Collaborative Network Monitoring by Means of Laplacian Spectrum Estimation and Average Consensus

Thi-Minh-Dung Tran, Alain Kibangou

▶ To cite this version:

Thi-Minh-Dung Tran, Alain Kibangou. Collaborative Network Monitoring by Means of Laplacian Spectrum Estimation and Average Consensus. International Journal of Control, Automation and Systems, 2019, 17 (7), pp.1826-1837. 10.1007/s12555-018-0638-0 . hal-02166871

HAL Id: hal-02166871

https://hal.science/hal-02166871

Submitted on 24 Oct 2019

Collaborative Network Monitoring by means of Laplacian Spectrum Estimation and Average Consensus

Thi Minh Dung Tran[*]and Alain Y. Kibangou[†]

Abstract

This paper concerns collaborative monitoring of the robustness of networks partitioned into sub-networks. We consider the critical threshold of a network and the effective graph resistance (Kirchhoff index) of a subgraph characterizing the interconnection of sub-networks, that are partitioned from the given network as robustness metric. In which, the critical threshold depends only on the two first moments of the degree distribution while the Kirchhoff index can be computed with Laplacian eigenvalues. Therefore, we show how to estimate jointly the Laplacian eigenvalues and the two first moments of the degree distribution in a distributed way.

1 Introduction

Interacting systems can be naturally viewed as networks modelled by graphs, whose vertices represent the components of the system while edges stand for the interactions between these components. The efficiency of a network can be evaluated through its functional robustness and structural robustness. The former usually stands for robustness against noise while the latter is related to the network performance despite changes in network topology (node or edge failure) [1]. Structural robustness has been an important topic in various domains: in distribution networks (e.g. power or water distribution networks), breakdowns can prevent service to customers; in communication networks, equipment failures may disrupt the network and block users from communicating; in contact networks, removing nodes (persons) by means of vaccination can prevent epidemic propagation.

Various structural robustness measures can be found in the literature. For instance, dozens of papers have studied on network robustness by considering the

---

[*] Thi Minh Dung Tran is with the Faculty of Electrical Engineering, The University of Danang - University of Science and Technology, 54 Nguyen Luong Bang street, Danang City, Vietnam. (e-mail: ttmdung@dut.udn.vn).

[†] Alain Y. Kibangou is with the Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France. (e-mails: alain.kibangou@univ-grenoble-alpes.fr).

size of the largest connected component (i.e. the largest set of nodes such that there exists a path between any two nodes) and stated that the larger the size of this component is, the more robust the network is. The existence of this giant component is characterized by the two first moments of the degree distribution of the network [2]. The removal of several nodes can break the network into several isolated components. The more nodes we remove, the higher are the chances that we damage a network. Therefore, it is worth asking how many nodes do we have to delete to fragment a network into isolated components? This leads to the definition of a critical threshold which depends only on the two first moments of the degree distribution [3].

In [4], the concept of natural connectivity as a spectral measure of robustness was introduced. It is expressed, in mathematical form, as the average eigenvalue of the adjacency matrix of the graph representing the network topology. The Laplacian spectrum can also be used to compute robustness indices. For instance, the second smallest eigenvalue of the Laplacian spectrum, which is also known as algebraic connectivity [5], is well-known as a critical parameter that influences on the performance and robustness of dynamical systems due to its main role in the connectivity of the graph [6]. Moreover, the overall Laplacian spectrum can be used to estimate the number of spanning trees and the effective graph resistance (Kirchhoff index) [7], which constitute also some robustness indices.

Theoretically, effective resistance between two vertices in a graph is defined as the electrical resistance between these two vertices when a resistor of unit resistance is placed along every edge and a potential difference is applied between the two vertices. Thus, the Kirchhoff index of the graph is the sum of the effective resistances over all pairs of nodes. The smaller the Kirchhoff index is, the more robust network becomes. It has been pointed out that adding an edge strictly decreases the Kirchhoff index and hence increases the robustness [8].

By considering consensus networks, functional robustness (i.e. robustness to noise) was defined as the expected dispersion of the network state from consensus [9]. It was shown that a greater Kirchhoff index is associated with a greater dispersion from consensus due to noise. A smaller Kirchhoff index indicates that the expected dispersion due to noise is not significant, meaning a greater functional robustness. Thus, Kirchhoff index allows considering both functional and structural robustness issues at the same time [1].

