Distributed Link Removal Using Local Estimation of Network Topology

Azwirman Gusrialdi, *Member, IEEE*, Zhihua Qu, *Fellow, IEEE*, and Sandra Hirche, *Senior Member, IEEE*

Abstract—This paper considers the problem of network structure manipulation in the absence of information on the global network topology. In particular, the problem of removing some of links is investigated in order to slow or stop the spread of disease in a network while preserving its connectivity. Existing methods solve this combinatorial problem in a centralized manner and they require the global information of network structure. In this paper, we propose a distributed design algorithm to compute a suboptimal solution to this problem efficiently by mimicking gradient based method, namely by iteratively removing one or multiple links at a time from the network. Specifically, using matrix perturbation analysis we formulate an optimization problem involving the eigenvector associated with the largest eigenvalue of the adjacency matrix and whose solution is equal to a suboptimal solution to the original problem. This strategy also enables us to overcome the combinatorial issue of the problem. Distributed algorithms to estimate the eigenvector and to verify network's connectivity are then proposed which facilitate us to solve the new optimization problem. In addition, topological insights into the proposed algorithm and optimality of its solution are also discussed. Finally, the proposed distributed design method is demonstrated and evaluated via several numerical examples.

Index Terms—Link removal, largest eigenvalue minimization, distributed algorithm, matrix perturbation.

1 Introduction

Complex networks as an interaction between individual entities can be found in many real world examples, not only in critical infrastructures such as telecommunication networks, power grids, transportation networks, water distribution networks but also in social networks and biological networks. One important feature in complex networks is its topology which captures the interaction between subsystems/components in a network via sensing, control, communication or physical interconnection. The research on network topology manipulation, including link addition/removal/rewiring, has attracted much interest in recent years, see e.g., [1], [2], [3], [4], [5], [6]. In this paper, we consider the problem of removing a fraction of links from a network such that the largest eigenvalue of the adjacency matrix is minimized while guaranteeing the network connectivity.

As a motivating application, it is known that the disease infecting node in a network can spread through the whole network and may cause outbreak whose speed is related to the network topology and characterized by the largest eigenvalue of its adjacency matrix [7]. Note that the term *disease* is quite general and is applicable to various applications such that the spread of viruses in networked computer system, cascading failures in networked power grid [8] or attacks in cyber-physical systems. One way to slow the spread of disease or contain its dissemination, other than by immunizing certain individual nodes [9], is by removing a fraction of existing links from the network until the largest eigenvalue of its adjacency matrix is below a certain threshold. This means that we do not need to immunize the individual nodes but rather control their interactions and the disease's path of spreading [3], [10]. Note that removing links in a network are equivalent to "unfriending" people in social network such as Facebook or disconnecting the physical lines via switches in power network.

In general the problem of link removals of complex network to minimize the largest eigenvalue of the adjacency matrix is NP-hard [1]. Therefore, most of the work focuses on heuristics and approximations of this problem. Although the authors in [3] propose a relaxation method based on semi-definite programming (SDP) to solve this problem, the approach does not scale to large-size networks and is not applicable for weighted networks. The works [1], [10] propose various heuristics such as the one based on eigenvalue sensitivity and number of hop walks of the graph to solve approximately the original problem. However, the authors do not consider connectivity preservation of the network. It is worth noting that minimization of the largest eigenvalue of a symmetric matrix has received a lot of attention since the seventies and several algorithms have been proposed, e.g. [11], [12]. Nonetheless, in this existing work the decision variable is continuous while, in this paper, the decision variable is discrete. Note that extending such a result from continuous variables to discrete ones is far from trivial, because mixed integer optimization problems are often too challenging and could only be solved in special cases. For example, even though it may be possible to apply a relaxation method in combination with a threshold operation to decide which links to be removed, how to optimally choose the threshold is not a trivial problem, especially when taking into account the constraint on the network connectivity. More importantly, all the work mentioned above assumes that the global network structure is known and available to the designer. However, in practice the global network structure information may not be available or may be difficult to accumulate in a centralized fashion because of geographical constraints such as in power grid [13]. Another reason is due to privacy concerns [14], [15], namely each individual node may not be willing to share its set of neighbors to the individuals who are not his/her friend as motivated from social network. This limitation

• *A. Gusrialdi and Z. Qu are with Dept. of Electrical and Computer Engineering, UCF, Orlando 32816, USA. S. Hirche is with the Institute for Information-Oriented Control, Department of Electrical Engineering and Information Technology, Technische Universität München, D-80290, Germany. Emails:{azwirman.gusrialdi, qu}@ucf.edu, hirche@tum.de.*

on the network structure information prevents applicability of the existing (centralized) approaches. On the other hand, a node in real world normally knows to whom it is connected in the network. Hence, a natural way to solve the link removal problem in the absence of global network structure information is by solving it distributively, namely based on the local information available to each node. Moreover, distributed algorithm also allows each node to make its own decision in responding to the changes in the network without the need of a centralized decision maker. Distributed algorithm for link removals is proposed in [16] where it is assumed that the healthy node can assess the state of its neighbors and based on this information, it will remove a link to a neighboring infected node with a certain probability. However, for certain scenarios it may not always be possible to assess the current state of other nodes in a network. In addition, the results in [16] is tailored to SIS epidemic model and the links being removed may destroy the connectivity of the network. The probability of removing a link is also chosen heuristically and there is no optimality discussion on solution obtained from the proposed algorithm. Recently, there has been a growing interest in reconstructing network topology. One of the most recent results which reconstructs the topology in a fully distributed fashion is presented in [17]. One might then ask whether it is possible to apply such distributed network reconstruction strategy to determine global network structure. We should mention that distributed reconstruction strategy proposed in [17] has several limitations such as it is only suitable for small size network and is limited to unweighted graph. More importantly, the algorithms require cooperation between the nodes, which may not be possible when the nodes do not want to reveal their neighboring information to others due to privacy reasons and thus decline to cooperate.

In this paper, we consider link removal problem for an undirected network in the absence of global network structure in order to control the spread of disease while guaranteeing the network connectivity. The problem is motivated by numerous applications such as: (i) diminishing the propagation of frequency oscillation or cascading failure in power grid while guaranteeing that each load can still be served by the generators; (ii) reducing distribution of malware in social network while ensuring important information can still be received by each person; (iii) removing virus from a computer network while guaranteeing e-mail or message can still be sent from/to any member in the network. The problem is formulated as to minimize the largest eigenvalue of adjacency matrix by removing a given number of links from the existing network. Since this problem is combinatorial in nature, we aim to compute one of its suboptimal solutions under the unavailability of global network topology information. Specifically, the contributions of this paper are fourfold.

(i) We present a unified distributed strategy to compute a suboptimal solution to link removal with connectivity preservation problem. The main idea is to mimic gradient based approach by removing the links iteratively. Specifically, using matrix perturbation analysis we formulate an optimization problem involving the eigenvector associated with the largest eigenvalue and whose solution is equal to a suboptimal solution to the original problem. We provide an explicit rule on choosing the stopping threshold for distributed power iteration to estimate the eigenvector so that the stability of the overall system due to the estimation error is guaranteed. Moreover, we present a simple distributed algorithm based on max-consensus protocol to verify the connectivity of a graph as a result of link removal. It is worth noting that distributed strategy proposed in this paper also has broad applications such as for computing a suboptimal solution to link addition problem whose goal is either to improve the network connectivity or maximize the spread of information in a network. Even though perturbation analysis has been used in topology design problem, it is still not clear how to implement it distributively while also taking into account network connectivity.

(ii) We provide analysis on optimality of the proposed approach in comparison to brute force search in the presence of global network topology information. Specifically, we present the gap between global optimal solution and solution to the distributed strategy for network with a large spectral gap.

(iii) We investigate topological properties on solution to the proposed distributed algorithm. It is shown from both analysis on complete graphs and simulations on random networks that the proposed strategy tends to balance degree distribution by reducing the gap between maximum and minimum degrees of the network. This topological insight may be used to reduce search space when solving the combinatorial problem in the presence of global network topology information.

(iv) We compare performance of the proposed distributed design method, the brute-force search and random removal strategy for different number of link removals via simulations on several random networks.

The organization of this paper is as follows: basic notions from graph theory and well-known results which will be used to develop the algorithm are presented in Section 2. After formally stating the problem in Section 3, the proposed connectivity-preserving distributed algorithms to remove some links from a network are described in Section 4. Moreover, its optimality and topological insights are discussed in Section 5. Finally, the proposed strategy is demonstrated and its performance is evaluated via several numerical examples in Section 6.

2 Notation and Preliminaries

In this section, we recall some basic notions from graph theory and well-known results which will be used for the development of distributed link removal strategy.

2.1 Notation

Let $\mathbb{R}$ be the set of real numbers and vector $1_n \in \mathbb{R}^n$ denote the column vector of all ones. Furthermore, $\mathrm{diag}(a) \in \mathbb{R}^{n\times n}$ represents the diagonal matrix with the elements of vector $a \in \mathbb{R}^n$ on its diagonal. For a given set $\mathscr{N}$, $|\mathscr{N}|$ denotes the number of the elements in this set. For a symmetric matrix C, let $\lambda_i(C)$ denote its eigenvalues which without loss of any generality is ordered as

$$\lambda_1(C) \leq \lambda_2(C) \leq \ldots \leq \lambda_n(C) = \lambda_{max}(C)$$

and its spectral radius is defined as $\rho(C) \triangleq \max_i(|\lambda_i(C)|)$.

Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be an *undirected* graph with a set of vertices $\mathscr{V} = \{1, 2, \ldots, n\}$ and a set of edges (links) $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$. Hence, if $(i, j) \in \mathscr{E}$, then $(j, i) \in \mathscr{E}$ and nodes i and j are neighbors. The neighborhood set of node i is given by $\mathscr{N}_i \subseteq \mathscr{V}$. The *degree* matrix of a graph, $\mathscr{D}(\mathscr{G}) \in \mathbb{R}^{n\times n}$, is given by $\mathscr{D}(\mathscr{G}) \triangleq \mathrm{diag}(d)$,

where $d = [d_1, \ldots, d_n]^T$ and $d_i = |\mathcal{N}_i|$. The *adjacency* matrix of a graph, $\mathscr{A}(\mathscr{G}) \in \mathbb{R}^{n\times n}$, is defined as

$$[\mathscr{A}(\mathscr{G})]_{ij} \triangleq \begin{cases} 1, & \text{if } (i,j) \in \mathscr{E}, \\ 0, & \text{otherwise.} \end{cases}$$

The *Laplacian* matrix of graph $\mathscr{G}$, $\mathscr{L}(\mathscr{G}) \in \mathbb{R}^{n\times n}$ is then given by

$$\mathscr{L}(\mathscr{G}) \triangleq \mathscr{D}(\mathscr{G}) - \mathscr{A}(\mathscr{G}). \tag{1}$$

A graph is *connected* if there is a path between any pair of distinct nodes. When a graph $\mathscr{G}$ is connected, the eigenvalues of its Laplacian satisfy $\lambda_1(\mathscr{L}(\mathscr{G})) = 0$ and $\lambda_2(\mathscr{L}(\mathscr{G})) > 0$. For graph $\mathscr{G}$, we denote $l_{\mathscr{G}} \in \{1, \ldots, |\mathscr{E}|\}$ the index of the edges in $\mathscr{G}$.

