}}} \lambda_2(\boldsymbol{L}^+_{m_a}) \approx \lambda_2(\boldsymbol{L}_0) + \frac{\max_{\tilde{\boldsymbol{y}}} \boldsymbol{\nu}_2^\top \Delta \boldsymbol{L}^+_{m_a} \boldsymbol{\nu}_2}{\|\boldsymbol{\nu}_2\|_2^2}.$$

Upon the factorization of $\Delta \boldsymbol{L}^+_{m_a}$, we arrive at the approximate algorithm (P1c) and the proof is completed.

6.2. Proof of Proposition 1

In order to prove the statement in Proposition 1, the following supporting lemma is necessary.

Lemma 4. *Consider two positive semi-definite and symmetric matrices* $\boldsymbol{C}_1, \boldsymbol{C}_2 \in \mathbb{R}^{n\times n}$*. Let* $\boldsymbol{\zeta}_n$ *and* $\boldsymbol{\xi}_n$ *be the eigenvectors associated to the dominant eigenvalues* $\lambda_n(\boldsymbol{C}_1)$ *and* $\lambda_n(\boldsymbol{C}_2)$*, respectively. Then, one has*

$$|\lambda_n(\boldsymbol{C}_1) - \lambda_n(\boldsymbol{C}_2)| \leq \|\boldsymbol{C}_1 - \boldsymbol{C}_2\|_2,$$
$$\|\boldsymbol{\zeta}_n - \boldsymbol{\xi}_n\|_2 \leq \frac{2\|\boldsymbol{C}_1 - \boldsymbol{C}_2\|_2}{\lambda_n(\boldsymbol{C}_1) - \lambda_{n-1}(\boldsymbol{C}_1)} \|\boldsymbol{\xi}_n\|_2.$$

Proof. The proof follows similar steps as those in the proof of Lemma 1 and Lemma 2 in [39]. □

The proof of Proposition 1 follows along the same steps as in the proof of Theorem 1 in [39] and thus is saved here.

6.3. Proof of Lemma 3

Consider a Lyapunov function candidate by the span norm

$$V(\tilde{\boldsymbol{\nu}}_2) = \max_{i\in\mathbb{V}} r_i(\tilde{\boldsymbol{\nu}}_2) - \min_{i\in\mathbb{V}} r_i(\tilde{\boldsymbol{\nu}}_2),$$

which is continuous for all $\tilde{\nu}_2 \in \mathbb{R} \setminus \{0\}$ and positive semi-definite.

The following computation,

$$\begin{aligned}
&\max_{i\in\mathbb{V}} r_i\left(\tilde{\boldsymbol{\nu}}_2(t+1)\right) - \max_{i\in\mathbb{V}} r_i\left(\tilde{\boldsymbol{\nu}}_2(t)\right) \\
=& \max_{i\in\mathbb{V}} \frac{\alpha(t)}{r_i(t)\tilde{\nu}_2^i(t)} \sum_{j=1}^{n} [\boldsymbol{Q}]_{ij} \frac{r_j(t)\tilde{\nu}_2^j(t)}{\alpha(t)} - \max_{i\in\mathbb{V}} r_i(t) \\
\leq& \max_{i\in\mathbb{V}} \frac{1}{r_i(t)\tilde{\nu}_2^i(t)} \sum_{j=1}^{n} [\boldsymbol{Q}]_{ij} \tilde{\nu}_2^j(t) \max_{k\in\mathbb{V}} r_k(t) - \max_{i\in\mathbb{V}} r_i(t) = 0
\end{aligned}$$

implies $\mathsf{max}_{i \in \mathbb{V}} r_i$ is non-decreasing, whereby verifying $\mathsf{max}_{i \in \mathbb{V}} r_i$ is non-increasing on the same principle. Thus, it then suffices to address that the max-min Lyapunov function V is reduced to a constant factor over a sufficiently large time interval. In particular, it follows that $V(t+1) - V(t) = 0$ if and only if $\tilde{\boldsymbol{\nu}}_2 \in \{\tilde{\boldsymbol{\nu}}_2 | r_1(\tilde{\boldsymbol{\nu}}_2) = \cdots = r_n(\tilde{\boldsymbol{\nu}}_2)\}$. With the consensus configuration at the equilibrium point and by the fact that