Furthermore, in order to improve structural robustness, advanced control strategies have been proposed from the robustness analysis approach. For instance, Bevrani et al. in [10] proposed a robust frequency control strategy in an islanded Microgrid (MG) to reduce the influence from the renewable sources, load change, wind power fluctuation, etc. In this study, a H_∞ and μ- synthesis robust controls was developed to enhance the secondary frequency control performance. A robust controller via μ-synthesis was again employed in [11] to reduce the frequency deviation in the presence of significant uncertainties. In [12], the authors have utilized the H_∞ control technique to design controller that are robust to the changes of the network and system nonlinearities in power sharing task. Moreover, optimization approaches, e.g., the Hopfield fuzzy neural network and combined particle swarm optimization (PSO) with fuzzy logic [13, 14] have been applied to reduce the frequency deviation. In these references, proposed control strategies and optimization approaches showed effectiveness in reducing effect of variation or uncertainties in networks.

In this paper, rather than to design a control strategy to improve system robustness, the purpose of our work is to focus onto observing interconnection within the network. In particular, we aim to monitor the critical threshold of the large network and the collaboration between the sub-graph's super-nodes of corresponding sub-networks partitioned from the large network by a robustness index (Kirchhoff index) to see whether it can survive in case of an attack or a failure before starting the operations of a system. Precisely, we consider evaluation of the two first moments of the degree distribution of the network and the Kirchhoff index of the given sub-graph by using data propagated in the network. According to the Kirchhoff index definition, estimation of the Laplacian spectrum is the key ingredient of the method to be developed. In this way, we can protect the system, improve its performance by avoiding the interruption during the operations as well as enhance the benefits technically and economically.

In the recent literature, three main approaches have been considered for the decentralized estimation of the Laplacian spectrum: Fast Fourier transform (FFT)-based methods [15, 16], local eigenvalue decomposition of given observability-based matrices [17], and distributed factorization of the averaging matrix $\mathbf{J}_N = \frac{1}{N}\mathbf{1}\mathbf{1}^T$ [18]. In the latter paper, a convex optimization problem was formulated and solved through an ADMM-based method. This method requires to run an average consensus protocol until its convergence and then use the average value and the transient measurements of the consensus protocol to solve the convex optimization problem. In this study, we extend the result in [18] by devising a method which allows estimating the average degree and the nonzero distinct Laplacian eigenvalues in decentralized way at the same breath. Here, an ADMM algorithm is employed to perform the factorization of the averaging matrix and to compute the average degree of the network concurrently. Then, set of non-zero Laplacian eigenvalues is obtained through the Laplacian eigenvalues retrieving algorithm. Consequently, a Brand-and-Bound linear programming (LP) is deployed to obtain the multiplicities of the gained eigenvalues to deduce the whole Laplacian spectrum. The Kirchhoff index is then estimated by a function of obtained Laplacian spectrum.

To simplify, we restrict our study to networks performing noise-free in the assumption that the global network topology (Laplacian Matrix) is unknown in a-priori. A dominant contribution of this paper is the possibility of implementing this monitoring scheme in a distributed manner.

The remainder of this paper is organized as follows: in Section 2, we state the problem under study. In Section 3, we recall the Laplacian-based robustness metrics and show how they can be linked with a polynomial allowing to factorize an averaging matrix using the Laplacian matrix. In Section 4, the proposed method for solving simultaneously the average consensus problem and the estimation of the Laplacian eigenvalues is described. The performance of the proposed method is evaluated in Section 5 by means of simulation results before concluding the paper.