Matrix C is nonnegative if all its elements are nonnegative. Nonnegative symmetric matrix C is irreducible if and only if its associated graph $\mathscr{G}$ is connected. Nonnegative matrix C is said to be primitive if there exists a positive integer k such that $C^k > 0$. A sufficient condition for matrix C to be primitive is that it is irreducible and has at least one positive diagonal element [18]. For a primitive matrix C, its largest eigenvalue $\lambda_{max}(C)$ satisfies $\lambda_{max}(C) > |\lambda_i(C)|$ for $i \neq n$ and thus we have $\rho(C) = \lambda_{max}(C)$ [19].

2.2 Max-Consensus Algorithm

Consider an undirected graph $\mathscr{G}$ with n nodes and let us assign state $x_i(t) \in \mathbb{R}$ to each node of $\mathscr{G}$. The max-consensus algorithm allows all nodes to compute distributively the maximum among the initial conditions $x_i(0)$ for $i = \{1, \cdots, n\}$. Specifically, each node executes the following update rule [20]

$$x_i(t+1) = \max_{j\in\mathcal{N}_i\cup\{i\}} x_j(t). \tag{2}$$

Algorithm (2) can also be modified in a straightforward manner to compute the minimum among the initial conditions $x_i(0)$, i.e., min-consensus protocol. It is known that update rule (2) makes all states $x_i(t)$ converge to $\max_i x_i(0)$ in no more than n steps, i.e., its computational complexity is equal to $O(n)$. It will be demonstrated throughout the paper that max-consensus algorithm serves as a unified framework to solve our problem.

2.3 Matrix Perturbation

In this paper we will utilize the following result on matrix perturbation [21, p.183].

Lemma 1. *Consider a symmetric matrix A. Let λ be a simple eigenvalue of A with eigenvector equal to v and let $\overline{A} = A + \Delta A$ be a perturbation of A. Then, there is a unique eigenvalue $\overline{\lambda}$ of $\overline{A}$ such that*

$$\overline{\lambda} = \lambda + \frac{v^T \Delta A v}{v^T v} + \mathrm{O}\big(\|\Delta A\|^2\big). \tag{3}$$

2.4 Distributed Power Iteration Method

Consider a primitive symmetric matrix $P \in \mathbb{R}^{n\times n}$. Its eigenvector $v_n = [v_{n,1}, \cdots, v_{n,n}]^T$ corresponding to the simple largest eigenvalue $\lambda_{max}(P)$ can be estimated using power method by performing the following iterations [22]

$$\hat{v}_n(t+1) = \frac{P\hat{v}_n(t)}{\|P\hat{v}_n(t)\|_\infty} \tag{4}$$

where $\hat{v}_n(t)$ is the estimate of v_n at the t-th iteration. Since $v_n > 0$, any choice of initial condition $\hat{v}_n(0) > 0$ satisfies $\hat{v}_n(0) \cdot v_n \neq 0$ guarantees that $\hat{v}_n$ will asymptotically converge to v_n with rate of convergence equal to $|\lambda^*(P)/\lambda_{max}(P)|^2$ where λ^* denotes the second largest eigenvalue of P in magnitude. Since the graph $\mathscr{G}$ associated with P is connected, the norm $\|P\hat{v}_n(t)\|_\infty$ can be computed distributively via max-consensus algorithm (2) by setting $x_i(0) = \sum_{j\in\mathcal{N}_i\cup\{i\}} [P]_{ij}\hat{v}_{n,j}(t)$. Hence the iteration (4) can be performed in a distributed fashion. In this paper, we adopt the following distributed stopping condition for iteration (4) [23]

$$\sqrt{\overline{n}}\|r(t)\|_\infty \leq \overline{\varepsilon}\left(\min_{i\in\mathcal{V}} \bar{h}_i(t)\right), \tag{5}$$

for a given threshold $\overline{\varepsilon} \in [0,1)$ where

$$r(t) \triangleq \left(\max_{i\in\mathcal{V}} \bar{h}_i(t)\right)\hat{v}_n(t) - P\hat{v}_n(t),$$
$$\bar{h}_i(t) \triangleq \frac{1}{\hat{v}_{n,i}(t)} \sum_{j\in\mathcal{N}_i\cup i} [P]_{ij}\hat{v}_{n,j}(t).$$

Similarly, condition (5) can also be checked via max/min-consensus protocol (2). At the end of distributed estimation process, node i will have the estimate of $v_{n,i}$. At each iteration t, each node requires at least $4\overline{n}$ steps to compute $\hat{v}_{n,i}$ since $\overline{n}$ steps are required to compute $\|P\hat{v}_n(t)\|_\infty$ and a total of $3\overline{n}$ steps are necessary to check stopping condition (5). Given threshold $\overline{\varepsilon}$, the estimation error of v_n is bounded by [23]

$$\|v_n - \hat{v}_n(t)\|_\infty \leq 2\overline{\varepsilon}\left(1 + \frac{\overline{\varepsilon}}{1-\overline{\varepsilon}}\right)\frac{\lambda_n(P)}{\lambda_n(P) - \lambda_{n-1}(P)}. \tag{6}$$

Remark 1. *In general the (distributed) power iteration method, e.g. [24], [25], [26] converges asymptotically. The work [27] presents distributed algorithm to estimate in finite-time the eigenvector corresponding to the spectral radius of a matrix. However, the result is only limited to matrix whose spectral radius is equal to one and thus is not applicable to our problem.*

Remark 2. *The reason we perform normalization in* (4) *is to overcome the overflow problem as the iteration numbers increase. Specifically, the nonzero components in the iteration vector may become extremely large values if $|\lambda_{max}| > 1$ or can approach zero if $|\lambda_{max}| < 1$. Since normalization has no effect on the convergence property of the power method [28, sec. 7.4.1], it can then be done intermittently.*

3 Problem Formulation

Consider an n node network whose connections can be represented by an (unweighted) undirected graph $\mathscr{G}_0 = \{\mathcal{V}, \mathscr{E}_0\}$. In addition, let us assume that the graph $\mathscr{G}_0$ is connected and our goal is to remove at most m_e number of links $\Delta\mathscr{E}^-$ from $\mathscr{E}_0$ such that the largest eigenvalue of adjacency matrix of the resulting graph $\bar{\mathscr{G}}_{m_e} = \{\mathcal{V}, \mathscr{E}_0 \setminus \Delta\mathscr{E}^-\}$ is minimized. As a motivating application, it is well-known that the largest eigenvalue λ_{max} is the key to spreading of disease under various types of epidemic models [7], [29], [30]. Specifically, the epidemics die out if $\lambda_{max} \leq \beta_c$ where β_c is a threshold that depends on the epidemic models. Hence, one method to control the epidemic is by removing some links from a network to reduce its largest eigenvalue. The problem in general can then be mathematically formulated as the following optimization problem:

$$\begin{aligned} \min_{\Delta\mathscr{E}^-} \quad & \lambda_{max}(\mathscr{A}(\bar{\mathscr{G}}_{m_e})), \\ \text{s.t.} \quad & |\Delta\mathscr{E}^-| \leq m_e, \\ & \Delta\mathscr{E}^- \subseteq \mathscr{E}_0, \\ & \lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e})) > 0, \end{aligned} \tag{P1a}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2813426, IEEE Transactions on Network Science and Engineering

where the last constraint implies that the resulting topology is connected. Using the labeling $l_{\mathscr{G}_0} \in \{1,\dots,|\mathscr{E}_0|\}$ on graph $\mathscr{G}_0$ introduced in the previous subsection, the optimization problem (P1a) can then be restated as:

$$\begin{aligned} \min_{\bar{y}} \quad & \lambda_{max}(\mathscr{A}(\mathscr{G}_0) - \Delta A^-), \\ \text{s.t.} \quad & \Delta A^- = \sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} A_{l_{\mathscr{G}_0}}, \\ & \lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e})) > 0, \\ & 1_{|\mathscr{E}_0|}^T \bar{y} \le m_e, \\ & \bar{y} \in \{0,1\}^{|\mathscr{E}_0|}, \end{aligned} \tag{P1b}$$

where $A_{l_{\mathscr{G}_0}}$ is a matrix with all zeros entries except for the ij-th and ji-th entries corresponding to the edge of label $l_{\mathscr{G}_0}$ which are equal to 1, and the vector $\bar{y} = [\bar{y}_1,\dots,\bar{y}_{|\mathscr{E}_0|}]^T$ is a Boolean vector, i.e., $\bar{y}_{l_{\mathscr{G}_0}} \in \{0,1\}$ where $\bar{y}_{l_{\mathscr{G}_0}} = 1$ means that the edge $l_{\mathscr{G}_0}$ in $\mathscr{E}_0$ is removed.

Remark 3. *The formulation in (P1b) can also be modified to take into account the degradation of the convergence speed to consensus as will be shown later in this paper. Note that the optimization (P1b) can be solved before the network is employed or after the employment since its topology may change.*

The optimization problem (P1b) is a combinatorial one whose global solution can be computed using brute-force search. However, the complexity increases exponentially with the network size. In addition, one common issue in solving (P1b), including its relaxation, for example in [3], is that the global knowledge on network topology $\mathscr{G}_0$ is required or assumed to be known. However, in reality such information is often not available due to, e.g., privacy reasons and geographical constraint or hard to obtain due to the size of the network. Hence, in this paper we impose the following constraint.

Constraint 1. *The overall network topology $\mathscr{G}_0$ is not available and each node i only knows the information on $\mathscr{N}_i$.*

The absence of information on global network topology makes it impossible to solve (P1b) in a centralized manner. Therefore, in this paper we propose a distributed strategy performed by each node to solve (P1b) using only its local information on the network structure given by $\mathscr{N}_i$.