$$\lim_{t \to \infty} \frac{\tilde{\boldsymbol{\nu}}_2^\top(t) \, \mathsf{diag}(r_1(t), \ldots, r_n(t)) \tilde{\boldsymbol{\nu}}_2(t)}{\tilde{\boldsymbol{\nu}}_2^\top(t) \tilde{\boldsymbol{\nu}}_2(t)} = \lim_{t \to \infty} \frac{\tilde{\boldsymbol{\nu}}_2^\top(t) \boldsymbol{Q} \tilde{\boldsymbol{\nu}}_2(t)}{\tilde{\boldsymbol{\nu}}_2^\top(t) \tilde{\boldsymbol{\nu}}_2(t)} = \lambda_n(\boldsymbol{Q}),$$

one can infer $\lim_{t \to \infty} r_i(t) = 1 - \beta \lambda_2(\boldsymbol{L})$ for all $i \in \mathbb{V}$.

6.4. Proof of Theorem 1

From the formulation in (6), we note the fact that $|r_i||\tilde{\nu}_2^i|$ amounts to the i-th entry of the matrix-vector product $|\boldsymbol{Q}\tilde{\boldsymbol{\nu}}_2|$ for all $i \in \mathbb{V}$. After carrying out the matrix (vector) multiplication, this fact allows us to expose the relation that

$$\frac{\tilde{\boldsymbol{\nu}}_2^\top(t) \, \mathsf{diag}(|r_1(t)|, \ldots, |r_n(t)|) \tilde{\boldsymbol{\nu}}_2(t)}{\tilde{\boldsymbol{\nu}}_2^\top(t) \tilde{\boldsymbol{\nu}}_2(t)} = \left| \frac{\tilde{\boldsymbol{\nu}}_2^\top(t) \boldsymbol{Q} \tilde{\boldsymbol{\nu}}_2(t)}{\tilde{\boldsymbol{\nu}}_2^\top(t) \tilde{\boldsymbol{\nu}}_2(t)} \right|,$$

which implies $|\tilde{\lambda}_2(t)| \geq \mathsf{min}_{i \in \mathbb{V}} |r_i(t)|$.

Moreover, the residual $\boldsymbol{\gamma}$ of $\tilde{\boldsymbol{\nu}}_2$ with respect to $\boldsymbol{Q}$ can be equivalently stated as the inverse vector of the *rejection* of $\boldsymbol{Q}\tilde{\boldsymbol{\nu}}_2$ from $\tilde{\boldsymbol{\nu}}_2$. Therefore, $\|\boldsymbol{\gamma}(t)\|_2$ is equal to the shortest distance from $\boldsymbol{Q}\tilde{\boldsymbol{\nu}}_2$ to the line spanned by $\tilde{\boldsymbol{\nu}}_2$. The denotation of $\boldsymbol{\rho}$ in (8) can be reformulated by $\boldsymbol{\rho} = (\mathsf{max}_{i \in \mathbb{V}} |r_i|) \tilde{\boldsymbol{\nu}}_2 - \boldsymbol{Q}\tilde{\boldsymbol{\nu}}_2$, leading to that

$$\|\boldsymbol{\gamma}\|_2 \leq \|\boldsymbol{\rho}\|_2 \leq \sqrt{\tilde{n}} \max_{i \in \mathbb{V}} |\rho_i|,$$

where the relation $\|\boldsymbol{\rho}\|_2 \leq \sqrt{\tilde{n}} \|\boldsymbol{\rho}\|_\infty$ is used.

To this end, one can immediately conclude that

$$\|\boldsymbol{\gamma}\|_2 \leq \sqrt{\tilde{n}} (\max_{i \in \mathbb{V}} |\rho_i|) \leq \epsilon (\min_{i \in \mathbb{V}} |r_i(t)|) \leq \epsilon |\tilde{\lambda}_2(t)|,$$

which provides a bound on the accuracy of the approximation of eigenvector as a result of the centralized stopping criterion given in Proposition 1.