2 Problem setting

We consider a large network $G^*(V^*, E^*)$, which is organized into M sub-networks S_ℓ, $\ell = 1, 2, \cdots, M$, [19]. We define $\mathcal{N}_i^* = \{j \in V^* : (i,j) \in E^*\}$ and its cardinality $|\mathcal{N}_i^*|$ as the set of neighbors of node i and its degree in G^* respectively. Each sub-network is monitored by a super-node i which knows the number N_ℓ of agents in the sub-network and the associated average degree x_ℓ. Node $i \in S_\ell$ is a super-node if it has at least one neighbor in a different subset, i.e. $\exists \ell^* \neq \ell$ s.t. $\mathcal{N}_i \bigcap S_{\ell^*} \neq \emptyset$. We consider that two sub-networks are connected if there exist edges linking at least two agents of these sub-networks. If two sub-networks are connected then their super-nodes are linked as illustrated in Fig. 1. The two first moments of the degree distribution are therefore given by

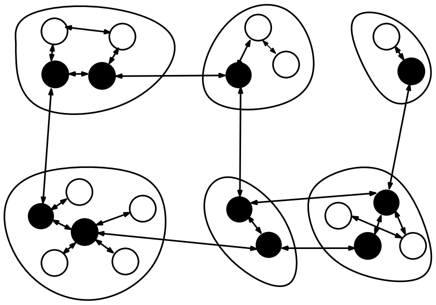

Figure 1: Network partitioned in 6 subsets. Super-nodes are depicted in black.

$$< k >= \frac{1}{|V^*|}\sum_{i=1}^{|V^*|} |\mathcal{N}_i^*|, \quad < k^2 >= \frac{1}{|V^*|}\sum_{i=1}^{|V^*|} |\mathcal{N}_i^*|^2. \tag{1}$$

The critical threshold f_c is defined as the fraction of nodes to be removed such that the network is completely disintegrated. Independently of the degree distribution, it is given by

$$f_c = 1 - \frac{1}{\frac{<k^2>}{<k>} - 1}. \tag{2}$$

A high value of f_c means that a higher fraction of nodes have to be removed to disconnect the giant component of the network. If $f_c = 1$, meaning that we have to remove all of its nodes to fragment this network.

Let $G = (\mathcal{V}, \mathcal{E})$ be the undirected graph representing a network with $N = |\mathcal{V}|$ super-nodes, which are black nodes in Fig. 1. We denote by $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$ the set of neighbors of node i in $G = (\mathcal{V}, \mathcal{E})$. The cardinality of this set, $|\mathcal{N}_i|$, defines the degree of the node.

Assumption 1 *: The subgraph, generated by super-nodes, $G = (\mathcal{V}, \mathcal{E})$, with $E \subset E^*$ and V the set of super-nodes, is connected.*$\blacksquare$

G captures the interactions between the partitions of G^*. Therefore, the large network is considered to be robust if the partitions are strongly linked each other and if the critical threshold f_c is high enough. As a consequence, the aim of nodes in G is to monitor the robustness of G^* by estimating the Kirchhoff index of G and the critical threshold f_c of G^*.

We denote by d_i the cardinality of super-node i in G (its degree) and by $\mathbf{L} \in \Re^N$, the Laplacian matrix of G, which has d_i as diagonal entries while the off-diagonal entries are zero (if $(i,j) \notin E$) or -1. We denote the Laplacian spectrum by $sp(\mathbf{L}) = \{\lambda_1^{m_1}, \lambda_2^{m_2}, \ldots, \lambda_{D+1}^{m_{D+1}}\}$, where the different Laplacian eigenvalues are in increasing order $0 = \lambda_1 < \lambda_2 < \ldots < \lambda_{D+1}$ since G is connected and superscripts stand for multiplicities $m_i = m(\lambda_i)$, while $\mathbf{\Lambda} = \{\lambda_2, \ldots, \lambda_{D+1}\}$ stands for the set of the nonzero distinct Laplacian eigenvalues. Hence, $m_1 = 1, m_i \in \mathcal{Z}^+$.

These eigenvalues can be used to evaluate the network robustness. Understanding whether a network is robust can help to protect and to improve the performance of the network efficiently. By the way, it is also used to design new networks that are able to perform well when facing with failures or attacks. There is a lot of robustness measures developed in the literature. However, we are interested in the average degree and Laplacian-based measures, the Kirchhoff index (a.k.a. effective graph resistance) as [8]:

$$\mathcal{R}_{\mathbf{L}} = N \sum_{i=2}^{D+1} \frac{m_i}{\lambda_i}. \tag{3}$$

One thing to remark here is that the graph is more robust according to the smaller $\mathcal{R}_{\mathbf{L}}$.

The aim of this paper is to devise a distributed algorithm for collaborative monitoring of network robustness through estimation of the two first moments of the degree distribution and the Laplacian eigenvalues. As a result, $\mathcal{R}_{\mathbf{L}}$ will be inferred to assess the performance of the given network.