Problem 1. *Assume that $\mathscr{G}_0$ is connected. Find a suboptimal solution or an upper bound to the solution to optimization (P1b) under constraint 1 and by solely using local information on $\mathscr{N}_i$.*

In this paper, for the sake of simplicity it is assumed that the nodes know an upper bound on the size of the network, denoted by $\overline{n}$. Otherwise, the value can be estimated distributively using existing methods such as the ones presented in [31], [32].

4 Distributed Link Removal Algorithm with Connectivity Preservation

In this section, we formulate an optimization problem whose solution is equal to a suboptimal solution to (P1b) and describe a method to solve it in a distributed manner.

4.1 A Suboptimal Solution to Link Removal Problem

Based on the matrix perturbation analysis presented in Section 2.3, the matrix ΔA^- in (P1b) can be treated as a perturbation added to the matrix $\mathscr{A}(\mathscr{G}_0)$. It is worth to note that perturbation analysis has also been used to study link addition [33] and node removal problems [34]. First, observe that since $\mathscr{G}_0$ is connected, then matrix $\mathscr{A}(\mathscr{G}_0)$ is nonnegative and irreducible. This implies that eigenvalue $\lambda_{max}(\mathscr{A}(\mathscr{G}_0))$ is simple [18, Theorem 8.4.4]. Using Lemma 1 and considering the matrix $\overline{A} = \mathscr{A}(\mathscr{G}_0) + \Delta A$ with $\Delta A = \Delta A^-$, the following lemma and corollary, whose proofs can be found in the appendices, provide condition so that

$$\overline{\lambda}_{max} = \lambda_{max}(\mathscr{A}(\mathscr{G}_0)) + \frac{v_n^T \Delta A v_n}{v_n^T v_n} + \mathrm{O}\big(\|\Delta A\|^2\big) \tag{7}$$

where $\overline{\lambda}_{max}$ is the largest eigenvalue of $\overline{A}$ and v_n is the eigenvector associated with $\lambda_{max}(\mathscr{A}(\mathscr{G}_0))$.

Lemma 2. *For a given number of link removals m_e, assume that the following condition on graph $\mathscr{G}_0$ is satisfied:*

$$|\lambda_{max}(\mathscr{A}(\mathscr{G}_0)) - \lambda_{n-1}(\mathscr{A}(\mathscr{G}_0))| > 2\sqrt{2m_e}. \tag{8}$$

Then, Equation (7) *holds.*

For a single link removal, i.e. $m_e = 1$, we can compute $\|\Delta A^-\|_2 = \|\mathscr{A}_{l_{\mathscr{G}_0}}\|_2$ explicitly and thus improve the bound in (8).

Corollary 1. *For $m_e = 1$, the condition* (8) *is given by*

$$|\lambda_{max}(\mathscr{A}(\mathscr{G}_0)) - \lambda_{n-1}(\mathscr{A}(\mathscr{G}_0))| > 2.$$

Note that the higher order term in (7) depends on the eigenvalues and eigenvectors of adjacency matrix $\mathscr{A}(\mathscr{G}_0)$ as can be seen in Section 5.1 and can be estimated/computed in a distributed manner, for example using the method proposed in [35]. In the rest of the paper, for simplicity we neglect the higher order term in (7). Under assumption (8), from (7) with $\Delta A = -\Delta A^-$ and since deleting a link will always decrease λ_{max} [36, Proposition 1.3.10] we then have

$$\overline{\lambda}_{max} \le \lambda_{max}(\mathscr{A}(\mathscr{G}_0)) + \Delta\lambda_{max}$$

where $\Delta\lambda_{max} = \frac{v_n^T(-\Delta A^-)v_n}{v_n^T v_n}$ and can be computed as

$$\begin{aligned} \Delta\lambda_{max} &= \frac{v_n^T \left(-\sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} A_{l_{\mathscr{G}_0}}\right) v_n}{v_n^T v_n} \\ &= -\frac{2}{v_n^T v_n} \sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} v_{n,i} v_{n,j} = -\Delta\bar{\lambda}_{max}. \end{aligned}$$

Note that since $\mathscr{A}(\mathscr{G}_0)$ is non-negative and irreducible matrix, we have $v_{n,i} > 0$, $\forall i$ which yields $\Delta\bar{\lambda}_{max} > 0$. Therefore we can write

$$\begin{aligned} \min_{\bar{y}} \overline{\lambda}_{max} &\le \lambda_{max}(\mathscr{A}(\mathscr{G}_0)) + \min_{\bar{y}}(-\Delta\bar{\lambda}_{max}) \\ &\le \lambda_{max}(\mathscr{A}(\mathscr{G}_0)) - \max_{\bar{y}} \Delta\bar{\lambda}_{max}. \end{aligned}$$

Hence, a suboptimal solution to (P1b) can be obtained by solving the following optimization problem

$$\begin{aligned} \max_{\bar{y}} \quad & \Delta\bar{\lambda}_{max} \\ \text{s.t.} \quad & \lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e})) > 0, \\ & 1_{|\mathscr{E}_0|}^T \bar{y} \le m_e, \\ & \bar{y} \in \{0,1\}^{|\mathscr{E}_0|} \end{aligned} \tag{P1c}$$

where

$$\Delta\bar{\lambda}_{max} = \frac{2}{v_n^T v_n} \sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} v_{n,i} v_{n,j}, \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2813426, IEEE Transactions on Network Science and Engineering

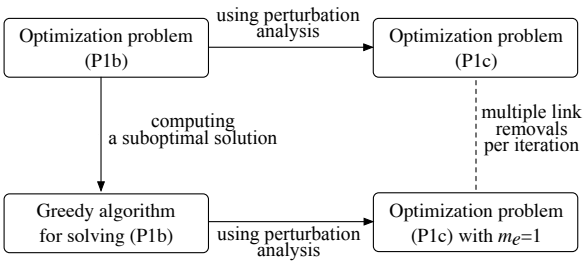


Fig. 1: Methods for computing a suboptimal solution to (P1b) using iterative and simultaneous link removal strategies

and $v_{n,i}$ denote the i-th element of the eigenvector v_n.

Remark 4. *Condition* (8) *can be checked in a distributed manner using the method in [37] or [35]. Note that network with a large spectral gap has been studied in the literature, see e.g., [38]. It is known that good expander graphs have large spectral gap [39, p. 182]. For example, the spectral gap of a complete graph with n nodes is equal to n. As can be seen from the previous discussion, when condition* (8) *is satisfied, it is sufficient to focus only on the movement of λ_{max} in order to predict or estimate $\overline{\lambda}_{max}$. In addition, large spectral gap also allows us to neglect the high order term in* (7) *[21, p.185] while obtaining a suboptimal solution close to the global one as also indicated by the simulation results in Section 6.1.*

When condition (8) is not satisfied or the spectral gap $(\lambda_{\max} - \lambda_{n-1})$ is small, distributed strategy proposed in the paper can still be used to compute a suboptimal solution and the solution can be improved by iteratively removing one (or some) link at a time from the graph $\mathscr{G}_0$ for a total maximum m_e iterations as shown in Section 6.1. This iterative strategy can also be seen as a greedy algorithm for solving optimization problem (P1b) as illustrated in Fig. 1. In addition, by iteratively removing one link at a time, the error introduced by the matrix perturbation will be minimized at each iteration. The iterative strategy is also suitable for some classes of network with $v_{n,i} = v_{n,j}$ for all $i \neq j$ such as in regular graph, where the first-order perturbation cannot predict the movement of eigenvalue λ_{max}. Another strategy is by incrementally reducing the weight of the link, namely the matrix ΔA in (7) is given by $\varepsilon \Delta A^-$ as shown in Section 6.1. By choosing $\varepsilon \ll 1$, equation (7) will still hold since the perturbation matrix ΔA is sufficiently small.

4.2 Choosing Threshold $\overline{\varepsilon}$ for Distributed Power Iteration Method

In order to solve optimization problem (P1c) distributively, we utilize distributed power method in Section 2.4 to distributively estimate v_n. To this end, we define the primitive matrix P in (4) as

$$P = I_n + \mathscr{A}(\mathscr{G}_0). \tag{10}$$

It is clear that the eigenvalues of P and $\mathscr{A}(\mathscr{G}_0)$ satisfy $\lambda_i(P) = 1 + \lambda_i(\mathscr{A}(\mathscr{G}_0))$ and they share the same set of eigenvectors. Since matrix P is primitive, we have $\rho(P) = \lambda_{max}(P)$ and thus $\rho(P) = 1 + \lambda_{max}(\mathscr{A}(\mathscr{G}_0))$. Moreover, since $\mathscr{G}_0$ is connected, the eigenvalue $\lambda_{max}(P)$ is simple and its corresponding eigenvector has positive elements.

Since power iteration method converges asymptotically, in the following we provide a guideline for choosing threshold $\overline{\varepsilon}$ in (5) which guarantees the solution to (P1c) based on the estimated eigenvector $\hat{v}_n$, are equal to the one using the true eigenvector v_n.

Lemma 3. *For connected graph $\mathscr{G}_0$ and a maximum number of links to be removed m, let the threshold $\overline{\varepsilon}$ in* (5) *be chosen as*

$$\overline{\varepsilon} < \frac{\overline{\gamma}(\lambda_n(\mathscr{A}(\mathscr{G}_0)) - \lambda_{n-1}(\mathscr{A}(\mathscr{G}_0)))}{(2+\overline{\gamma})(d_{max}+1)} \tag{11}$$

where d_{max} is the maximum degree of $\mathscr{G}_0$,

$$\begin{aligned} &\overline{\gamma} = \frac{\sqrt{b^2 - 4c} - b}{2}, \\ &b = \frac{1}{2}(\hat{v}_{n,i} + \hat{v}_{n,j} + \hat{v}_{n,s} + \hat{v}_{n,q}), \;\; c = \frac{1}{2}(\hat{v}_{n,s}\hat{v}_{n,q} - \hat{v}_{n,i}\hat{v}_{n,j}) \end{aligned} \tag{12}$$

and the pairs $(i,j) \in \mathscr{E}_0$ and $(s,q) \in \mathscr{E}_0$ are given by

$$\begin{aligned} \text{argmin} \quad & (\hat{v}_{n,i}\hat{v}_{n,j} - \hat{v}_{n,s}\hat{v}_{n,q}) \\ s.t. \quad & (i,j) \in \mathscr{E}_0^{m+1}, (s,q) \in \mathscr{E}_0^{m+1}, \\ & \hat{v}_{n,i}\hat{v}_{n,j} > \hat{v}_{n,s}\hat{v}_{n,q} \end{aligned} \tag{13}$$

with $\mathscr{E}_0^{m+1} \subset \mathscr{E}_0$ is the set of links with $(m+1)$ largest $\hat{v}_{n,i}\hat{v}_{n,j}$. We then have

$$\text{argmax} \sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} \hat{v}_{n,i}\hat{v}_{n,j} = \text{argmax} \sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} v_{n,i} v_{n,j}$$

with $1^T_{|\mathscr{E}_0|}\bar{y} \leq m$.

Proof. See Appendix C. □

In order to compute (11), the nodes start with a sufficiently small $\overline{\varepsilon}$ to obtain $\hat{v}_n$ and then improve the estimates by reducing $\overline{\varepsilon}$ such that condition in (11) is satisfied. While d_{max} can be computed distributively using max-consensus protocol in (2) with $x_i(0) = d_i$, we only need information on the lower bound of the spectral gap for checking condition (11). If the spectral gap is known to be sufficiently large, we can replace $\lambda_n(\mathscr{A}(\mathscr{G}_0)) - \lambda_{n-1}(\mathscr{A}(\mathscr{G}_0))$ in (11) with a sufficiently small value, e.g., 0.0001. The set $\mathscr{E}_0^{m+1} \in \mathscr{E}_0$ in (13) can be computed either by iteratively performing maximum consensus protocol, distributed sorting [40], or distributively propagating the local computation $\hat{v}_{n,i}\hat{v}_{n,j}$ with $(i,j) \in \mathscr{E}_0$ to the rest of the network [37], [41]. Furthermore, If $m = 1$, the pairs (i,j) and (s,q) in (12) are then given by the ones with the largest and second largest values of $\hat{v}_{n,i}\hat{v}_{n,j}$ respectively.

4.3 Distributed Connectivity Verification

Let the link (i^*, j^*) be given by the solution to

$$(i^*, j^*) = \underset{(i,j)\in\mathscr{G}_0}{\text{argmax}} \, \hat{v}_{n,i}\hat{v}_{n,j}. \tag{14}$$

In the following we utilize the max-consensus protocol (2) to check distributively connectivity of graph $\bar{\mathscr{G}}_e = (\mathscr{V}, \mathscr{E}_0 \setminus (i^*, j^*))$. To this end, we first set initial values in (2) as $x_i(0) = 0, \forall i \in \mathscr{V}$. After computing the candidate link to be removed (i^*, j^*), either node i^* or j^* modifies its initial value to $x_{i^*}(0) = 1$ or $x_{j^*}(0) = 1$ (nodes i^*, j^* can either locally negotiate, e.g. based on the estimate $\hat{v}_{n,i}$, or randomly choose, e.g. by flipping a coin, among themselves to decide which one modifies its initial value to one), while the remaining nodes keep their initial values unchanged. Without loss of generality, assume that $x_{i^*}(0) = 1$ and $x_j = 0, \forall j \neq i^*$. All nodes then execute (2) on graph $\bar{\mathscr{G}}_e = (\mathscr{V}, \mathscr{E}_0 \setminus (i^*, j^*))$, which means that

Algorithm 1 Distributed connectivity verification for $\bar{\mathscr{G}}_e = (\mathscr{V}, \mathscr{E}_0 \setminus (i^*, j^*))$

Require: $\mathscr{G}_0 = (\mathscr{V}, \mathscr{E}_0)$ is connected and a candidate link to be removed $(i^*, j^*) \in \mathscr{E}_0$ given in (14)
1: set $x_{i^*}(0) = 1$ and $x_j(0) = 0$ for all $j \neq i^*$
2: initialize $flag_i(0) = 0$ for all nodes
3: set $\bar{\mathscr{G}}_e = (\mathscr{V}, \mathscr{E}_0 \setminus (i^*, j^*))$
4: execute max-consensus (2) for $\bar{n}$ iterations on graph $\mathscr{G} = \bar{\mathscr{G}}_e$
5: **if** $x_{i^*}(\bar{n}) \neq x_{j^*}(\bar{n})$ **then**
6: $\bar{\mathscr{G}}_e$ is disconnected
7: $flag_{i^*}(0) \leftarrow 1$ and $flag_{j^*}(0) \leftarrow 1$
8: **else**
9: $\bar{\mathscr{G}}_e$ is connected
10: **end if**
11: execute max-consensus protocol (2) for $\bar{n}$ iterations with initial conditions equal to $flag_i(0)$ for $i = \{1, \cdots, n\}$
12: **if** $flag_i(\bar{n}) = 1$ **then**
13: node i knows that the graph $\bar{\mathscr{G}}_e$ is disconnected
14: **else**
15: node i knows that the graph $\bar{\mathscr{G}}_e$ is connected
16: **end if**

node i^* does not use the information it received from node j^* in its consensus protocol and vice versa. If the resulting graph $\bar{\mathscr{G}}_e$ is connected, then after $\bar{n}$ steps we have $x_i(\bar{n}) = 1, \forall i \in \mathscr{V}$. However, when the removal of link (i^*, j^*) results in disconnected graph $\bar{\mathscr{G}}_e$, then $x_{j^*}(\bar{n}) = 0$. To be more precise, the nodes are partitioned into two groups denoted by $\mathscr{V}_1$ and $\mathscr{V}_2$ where, with no loss of generality, $i^* \in \mathscr{V}_1$ and $j^* \in \mathscr{V}_2$ whose states are equal to $x_i(\bar{n}) = 1$, $i \in \mathscr{V}_1$ and $x_j(\bar{n}) = 0$, $j \in \mathscr{V}_2$ respectively. Hence, the pair of nodes (i^*, j^*) can estimate connectivity of the resulting network by checking whether $x_{i^*}(\bar{n}) = x_{j^*}(\bar{n})$ and broadcast the result to the remaining nodes in the network using maximum consensus protocol. The pseudo-code for distributed connectivity detection is shown in Algorithm 1.

Algorithm 1 can also be used to check connectivity of the resulting graph when multiple links are being removed simultaneously. To be more precise, let the set of links to be removed be given by $\mathscr{E}^c \subset \mathscr{E}_0$ with $|\mathscr{E}^c| = m_e > 1$. The initial states of nodes are chosen similar to the case of a single link removal (i.e., only one node (in the set $\mathscr{E}^c$) with initial value equal to one while the rest are zero) and all nodes execute max-consensus protocol (2) on graph $\bar{\mathscr{G}}_{m_e} = (\mathscr{V}, \mathscr{E}_0 \setminus \mathscr{E}^c)$. If there exists at least one pair of nodes $(i^*, j^*) \in \mathscr{E}^c$ that after $\bar{n}$ steps $x_{i^*}(\bar{n}) \neq x_{j^*}(\bar{n}) \neq 1$, which can be checked locally, it can then be concluded that the resulting graph $\bar{\mathscr{G}}_{m_e}$ is disconnected.

4.4 The Complete Distributed Link Removal Algorithm

The pseudo-code of the distributed algorithms to compute a suboptimal solution to (P1b) by removing a single link (resp. multiple links) iteratively is summarized in Algorithm 2 (resp. Algorithm 3). Note that removing multiple links may result in faster computation since the number of iterations in Algorithm 2 is less than m_e. However, the performance may degrade in comparison to the removal of one link at a time as will be shown in Section 6. Furthermore, by setting $m = m_e$, Algorithm 3 can also be applied to remove m_e links simultaneously. Both Algorithms 2 and 3 utilize (distributed) power iteration and max-consensus algorithms. As discussed in Section 2, the convergence rate of

Algorithm 2 Iteratively removing one link at a time from $\mathscr{G}_0$

Require: $\mathscr{G}_0$ is connected, threshold $\bar{\varepsilon}$ satisifes (11) and node i knows $\mathscr{N}_i$, m_e
1: set $\bar{y} = [0, \ldots, 0]^T \in \mathbb{R}^{|\mathscr{E}_0|}$
2: initialize $e = 1$
3: **while** $e \leq m_e$ **do**
4: estimate v_n distributively using (4), (5) whose estimation is given by $\hat{v}_n$
5: node i computes $l^*_{\mathscr{G}_{e-1},i} = \text{argmax}_{\bar{y}_{l_{\mathscr{G}_{e-1}}}} \hat{v}_{n,i}\hat{v}_{n,j}$ for $j \in \mathscr{N}_i$
6: compute $l^*_{\mathscr{G}_{e-1}} = \text{argmax}\, \hat{v}_{n,i}\hat{v}_{n,j}$ with $(i,j) \in l^*_{\mathscr{G}_{e-1},i}$ using max-consensus (2) with $x_i(0) = \hat{v}_{n,i}\hat{v}_{n,j}$, $(i,j) \sim l^*_{\mathscr{G}_{e-1},i}$
7: check connectivity of $\bar{\mathscr{G}}_{e-1} = (\mathscr{V}, \mathscr{E}_{e-1} \setminus (i^*, j^*))$ using Algorithm 1
8: **if** $\bar{\mathscr{G}}_{e-1}$ is disconnected **then**
9: back to steps 5-7 with $\mathscr{N}_{i^*} \leftarrow \{\mathscr{N}_{i^*} \setminus j^*\}$ and $\mathscr{N}_{j^*} \leftarrow \{\mathscr{N}_{j^*} \setminus i^*\}$
10: **if** $\mathscr{N}_i = \emptyset$ for all i **then**
11: **break**
12: **end if**
13: **else**
14: continue to step 16
15: **end if**
16: $\bar{y}_{l^*_{\mathscr{G}_{e-1}}} \leftarrow 1$
17: update $\mathscr{G}_e \leftarrow \bar{\mathscr{G}}_{e-1}$
18: $e \leftarrow e + 1$
19: **end while**
20: solution to (P1c) is given by $\bar{y}$

Algorithm 3 Removing m links at a time from the network $\mathscr{G}_k$

Require: graph $\mathscr{G}_k$ is connected, threshold $\bar{\varepsilon}$ satisfies (11) and node i knows $\mathscr{N}_i$, m
1: set $\bar{y} = [0, \ldots, 0]^T \in \mathbb{R}^{|\mathscr{E}_0|}$
2: estimate v_n corresponding to $\lambda_{max}(\mathscr{A}(\mathscr{G}_k))$ distributively
3: **for** $e = 1, \ldots, m$ **do**
4: execute steps 5–16 in Algorithm 2
5: update $\mathscr{N}_{i^*} \leftarrow \{\mathscr{N}_{i^*} \setminus j^*\}$ and $\mathscr{N}_{j^*} \leftarrow \{\mathscr{N}_{j^*} \setminus i^*\}$ where $(i^*, j^*) \sim l^*_{\mathscr{G}_k}$
6: **end for**
7: the set of links to be removed is given by $\bar{y}$

power iteration is equal to $|\lambda^*(P)/\lambda_{max}(P)|^2$ where P is given in (10) while the complexity of max-consensus algorithm is $O(n)$. Moreover, note that the max-consensus protocol (2) is used at each iteration of the power method (namely for computing $\|P\hat{v}_n(t)\|_\infty$ and checking stopping condition (5)) and also for computing (9) and checking the connectivity (Algorithm 1).