3 Kirchhoff index and Averaging matrix factorization

For each node $i \in \mathcal{V}$, let $x_i(t)$ denotes the information state of node i, which is a parameter of interest i.e. frequency, voltage, velocity, temperature, and so on at time-step t. Define $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_N(t)]^T$, where N is the number of super-nodes in the network $G = (\mathcal{V}, \mathcal{E})$. It has been shown in [20] and [21] that the average consensus value can be obtained in D steps as follows:

$$\bar{x} = \mathbf{x}(D) = \prod_{t=D}^{1} \mathbf{W}_t \mathbf{x}(0) = \frac{1}{N}\mathbf{1}\mathbf{1}^T \mathbf{x}(0) \text{ for all } \mathbf{x}(0) \in \Re^N. \tag{4}$$

where $\mathbf{W}_t = \mathbf{I}_N + \alpha_t \mathbf{L}$, where $\mathbf{I}_N$ is an $N \times N$ identity matrix and α_t being parameters to be designed. In [20], the solution was given by $\alpha_t = -\frac{1}{\lambda_{t+1}}$, λ_t being a nonzero Laplacian eigenvalue. The uniqueness has been proved in [22].

Lemma 1 *[22] Let* $\lambda_2, \cdots, \lambda_{D+1} \neq 0$ *be the* D *distinct nonzero Laplacian eigenvalues, then, up to permutation, the sequence* $\{\alpha_i\}_{i=1,\cdots,D}$*, with* $\alpha_i = \frac{1}{\lambda_{i+1}}$, $i = 1, 2, \cdots, D$*, is the unique sequence if it allows getting* $\boldsymbol{J}_N = \frac{1}{N}\mathbf{1}\mathbf{1}^T = \prod_{i=1}^{D}(\boldsymbol{I}_N - \alpha_i \boldsymbol{L}) = \sum\limits_{t=0}^{D} c_t \mathbf{L}^t = P(c_0, c_1, \cdots, c_D; \mathbf{L})$ *with*

$$c_t = \begin{cases} 1, & \textit{if } t = 0. \\ (-1)^t \sum_{i<j<\cdots<t} \alpha_i \alpha_j \cdots \alpha_t, & \textit{if } t = 1, \ldots, D-1. \\ (-1)^D \prod_{i=1}^{D} \alpha_i, & \textit{if } t = D. \end{cases} \tag{5}$$

Therefore, in order to implement the proposed method, the network and sub-networks should be known. Meaning that, the number of the components N_ℓ in a sub-network should be known by adding a learning mechanism as a configuration step. Practically, most systems where communications are involved, learning sequences are used for communication channel identification or for synchronization. In [23], the authors have proposed a method using random walks to estimate the global properties of large connected undirected graphs such as number of vertices, edges, etc.. In this paper, we do not dig into this field. Indeed, we assume that a large network $G^*(V^*, E^*)$ is already partitioned into M subsets as described in [19] and all super-nodes know its subset's agents number N_ℓ and the average degree $\bar{d}_\ell$ of its sub-network. On the other hand, we relax the the initial condition that D - number of distinct Laplacian eigenvalues should be known by considering the h-th polynomial $P(c_0, c_1, \cdots, c_h; \mathbf{L})$ where $h = N - 1$ instead.

It is now well-known that this polynomial is minimal if and only if $\alpha_i = \frac{1}{\lambda_i}$, $i = 2, 3, \cdots, D+1$ [20]. We will denote by P^* this unique polynomial with minimum degree. Moreover, any polynomial $P(.)$ such that $P(., \mathbf{L}) = \mathbf{J}_N$ is divisible by $P^*(.)$ [18]. It can be stated that the distinct Laplacian eigenvalues can be derived by solving the averaging matrix factorization presented in the next Section. One can easily show that $N|c_1| \leq \mathcal{R}_{\mathbf{L}}$ with the equality holding only if the Laplacian eigenvalues are all simple. However, Kirchhoff index is a function of the whole Laplacian spectrum. In the other words, the multiplicities must be obtained to get $\mathcal{R}_{\mathbf{L}}$ by solving the next optimization.