In the following we provide some remarks on the proposed distributed algorithms.

(i) It is worth noting that initial values used to estimate the eigenvector and to check the connectivity are not the true initial values of the nodes and thus do not contain any physical meaning. Therefore, the proposed algorithm preserves privacy of nodes in a network in terms of their true initial values and does not collide with the privacy preserving consensus , e.g., the one proposed in [42].

(ii) The set of links that disconnect the resulting network obtained at each iteration in Algorithm 2 can be saved and those links will be excluded in the next iteration to shorten

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2813426, IEEE Transactions on Network Science and Engineering

the time required for checking the connectivity of the resulting network. In addition, for a large size network, in many situations its eigenvector does not change much by a small perturbation (i.e., single link removal) [43]. Hence, the estimated eigenvector at the current iteration t can be used as an initial value $\hat{v}_n(0)$ for the next iteration $t+1$ to speed up the convergence of power method.

(iii) When the initial network $\mathscr{G}_0$ is disconnected, then algorithms 2 or 3 can be applied in parallel to all subnetworks.

(iv) As shown in Algorithms 2 and 3, the proposed approach is not only able to find a suboptimal solution to (P1b) distributively but also to avoid the combinatorial issue without applying any relaxation to the Boolean variable.

(v) When graph $\mathscr{G}_0$ is given by a weighted undirected graph, Equation (9) can then be written as

$$\Delta\bar{\lambda}_{max} = \frac{2}{v_n^T v_n} \sum_{l_{\mathscr{G}_0}=1}^{|\mathscr{E}_0|} \bar{y}_{l_{\mathscr{G}_0}} w_{ij} v_{n,i} v_{n,j} \tag{15}$$

where w_{ij} denote the weight of link (i,j). Hence, the algorithms 2 and 3 can be extended in a straightforward manner to solve (P1c) in a distributed manner with $\Delta\bar{\lambda}_{max}$ is defined in (15) and provided that each node i knows w_{ij} for $j \in \mathscr{N}_i$.

(vi) The proposed distributed algorithm can also be extended when desired critical value β_c^* is given and the following additional constraint in (P1b) is required to be satisfied [3]

$$\lambda_{max}(\mathscr{A}(\bar{\mathscr{G}}_{m_e})) < \beta_c^*.$$

(vii) The link removal implies that the speed of convergence to consensus, defined by $\lambda_2(\mathscr{L})$ be reduced. In order to study the trade-off between the speed for reducing the spread of disease and the convergence speed to consensus, we can formulate and solve the following optimization problem:

$$\begin{aligned} \min_{\Delta\mathscr{E}^-} \quad & (1-\zeta)\lambda_{max}(\mathscr{A}(\bar{\mathscr{G}}_{m_e})) - \zeta\lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e})) \\ \text{s.t.} \quad & |\Delta\mathscr{E}^-| \le m_e, \ \Delta\mathscr{E}^- \subseteq \mathscr{E}_0, \\ & \lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e})) > 0, \end{aligned}$$

where $\zeta \in [0,1]$ is a constant representing the trade-off between the two metrics. Similarly, the value of $\lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e}))$ can be expressed as $\lambda_2(\mathscr{L}(\bar{\mathscr{G}}_{m_e})) = \lambda_2(\mathscr{L}(\bar{\mathscr{G}}_0)) + \Delta\lambda_2$. Note that λ_2 can also be approximated similarly in a distributed manner [2], [41].

(viii) The proposed distributed strategy can also be applied to link addition problem whose goal is to maximize $\lambda_{max}(\mathscr{A})$ (e.g., maximizing the spread of information in a network). Similary, by using perturbation analysis as presented in Section 4.1 the problem can be approximated as to maximize $\Delta\lambda_{max}$ and thus, the strategy described in Section 4.2 can be utilized to compute a suboptimal solution. Another possible objective is to maximize the algebraic connectivity, i.e., $\lambda_2(\mathscr{L})$ as discussed in [41].

5 Optimality and Topological Insight

This section discusses the optimality of the proposed approach in comparison to the brute-force search in the presence of global network topology information. Moreover, the topological insight of the proposed algorithm is also investigated.

5.1 Optimality of the Solution

In the following, we analyze the error between the solutions to (P1b) and (P1c). First, we discuss the optimality gap between the solution to (P1c) based on the simultaneous link removal strategy and the global solution to (P1b). Assume that condition (8) is satisfied and let the solution to optimization (P1c) for the simultaneous removal of m_e links be given by $\mathscr{E}^*$. The exact largest eigenvalue of the perturbed matrix by removing the links $\mathscr{E}^*$ can be computed as [21]

$$\begin{aligned} \bar{\lambda}_{max}^{\mathscr{E}^*} =& \lambda_{max} - \sum_{(i^*,j^*)\in\mathscr{E}^*} 2v_{n,i^*}v_{n,j^*} \\ &+ \sum_{j=1}^{n-1} \frac{\left[\sum_{(i^*,j^*)\in\mathscr{E}^*}\left(v_{j,i^*}v_{n,j^*} + v_{j,j^*}v_{n,i^*}\right)\right]^2}{\lambda_{max}-\lambda_j} + \mathrm{O}\left(\|\Delta A^-\|^3\right). \end{aligned} \tag{16}$$

Similarly, let the solution to the original optimization problem (P1b) be given by the set $\overline{\mathscr{E}}$. The exact largest eigenvalue of the perturbed matrix by removing the links $\overline{\mathscr{E}}$ is then given by

$$\begin{aligned} \bar{\lambda}_{max}^{\overline{\mathscr{E}}} =& \lambda_{max} - \sum_{(\bar{i},\bar{j}\in\overline{\mathscr{E}})} 2v_{n,\bar{i}}v_{n,\bar{j}} \\ &+ \sum_{j=1}^{n-1} \frac{\left[\sum_{(\bar{i},\bar{j}\in\overline{\mathscr{E}})}\left(v_{j,\bar{i}}v_{n,\bar{j}} + v_{j,\bar{j}}v_{n,\bar{i}}\right)\right]^2}{\lambda_{max}-\lambda_j} + \mathrm{O}\left(\|\Delta A^-\|^3\right). \end{aligned} \tag{17}$$

Therefore, the optimality gap between the solutions to (P1b) and (P1c), i.e., $|\bar{\lambda}_{max}^{\overline{\mathscr{E}}} - \bar{\lambda}_{max}^{\mathscr{E}^*}|$ is equal to the difference between (17) and (16). As can be observed, the gap depends on all eigenvalues together with the associated eigenvectors of adjacency matrix $\mathscr{A}(\mathscr{G}_0)$. Note that for a network with large spectral gap $(\lambda_{max}-\lambda_j)$ and $j \neq n$, the second and higher terms in (16) and (17) become small and thus can be neglected. Hence, in this case the gap depends solely on the eigenvector associated with λ_{max}. This also indicates that the gap between (P1c) and (P1b) is small.

Next, we discuss the gap between the solution obtained from Algorithm 2 and the global solution to (P1b). As can be seen from Fig. 1, the gap is equal to the summation of the error from the greedy algorithm and the perturbation based approximation. It is unknown if there is a bound on the error between the solution to (P1b) and the one based on greedy algorithm unless the function $\lambda_{max}(\mathscr{A}(\mathscr{G}_0))$ shares a similar property to supermodular function of the edges, which to the best of the authors' knowledge, is still an open problem. The gap between the greedy based algorithm and its perturbation based approximation is given by the differences between (16) and (17). From the simulations presented in Section 6.1, it is observed that when the spectral gap $(\lambda_{max}-\lambda_{n-1})$ is large, the gap between the solutions is quite small. In addition, it is worth noting that the value of

$$\left[\bar{\lambda}_{max}^{\mathscr{E}^*} - (\lambda_{max} - \sum_{(i^*,j^*)\in\mathscr{E}^*} 2v_{n,i^*}v_{n,j^*})\right]$$

for the simultaneous multiple link removal strategy is larger than single link removal strategy, meaning that the first-order approximation poorly approximates the exact eigenvalue of the perturbed matrix for the simultaneous multiple link removal case. This is one of the reasons why in general removing iteratively one link at at time yields a better solution compared to simultaneous link removal strategy as mentioned in Section 4.1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2813426, IEEE Transactions on Network Science and Engineering

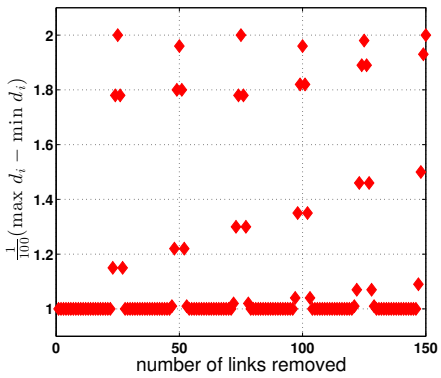


Fig. 2: Average degree gap for link removals of regular graph, which are randomly generated for 100 times, with network size of 50 nodes and node degree equal to 20. The number of links being removed varies between 1 and 150 and a single link is removed iteratively. As can be seen, the proposed strategy tends to balance the degree distribution of the network

5.2 Topological Insights into the Proposed Strategy

In the following we provide topological insights into Algorithm 2. We start by introducing the degree's gap δ of a network which is defined as

$$\delta = \max_i d_i - \min_j d_j. \tag{18}$$

We then have the following as a result of applying Algorithm 2 to a complete graph whose proof can be found in Appendix D.

Lemma 4. *Consider a complete n-node network of degree* $(n-1)$ *with* $\delta = 0$ *and assume that* $m_e \leq \lfloor \frac{n}{2} \rfloor$. *Then, the degree's gap of the resulting network after removing* m_e *links by applying algorithm 2 satisfies* $\delta \leq 1$.

In addition, we perform simulations on 100 randomly generated regular graphs with size of 50 nodes and degree equal to 20. For each graph, we apply Algorithm 2 where the number links to be removed m_e is varied between 1 and 150. Figure 2 shows the average degree's gap δ for each m_e. As can be observed, the degree's gap of the resulting graph is bounded by two, i.e. $\delta \leq 2$. It is shown from Lemma 4 and simulations results in Fig. 2 that algorithm 2 tends to reduce the degree's gap or balance the degree distribution of a network, i.e., to make the resulting network to be as regular as possible, which supports the claim in [1]. We will also show later in Section 6 that a similar observation can also be found in several random networks and also real network such as power grid. Note that it is shown in [44] that the difference between the eigenvector v_n of a graph and its degree vector to be the smallest (i.e., $q^T q = 0$ with $q = v_n - \frac{d}{\sqrt{d^T d}}$) when $\lambda_{max} = W_2/W_1$ where W_k denotes the total number of k hop walks between any two nodes which can be the same. In other words, for this type of graph the element of the eigenvector $v_{n,i}$ is "proportional" to the degree of node. Therefore, for network with $\lambda_{max} = W_2/W_1$ (e.g., large size Erdös-Rényi random graphs), the proposed strategy in Algorithm 2 is equivalent to removing links whose corresponding nodes have the largest degree which is similar to the strategy presented in [16].

The observation in Lemma 4 does not always hold when multiple links are removed at a time as given by algorithm 3 which will be shown later in Section 6. The observation that Algorithm 2 tends to reduce the degree's gap may be used for reducing the search space of the original optimization problem (P1b).

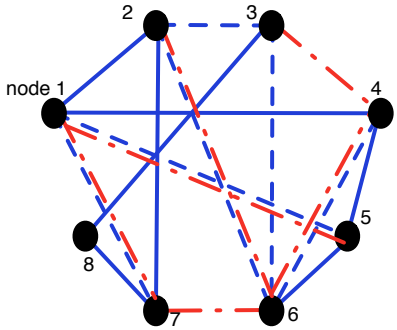


Fig. 3: A network $\mathcal{G}_0$ consisting of 8 nodes. The dashed and dash-dot lines are the global and suboptimal solutions from Algorithm 2 respectively for $m_e = 6$

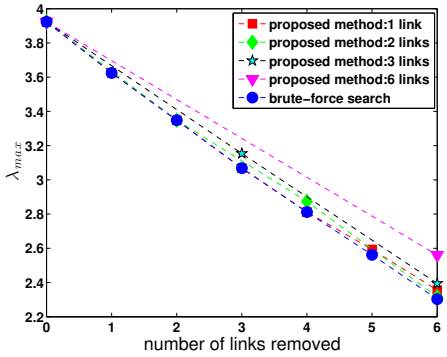


Fig. 4: Comparison between the proposed distributed design method for different number of links removed at each iteration and brute-force search for a network given in Fig. 3

6 Numerical Examples

In this section, we demonstrate and evaluate the proposed distributed strategy via several numerical examples.

6.1 Demonstration and Comparison with Centralized Approach

First, we demonstrate the distributed design procedure and compare it with the centralized approach in the present of global network topology information. We consider a network of 8 nodes whose initial structure $\mathcal{G}_0$ is shown in Fig. 3. We choose a small size network so that the comparison with the centralized approach, which in general is NP-hard, becomes possible. The number of links to be removed m_e is varied between 1 and 6. We apply algorithm 2 in order to remove iteratively a single link at a time. At each iteration $i = \{1, \ldots, m_e\}$, the nodes first estimate v_n distributively. The estimate of v_n for graph $\mathcal{G}_0$ is given by

$$\hat{v}_n = [0.799, 0.857, 0.786, 0.83, 0.67, 1, 0.779, 0.399]^T.$$

In addition, we also apply Algorithm 3 and modify the number of links removed at each iteration to 2, 3 and 6 links. As a comparison we also solve the original optimization problem (P1b) by performing a brute-force search. Solutions to (P1c) using Algorithms 2, 3 and the optimal solution to (P1b) are shown in Fig. 4. First, it can be observed from Fig. 4 that the Algorithm 2 results in a better performance in comparison to the simultaneous link removal strategy for $m_e = 6$. Next, for the iterative single link

Algorithm 4 Iterative link removals based on continuous perturbation matrix

Require: graph $\mathscr{G}_0$ is connected, m_e, $\varepsilon \ll 1$
1: set $\bar{y} = [0,\dots,0]^T \in \mathbb{R}^{|\mathscr{E}_0|}$
2: initialize $A = \mathscr{A}(\mathscr{G}_0)$
3: **while** $\|A\|_0 > 2(|\mathscr{E}_0| - m_e)$ **do**
4: compute the optimal link (pair of nodes) $(i^*, j^*) \sim l^*_{\mathscr{G}_0}$ using algorithm 2 for the matrix A
5: $A \leftarrow A - \varepsilon A_{l^*_{\mathscr{G}_0}}$
6: **end while**
7: **if** $[A]_{ij} > 0$ **then**
8: $[A]_{ij} = 1$
9: **end if**
10: **if** $[\mathscr{A}_{\mathscr{G}_0} - A]_{ij} = 1$ **then**
11: $y_{l_{\mathscr{G}_0}} \leftarrow 1$ where $l_{\mathscr{G}_0} \sim (i,j)$
12: **end if**
13: the set of links to be removed is given by $\bar{y}$

removal strategy and $m_e = 1,2,3$, the solutions to (P1c) are nearly optimal, and in fact equivalent to that of (P1b), i.e., there is no performance loss in spite of the absence of the global network topology. Note that, the value $|\lambda_{max}(\mathscr{A}(\mathscr{G}_i)) - \lambda_{n-1}(\mathscr{A}(\mathscr{G}_i))|$ are equal to 2.6291, 2.2516, 2.0540 for $i = 0,1,2$ respectively which satisfy (8). However, as m_e increases, the gap between the values of λ_{max} and the solutions to (P1c) and (P1b) becomes larger as the spectral gap becomes smaller. Note that, it can still be seen that the resulting λ_{max} in (P1c) is still close to the one in (P1b). Next, we look at the results for removing multiple links at each iteration (Algorithm 3). When $m_e = 4$, removing a single link at each iteration yields a better performance when compared to removing 2 links. However, for $m_e = 6$ removing 2 links at each iteration results in a better performance than the single link removal. Hence, it can be observed that in general removing a small number of links (a single link or 2 links) at each iteration yields a better performance than removing 3 links or more links simultaneously. The question on what is the optimal number of links to be removed at each iteration is subject to our future work. The degree's gap δ of the original network is equal to three and the degree's gap from both the global and suboptimal solutions are equal to one.

Finally, we compare the performance of Algorithm 2 with the one where ΔA in (3) is replaced by a "continuous" perturbation matrix $\varepsilon(\Delta A^-)$ with $\varepsilon \ll 1$ and $\Delta A^- = \bar{y}_{l_{\mathscr{G}_0}} A_{l_{\mathscr{G}_0}}$. Specifically, we modify Algorithm 2 and apply a simple threshold operation as described in Algorithm 4 to compute a set of links to be removed. The results are summarized in Table 1 in which Algorithm 2 yields a better result compared to Algorithm 4 for $m_e = 1,\cdots,4$.

TABLE 1: Comparison of $\lambda_{max}(\mathscr{A}(\mathscr{G}_{m_e}))$ obtained from Algorithms 2 and 4 for the network depicted in Fig. 3

m_e	ΔA^-	$\varepsilon(\Delta A^-)$	
		$\varepsilon = 0.0001$	$\varepsilon = 0.00001$
1	3.624	3.657	3.657
2	3.348	3.497	3.398
3	3.068	3.244	3.214
4	2.812	2.948	2.954
5	2.593	2.659	2.592
6	2.355	2.302	2.355

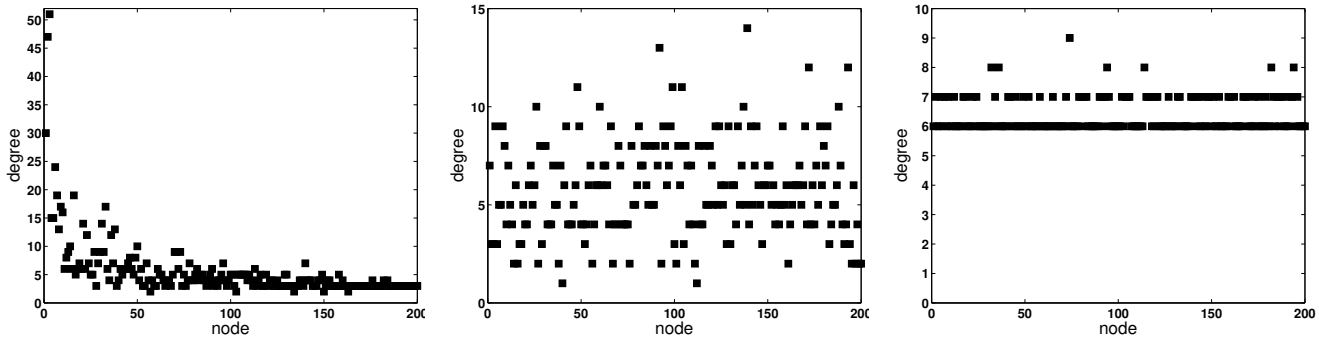
Note that the solutions from both Algorithms 2 and 4 are close to the global optimal ones as confirmed from Fig. 4. The computational time required by Algorithm 2 is much less than the one needed by Algorithm 4 in terms of the number of estimation of ν_n. In addition, it can be observed that a smaller ε does not necessarily result in a better performance, e.g., for the case of $m_e = 4,6$. This may also indicate that choosing the optimal threshold for Algorithm 4 is not a trivial problem. However, this issue is beyond the scope of the current paper.