Proposition 1 *Consider a connected undirected graph of* N *vertices with degree sequence* $\{d_i\}$ *and Laplacian matrix* $\mathbf{L}$*, having* D *distinct nonzero eigenvalues* $\boldsymbol{\Lambda} \in \Re^D$*. Let* $\mathbf{m} \in {\mathbb{Z}^+}^{D\times 1}$ *be the vector of the corresponding multiplicities and be obtained by solving the linear integer programming (6):*

$$\begin{aligned} \min_{\mathbf{m} \in {\mathbb{Z}^+}^{D\times 1}} \quad & \boldsymbol{\Lambda}^T \boldsymbol{m}, \\ s.t. \quad & \mathbf{A}\mathbf{m} = \boldsymbol{b} \qquad (\mathcal{M}) \\ & \mathbf{m} \in {\mathbb{Z}^+}^{D\times 1} \end{aligned} \tag{6}$$

where $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{\Lambda}^T \\ \mathbf{1}^T\end{bmatrix}$ *and* $\boldsymbol{b} = \begin{bmatrix}\sum_{i=1}^N d_i \\ N-1\end{bmatrix}$*. Then the effective graph resistance is given by (3).*

Proof: As stated previously, the roots of the polynomial of minimum degree such that $\mathbf{J}_N = P^*(c_0, c_1, \cdots, c_D; \mathbf{L})$ are the inverses of the D distinct Laplacian eigenvalues. In order to compute the effective graph resistance $\mathcal{R}_{\mathbf{L}}$, we need to compute the multiplicities of the eigenvalues. We know that $\sum_{i=1}^{D+1} m_i\lambda_i = \sum_{i=1}^N d_i$ [24]. Multiplicities $\mathbf{m}$ can then be obtained by solving the following optimization problem:

$$\min_{\mathbf{m}\in\mathbb{Z}^{+D\times 1}} \ \boldsymbol{\Lambda}^T\mathbf{m} - \sum_{i=1}^{N} d_i, \quad s.t. \qquad \mathbf{1}^T\mathbf{m} = N-1, \ \ \mathbf{m}\in\mathbb{Z}^{+D\times 1}.$$

Since all multiplicities are positive integers, the problem above can be rewritten strictly equivalently in linear programming form as (6) where $\mathcal{M}$ is the constraint set. ■

The problem (6) can be efficiently solved using the Branch-and-Bound method [25].

Algorithm 1 *Multiplicities estimation*

```
• Inputs: Number of agents $N$, $d_i$, $\mathbf{\Lambda}$.
• Output: $\mathbf{m}$
1. Initialization:
    (a) $\mathcal{H} = \{\}$ set of multiplicities, $F_{best} = \{\}$ function value of feasible
        solution, $V = \{\}$ set of nodes, that the program will visit.
    (b) $k \leftarrow 0$;
2. Calculate $\mathbf{m}$ using (6)
    • If the stopping criteria is satisfied, then stop. Else, calculate Upper
      Bound (UB) - the function value of the rounded-up feasible solution,
      then set $F_{best} \leftarrow UB$.
    • $\mathcal{H}_1 \leftarrow \boldsymbol{m}[0]$.
    • Order of variables in $\mathcal{H}_1$ to be branched on.
3. While $\mathcal{H}_1 \leftarrow \emptyset$, select an ordered variable $\hat{m}_j$ to be branched on and $\mathcal{H}_1 \leftarrow$
   $\mathcal{H}_1 \backslash \{\hat{m}_j\}$.
    (a) $k \leftarrow k+1$;
    (b) Create 2 new nodes for 2 new sub-problems.
    (c) $\boldsymbol{m}^1 \leftarrow argmin\, \boldsymbol{\Lambda}^T\boldsymbol{m}, \ s.t\, \boldsymbol{m} \in \boldsymbol{M}_1 \leftarrow \mathcal{M} \cup \{m_j \leq \lfloor \hat{m}_j \rfloor\}$ .
    (d) $\boldsymbol{m}^2 \leftarrow argmin\, \boldsymbol{\Lambda}^T\boldsymbol{m}, \ s.t\, \boldsymbol{m} \in \boldsymbol{M}_2 \leftarrow \mathcal{M} \cup \{m_j \geq \lceil \hat{m}_j \rceil\}$
```