6.2 Evaluation on Random Networks

In the following we evaluate the proposed strategy on random networks, namely Barabási-Albert Scale Free model, Gilbert stochastic and Watts-Strogatz models consisting of 200 nodes and with number of links $|\mathscr{E}_0|$ equal to 570, 574 and 639 respectively. These random networks share special properties such as their degree distribution as illustrated in Fig. 5 which may help us to gain insights on the proposed strategy as will be shown later. For each type of random network we first remove a single link using Algorithm 2 where the value m_e is varied between a single link and 30% of $|\mathscr{E}_0|$. In addition, we compare the proposed strategy with random link removal, while preserving the connectivity of the resulting network, as the simplest alternative of distributed strategy. It is clear from Fig. 6 that Algorithm 2 outperforms the random link removal strategy. Moreover, in order to obtain a desired λ_{max}, more links are required to be removed when using the random strategy which is undesired since it can deteriorate significantly the speed of the consensus algorithm.

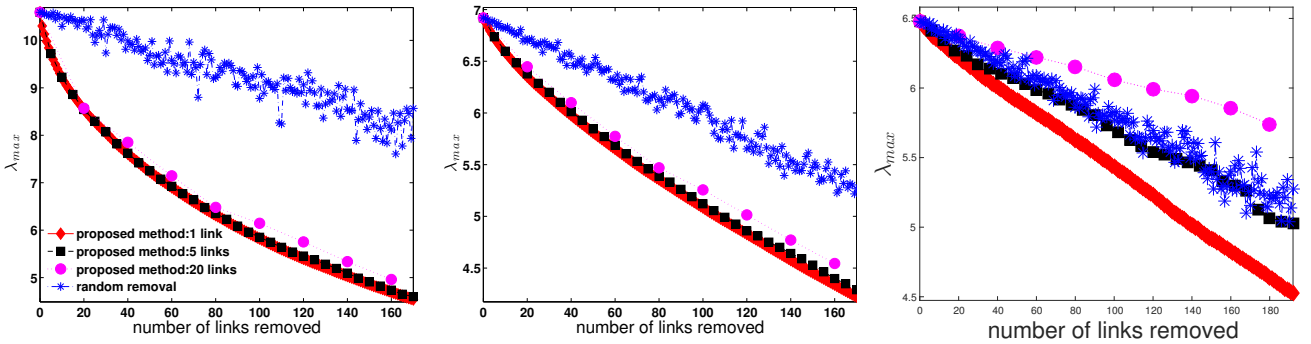
Next, we apply Algorithm 3 by removing 5 links and 20 links at each iteration. As can be observed from Figs. 6a and 6b, for Barabási-Albert Scale Free and Gilbert stochastic network models the resulting value of λ_{max} from removing 5 and 20 links at each iteration are close to the case of a single link removal. This means that for these network models, we can reduce the total computational effort of the proposed distributed strategy in terms of the number of iterations without harming the quality of its solutions. However, as can be seen from Fig. 6c, the gap of λ_{max} between a single and multiple link removals at each iteration for the Watts-Strogats model is quite large. Specifically, the performance of Algorithm 2 for 20 link removals at each iteration turns out to be worse than the random link removal strategy. In order to get a hint of the cause for this performance degradation, we investigate the topological property, namely the degree distribution of the resulting network given by Algorithms 2 and 3. To this end, we compute the degree's gap δ defined in (18) and the total difference of the degree in the network given by $\sum_{i\neq j} |d_i - d_j|$. As can be seen from Figs. 7 and 8, in general lower values of δ and $\sum_{i\neq j} |d_i - d_j|$ yields lower values of λ_{max}, i.e., results in a better performance. Furthermore, the solutions given by algorithm 2, i.e., by iteratively removing one link at a time, for all three random networks tend to reduce the degree's gap δ which support our analysis in Section 5.2. It is worth noting that for the case of Watts-Strogats network, removing 20 links at each iteration in general yields a larger degree's gap δ and total difference of the degree in comparison to the one obtained by the random link removals strategy as depicted in Figs. 7c and 8c. This is one of the reasons why removing 20 links at each iteration results in a worse performance compared to random link removal strategy. Hence, the simulation results suggest and we conjecture that when a network has similar degree distribution to that of regular graph, it is recommended to remove one or small number of links at a time in order to yield good quality of solutions. Note

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2813426, IEEE Transactions on Network Science and Engineering



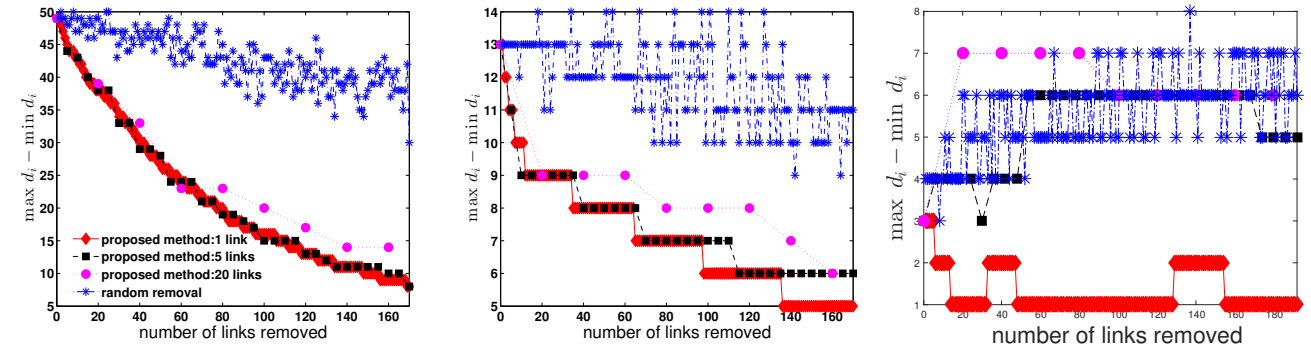
(a) Barabási-Albert scale free model (b) Gilbert stochastic model (c) Watts-Strogatz model

Fig. 5: Degree distribution of Random networks model of 200 nodes considered in the numerical example



(a) Barabási-Albert scale free model (b) Gilbert stochastic model (c) Watts-Strogatz model

Fig. 6: Proposed approach vs. random link removal strategy for random network of 200 nodes. The proposed distributed algorithm outperforms the random link removal strategy for all three random networks. For (a) and (b), multiple link removals at each step yields a performance near the single link removal at each step. On the other hand, for case (c) multiple link removals results in much worse performance compared to the single link removal and may be worse than the random link removal strategy

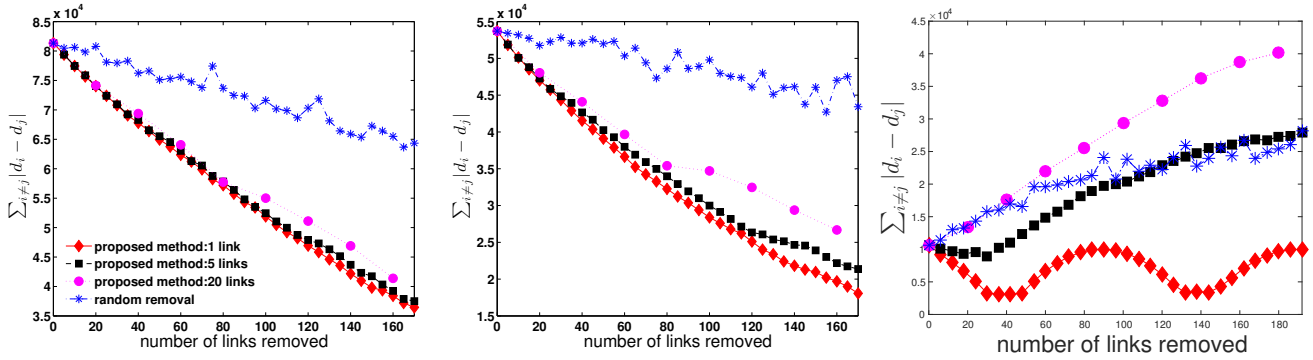


(a) Barabási-Albert scale free model (b) Gilbert stochastic model (c) Watts-Strogatz model

Fig. 7: Degree's gap for different link removals strategies. As can be observed, algorithm 2 tends to reduce the degree gap of the three random networks. On the other hand, algorithm 3 does not always reduce the degree's gap, especially for case (c)

that the observations on the degree's gap and the total difference of the degree might be used as a hint to decide whether a given number of links to be removed at each iteration is a good choice. In addition, this insight may also be combined with the proposed strategy to improve the quality of solutions obtained by removing multiple links at a time for Watts-Strogatz network.

Finally, let us assume that at each iteration m number of links are going to be removed from the networks in Fig. 5 while preserving their connectivity. Fig. 9 shows the number of links involved in the connectivity verification at each iteration for the proposed strategy (with different values of m), which are not less than m as can be observed from Algorithm 3. It can be seen that for the case of $m = 1,5$ the number of links involved in the connectivity verification for all the networks considered in Fig. 9 are equal to m. However, for the case of $m = 20$ and Watts-Strogatz network model the number of links involved in the connectivity verification are slightly larger than m while for the scale-free and Gilbert stochastic network models the number are equal to m.



(a) Barabási-Albert scale free model (b) Gilbert stochastic model (c) Watts-Strogatz model

Fig. 8: Total degree difference over all nodes in the network for different link removals strategies. Lower values reflect the balance of degree distribution of the network

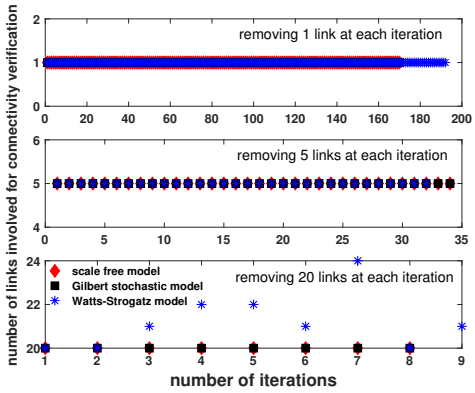


Fig. 9: Comparison of number of links involved in connectivity verification for different number of links removed at each iteration

7 Conclusion

This paper proposes distributed strategy for removing links from a network in the absence of global network structure information. The strategy is first to formulate a new optimization problem involving eigenvector corresponding to the largest eigenvalue and whose solution is equal to the original optimization problem. Distributed algorithm to estimate the eigenvector is then proposed which facilitates us to compute a suboptimal solution distributively. Even though in this paper we focus on link removal, the proposed distributed strategy is also applicable for computing suboptimal solution to link addition problem whose goal is to improve the network connectivity or maximize the spread of information in a network. Hence, the proposed distributed strategy has broad applications. Possible future work include solving link removal problem by considering the state of the nodes and also exploring alternative distributed strategy, e.g., by relaxing the binary decision variables (similar to the idea discussed in [45]) in combination with development of optimal threshold operation to recover the binary variables.

Acknowledgments

This work of A. Gusrialdi and Z. Qu is supported in part by US National Science Foundation under grants ECCS-1308928 and CCF-0956501, by US Department of Energy's award DE-EE0006340, by US Department of Transportation's award DTRT13-G-UTC51, by L-3 Communication's contract 11013I2034, by Leidos' contract P010161530, and by Texas Instruments' award. The work of S. Hirche is funded by the German Research Foundation (DFG) within the Priority Program SPP 1914 on Cyber-Physical Networking.

References

[1] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. van de Bovenkamp, D. Liu, and H. Wang, "Decreasing the spectral radius of a graph by link removals," *Physical Review E*, vol. 84, p. 016101, Jul 2011.

[2] A. Gusrialdi and Z. Qu, "Growing connected networks under privacy constraint: achieving trade-off between performance and security," in *IEEE Conference on Decision and Control*, 2015, pp. 312–317.

[3] A. N. Bishop and I. Shames, "Link operations for slowing the spread of disease in complex networks," *Europhysics Letters*, vol. 95, no. 18005.

[4] A. Gusrialdi, "Performance-oriented communication topology design for distributed control of interconnected systems," *Mathematics of Control, Signals, and Systems, special issue on Control, Communication, and Complexity*, vol. 25(4), pp. 559–585, 2013.

[5] M. Fardad, F. Lin, and M. R. Jovanovic, "Design of optimal sparse interconnection graphs for synchronization of oscillator networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2457–2462, 2014.

[6] J. Lunze, "Self-organising disturbance attenuation for synchronised agents with individual dynamics," *International Journal of Control*, vol. 88, no. 3, pp. 469–483, 2015.

[7] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *Proc. 22nd International Symposium on Reliable Distributed Systems*, 2003, pp. 25–34.

[8] Y. Koç, M. Warnier, P. Van Mieghem, R. E. Kooij, and F. M. Brazier, "A topological investigation of phase transitions of cascading failures in power grids," *Physica A: Statistical Mechanics and its Applications*, vol. 415, pp. 273–284, 2014.

[9] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. J. Pappas, "Optimal resource allocation for network protection against spreading processes," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 99–108, 2014.

[10] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 245–254.

[11] M. L. Overton, "On minimizing the maximum eigenvalue of a symmetric matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, no. 2, pp. 256–268, 1988.

[12] M. K. H. Fan and B. Nekooie, "On minimizing the largest eigenvalue of a symmetric matrix," in *IEEE Conference on Decision and Control*, 1992, pp. 134–139.

[13] Esri, "Enterprise gis and the smart electric grid," Tech. Rep., 2012.

[14] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *Security Privacy, IEEE*, vol. 7, no. 3, pp. 75–77, 2009.

[15] N. Li, N. Zhang, and S. Das, "Preserving relation privacy in online social network data," *IEEE Internet Computing*, vol. 15, no. 3, pp. 35–42, May 2011.

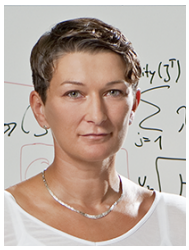
[16] I. Tomovski and L. Kocarev, "Simple algorithm for virus spreading control on complex networks," *IEEE Transactions on Circuits and Systems I*, vol. 59, no. 4, pp. 763–771, April 2012.

[17] F. Morbidi and A. Y. Kibangou, "A distributed solution to the network reconstruction problem," *Systems & Control Letters*, vol. 70, pp. 85–91, 2014.

[18] R. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, New York, 2009.
[19] F. Bullo, *Lectures on Network Systems*. Version 0.95(i), May 2017, with contributions by J. Cortes, F. Dörfler, and S. Martinez.
[20] B. Nejad, S. Attia, and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies," in *International Symposium on Information, Communication and Automation Technologies*, 2009, pp. 1–7.
[21] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*. Boston: Academic Press, 1990.
[22] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
[23] E. Ramirez-Llanos and S. Martinez, "Distributed stopping criteria for the power iteration applied to virus mitigation," in *49th Asilomar Conference on Signals, Systems and Computers*, Nov 2015, pp. 1328–1332.
[24] G. Canright, K. Engø-Monsen, and M. Jelasity, "Efficient and robust fully distributed power method with an application to link analysis," *Department of Computer Science, University of Bologna, Tech. Rep. UBLCS-2005-17*, pp. 2005–17, 2005.
[25] Z. Qu, C. Li, and F. Lewis, "Cooperative control with distributed gain adaptation and connectivity estimation for directed networks," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 450–476, 2014.
[26] C. Li and Z. Qu, "Distributed finite-time consensus of nonlinear systems under switching topologies," *Automatica*, vol. 50, no. 6, pp. 1626–1631, 2014.
[27] T. Charalambous, M. Rabbat, M. Johansson, and C. Hadjicostis, "Distributed finite-time computation of digraph parameters: left-eigenvector, out-degree and spectrum," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 137–148, 2016.
[28] J. Ghaboussi and X. S. Wu, *Numerical Methods in Computational Mechanics*. CRC Press, 2016.
[29] H. J. Ahn and B. Hassibi, "Global dynamics of epidemic spread over complex networks," in *IEEE Conference on Decision and Control*, Dec 2013, pp. 4579–4585.
[30] B. A. Prakash, D. Chakrabarti, N. Valler, M. Faloutsos, and C. Faloutsos, "Threshold conditions for arbitrary cascade models on arbitrary networks," *Knowledge and Information Systems*, vol. 33, no. 3, pp. 549–575, 2012.
[31] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed cardinality estimation in anonymous networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 645–659, March 2014.
[32] S. Zhang, C. Tepedelenlioglu, A. Spanias, and M. K. Banavar, "Node counting in wireless sensor networks," in *49th Asilomar Conference on Signals, Systems and Computers*, Nov 2015, pp. 360–364.
[33] M. Jalili and X. Yu, "Enhancement of synchronizability in networks with community structure through adding efficient inter-community links," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 2, pp. 106–116, April 2016.
[34] T. Watanabe and N. Masuda, "Enhancing the spectral gap of networks by node removal," *Physical Review E*, vol. 82, no. 4, p. 046102, 2010.
[35] A. Gusrialdi and Z. Qu, "Distributed estimation of all the eigenvalues and eigenvectors of matrices associated with strongly connected digraphs," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 328–333, 2017.
[36] D. Cvetković, P. Rowlinson, and S. Simić, *An Introduction to the Theory of Graph Spectra*. London Math. Society Student Texts 75, Cambridge University Press, Cambridge, 2010.
[37] C. Li and Z. Qu, "Distributed estimation of algebraic connectivity of directed networks," *Systems and Control Letters*, vol. 62, pp. 517–524, 2013.
[38] N. Saito and Y. Iba, "Probability of graphs with large spectral gap by multicanonical monte carlo," *Computer Physics Communications*, vol. 182, no. 1, pp. 223–225, 2011.
[39] V. Karyotis, E. Stai, and S. Papavassiliou, *Evolutionary Dynamics of Complex Communications Networks*. CRC Press, 2013.
[40] A. Chiuso, F. Fagnani, L. Schenato, and S. Zampieri, "Gossip algorithms for distributed ranking," in *Proceedings of American Control Conference*, June 2011, pp. 5468–5473.
[41] D. Xue, A. Gusrialdi, and S. Hirche, "A distributed strategy for near-optimal network topology design," in *The 21th International Symposium on Mathematical Theory of Networks and Systems*, 2014, pp. 7–14.
[42] N. Manitara and C. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *European Control Conference*, July 2013, pp. 760–765.
[43] A. Milanese, J. Sun, and T. Nishikawa, "Approximating spectral impact of structural perturbations in large networks," *Physical Review E*, vol. 81, no. 4, p. 046112, 2010.
[44] C. Li, H. Wang, and P. Mieghem, *NETWORKING 2012: 11th International IFIP TC 6 Networking Conference, Prague, Czech Republic, May 21-25, 2012, Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Degree and Principal Eigenvectors in Complex Networks, pp. 149–160.
[45] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

Azwirman Gusrialdi (S'08-M'12) was a post-doctoral researcher at the Department of Electrical and Computer Engineering, University of Central Florida (UCF), Orlando. He received both the B.Eng and M.Eng degrees in Mechanical & Control Engineering from Tokyo Institute of Technology, Japan, in 2006 and 2008 respectively, and Dr.-Ing. degree in Control Engineering from Technische Universität München, Germany, in 2012. His research interests include design of resilient network and systems, cooperative control and distributed optimization for networked cyber-physical systems.

Zhihua Qu (M'90-SM'93-F'09) received the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in June 1990. Since then, he has been with the University of Central Florida (UCF), Orlando. Currently, he is the SAIC Endowed Professor in College of Engineering and Computer Science, a Professor and the Chair of Electrical and Computer Engineering, and the Director of FEEDER Center (one of DoE-funded national centers on distributed technologies and smart grid). His areas of expertise are nonlinear systems and control, resilient and cooperative control, with applications to energy and power systems.



Sandra Hirche holds the TUM Liesel Beckmann Distinguished Professorship and heads the Chair of Information-oriented Control in the Department of Electrical and Computer Engineering at Technical University of Munich, Germany (since 2013). She received the diploma engineer degree in Aeronautical and Aerospace Engineering in 2002 from the Technical University Berlin, Germany, and the Doctor of Engineering degree in Electrical and Computer Engineering in 2005 from Technical University of Munich, Germany. From 2005-2007 she has been a PostDoc fellow of the Japanese Society for the Promotion of Science at the Fujita Laboratory at Tokyo Institute of Technology, Japan. Prior to her present appointment she has been an associate professor at Technical University of Munich. Her main research interests include networked dynamical systems, cooperative and distributed control, and event-triggered control with applications in human-in-the-loop systems, robotics, and infrastructure systems. She has published more than 150 papers in international journals, books, and refereed conferences. Dr. Hirche has served or is serving on the Editorial Boards of the IEEE Transactions on Control Systems Technology and the IEEE Transactions on Haptics. She has received multiple awards such as the Rohde & Schwarz Award for her Ph.D. thesis in 2005, the IFAC World Congress Best Poster Award in 2005 and together with students Best Paper Awards of IEEE Worldhaptics and IFAC Conference of Manoeuvring and Control of Marine Craft in 2009.