- *If* $\boldsymbol{m}^1$ *(or* $\boldsymbol{m}^2$*) is a vector of integers,* $\mathcal{H} \leftarrow \boldsymbol{m}^1(or\, \boldsymbol{m}^2)$ *then stop.*
- *If there is infeasible solution, then prun the corresponding node.*
- *Compute* UB_1 *(or* UB_2*).*

(e) *If* $UB_1(UB_2) < UB_2(UB_1)$*, then pick* $\boldsymbol{m}^1$ *(or* $\boldsymbol{m}^2$*) is the next node to create 2 new sub-problems and* $V \leftarrow V \cup \boldsymbol{m}^2(\boldsymbol{m}^1)$*, and* $\mathcal{M} \leftarrow \boldsymbol{M}_1(\boldsymbol{M}_2)$*.*

4. Return step 3a.

From this, we have obtained the whole Laplacian spectrum $sp(\mathbf{L})$, we can calculate the robustness index, and predict the efficiency of the given network. In what follows, we show how to estimate the coefficients of the polynomial $P(.;\mathbf{L})$ in a distributed way, without the full knowledge of the network. The problem is now equivalent to distributed estimation of Laplacian eigenvalues.

4 Distributed estimation of Laplacian eigenvalues and computation of the average degree

Given an initial condition $\mathbf{x}(0)$, it is well-known that with the protocol

$$\mathbf{x}(t+1) = (\mathbf{I} - \alpha\mathbf{L})\mathbf{x}(t), \tag{7}$$

where $\mathbf{x}(t) = (x_1(t), x_2(t), ..., x_N(t))^T$ and α is appropriately chosen step-size [26], $\mathbf{x}(t)$ converges to $\bar{x}\mathbf{1}$, $\bar{x} = \frac{1}{N}\sum\limits_{i=1}^{N} x_i(0)$. In what follows, we show how transient measurements can be used to find the polynomial $P(.;\mathbf{L})$.

Since D is not known, we look for a polynomial $P(c_0, c_1, \cdots, c_h; \mathbf{L})$, with large enough degree, such that $\mathbf{J}_N = P(c_0, c_1, \cdots, c_h; \mathbf{L})$. We get $\bar{x}\mathbf{1} = \mathbf{J}_N\mathbf{x}(0) = \sum\limits_{t=0}^{h} c_t\mathbf{q}(t) = \mathbf{Qc}$, with $\mathbf{Q} = (\mathbf{q}(0)\ \mathbf{q}(1)\ \cdots\ \mathbf{q}(h)) \in \Re^{N\times(h+1)}$ and $\mathbf{q}(t) = \mathbf{L}^t\mathbf{x}(0)$. Note that $\mathbf{q}(t)$ can be constructed recursively as follows:

$$\mathbf{q}(t) = (-\alpha)^{-t}\mathbf{x}(t) - \sum_{k=0}^{t-1} \begin{pmatrix} t \\ k \end{pmatrix} (-\alpha)^{k-t}\mathbf{q}(k), \tag{8}$$

where $\begin{pmatrix} t \\ k \end{pmatrix}$ stands for the binomial coefficient.

Therefore, the vector $\mathbf{c}$ of the polynomial coefficients $\{c_t\}$ can be obtained by solving the linear system of equations

$$\mathbf{Qc} = \bar{x}\mathbf{1}. \tag{9}$$

Lemma 2 *Consider a connected undirected graph* $G = (\mathcal{V}, \mathcal{E})$ *with* $D+1$ *distinct Laplacian eigenvalues, an initial condition* $\mathbf{x}(0)$*, its average value* $\bar{x}$*, and* $\mathbf{Q} \in \Re^{N\times(h+1)}$ *built by using (8) with the transient measurements of (7). Assuming that* $\mathbf{Q}$ *has no all zero column, Equation (9) has at least one solution if* $h \geq D$*.*

Proof: Assuming that $\mathbf{Q}$ has no all zero column guarantees that (9) is well posed. In order to study the existence of solutions, we study the rank of the augmented matrix $\begin{pmatrix} \mathbf{Q} & \bar{x}\mathbf{1} \end{pmatrix}$. We know that the average value $\bar{x}\mathbf{1}$ results in the linear combination of at least the $D+1$ first columns of $\mathbf{Q}$ (see [18]). If $h \geq D$, augmenting $\mathbf{Q}$ with $\bar{x}\mathbf{1}$ does not increase the rank. Since $rank(\mathbf{Q}) \leq h+1$, the system of equations has infinitely many solutions. And, a unique solution is obtained for $h+1 = rank(\mathbf{Q})$. ■

As stated in [18], the polynomials whose coefficients vector $\mathbf{c}$ is solution of (9) are divisible by $P^*(.)$. Therefore, from such a solution, a leave-one-out procedure can be devised to find the roots of $P^*(.)$, which are the inverse of the Laplacian eigenvalues. Such a procedure will be described in Section 4.1

Solving (9) requires the knowledge of the whole matrix $\mathbf{Q}$ and the average value $\bar{x}$. In addition, to set properly the degree of the polynomial an upper-bound of D must be know. In the sequel, we will set $h = N-1$, since $1 \leq D \leq N-1$, and solve the problem in a distributed way, where each super-node has accessed to a submatrix $\mathbf{Q}_i \in \Re^{(s_i+1)\times N}$ corresponding to a selection of rows of $\mathbf{Q}$. We will assume that $\mathbf{Q}_i$ is full row rank[1]. The idea is to estimate the polynomial coefficients vector $\mathbf{c}$ in a distributed way. We adopt the following formulation:

$$\min_{\mathbf{c}_i} \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \|\mathbf{c}_i - \mathbf{c}_j\|^2 \quad s.t. \quad \mathbf{Q}_i\mathbf{c}_i = \bar{x}\mathbf{1},\ i = 1, \cdots, N, \tag{10}$$

where $\mathbf{c}_i$ stands for the local version of $\mathbf{c}$. This problem can be efficiently solved if $\bar{x}$ is known [18]. In what follows, we assume that the N first transient measurements of (7) are used to compute both the coefficients $\mathbf{c}$ of the factorization of the averaging matrix and the consensus value $\bar{x}$.

Assumption 2 *The super-nodes in a subset S_ℓ know the fraction ϵ_ℓ of nodes in the set and the two first moments of the degree distribution of the set, i.e the average degree $\bar{d}_\ell$ and the second order moment $\bar{d^2_\ell}$.*

Each super-node $i \in S_\ell$ initializes (7) with $x_i(0) = \frac{\epsilon_\ell \bar{d}_\ell}{M_\ell}$, where M_ℓ stands for the number of super-nodes in S_ℓ.

Assumption 3 *The scalars $\frac{\epsilon_\ell \bar{d}_\ell}{M_\ell}$, $\ell = 1, 2, \cdots, M$, are not all equal.*

With these assumptions, $\mathbf{x}(0)$ is not proportional to $\mathbf{1}$ in order to guarantee that $\mathbf{Q}$ has no all zero column. It can be checked by running a max and a min consensus which converge in finite-time. If the max and the min values are different then the assumption is fulfilled. One can note that the consensus value will be the average degree divided by M. With these assumptions, we propose to solve the following problem:

$$\begin{aligned} \min_{\bar{x}_i, \mathbf{c}_i, i=1,\cdots,N} \quad & \frac{\beta}{2}\sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \|\mathbf{c}_i - \mathbf{c}_j\|^2 + \frac{1-\beta}{2}\sum_{i=1}^{N}(\bar{x}_i - x_i(0))^2, \\ \text{s.t.} \quad & \mathbf{Q}_i\mathbf{c}_i = \bar{x}_i\mathbf{1}, \end{aligned} \tag{11}$$

---

[1] Each super-node i has a corresponding $(d_i+1)\times N$ matrix. It can select appropriately the maximum number $s_i \leq d_i + 1$ of rows to get a full row rank matrix $\mathbf{Q}_i$

where β is an adjustable scalar parameter ($0 \leq \beta \leq 1$) and $\bar{x}_i$ is a local estimate of the average value. ADMM is employed to solve this convex optimization problem due to its easy-to-implement aspect and also its well-studied convergence properties in [27, 28, 29, 30].

By introducing the auxiliary variables $\{\mathbf{z}_{ij}, \mu_{ij}\}$, the problem (11) is now reformulated as follows:

$$\begin{aligned}
\min_{\bar{x}_i, \mathbf{c}_i, i=1,\cdots,N} \quad & \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\|\mathbf{c}_i-\mathbf{c}_j\|^2 + \frac{1-\beta}{2}\sum_{i=1}^{N}(\bar{x}_i - x_i(0))^2, \\
\text{s.t} \quad & \bar{x}_i = \mu_{ij}, \quad i=1,\dots,N, \quad j\in\mathcal{N}_i. \\
& \mathbf{c}_i = \mathbf{z}_{ij}. \\
& \mathbf{z}_{ij} = sign(t) z_{ij}^{t}, t=0,\dots,h=N-1 \\
& \mathbf{c}_i \in C_i(\bar{x}_i).
\end{aligned}$$

where $sign(t) = \begin{cases} 1 & \text{if } t \text{ is even,} \\ -1 & \text{if } t \text{ is odd} \end{cases}$. and $C_i(\bar{x}_i)$ denotes by the constrained set of node i: $C_i(\bar{x}_i) = \{\mathbf{c} \in \Re^N |\ \mathbf{Q}_i\mathbf{c} = \bar{x}_i\mathbf{1}\}$. With this scheme, the numerical tractability of the combination of the two convex optimization problems can be overcome, [31, 32]. Now, with some dual variables $\boldsymbol{\upsilon}$ and $\mathbf{y}$, the augmented Lagrangian is defined as:

$$L_{\rho_1\rho_2}(\bar{\mathbf{x}}, \mathbf{c}, \boldsymbol{\mu}, \mathbf{z}, \boldsymbol{\upsilon}, \mathbf{y}) = L_{\rho_1}(\mathbf{c}, \mathbf{z}, \mathbf{y}) + L_{\rho_2}(\bar{\mathbf{x}}, \boldsymbol{\mu}, \boldsymbol{\upsilon}).$$

where:

$$\begin{aligned}
L_{\rho_1}(\mathbf{c}, \mathbf{z}, \mathbf{y}) = {} & \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\|\mathbf{c}_i-\mathbf{c}_j\|^2 + \sum_{j\in N_i}\frac{\rho_1}{2}\|\mathbf{c}_i - \mathbf{z}_{ij}\|^2 \\
& + \sum_{j\in N_i}\mathbf{y}_{ij}^T(\mathbf{c}_i - \mathbf{z}_{ij}), \\
L_{\rho_2}(\bar{\mathbf{x}}, \boldsymbol{\mu}, \boldsymbol{\upsilon}) = {} & \frac{1-\beta}{2}\sum_{i=1}^{N}(\bar{x}_i - x_i(0))^2 + \sum_{j\in\mathcal{N}_i}\frac{\rho_2}{2}(\bar{x}_i - \mu_{ij})^2 \\
& + \sum_{j\in\mathcal{N}_i}\upsilon_{ij}(\bar{x}_i - \mu_{ij}),
\end{aligned}$$

where ρ_1, ρ_2 are penalty terms assumed to be constant.

The iterative solution is described below:

- The average consensus value $\bar{x}_i$ minimization:

$$\bar{x}_i[k+1] = \arg\min\ L_{\rho_2}(\bar{x}_i[k], \mu_{ij}[k], \upsilon_{ij}[k]). \tag{12}$$

- The polynomial coefficients $\mathbf{c}_i$ minimization:

$$\hat{\mathbf{c}}_i = \text{argmin } L_{\rho_1}(\mathbf{c}_i[k], \mathbf{z}_{ij}[k], \mathbf{y}_{ij}[k]), \tag{13}$$

$$\mathbf{c}_i[k+1] = \Omega_{C_i(\bar{x}_i[k+1])}[\hat{\mathbf{c}}_i] \tag{14}$$

  where $\Omega_{C_i(\bar{x}_i[k+1])}[\cdot]$ stands for the projection onto the constraints set $C_i(\bar{x}_i[k+1])$ of the vector in argument.

- The auxiliary variables μ_{ij} minimization with the constraint $\mu_{ji} = \mu_{ij}$:

$$\mu_{ij}[k+1] = \text{argmin } L_{\rho_2}(\bar{x}_i[k+1], \mu_{ij}[k], \upsilon_{ij}[k]). \tag{15}$$

- The auxiliary variables $\mathbf{z}_{ij}$ minimization:

$$\mathbf{z}_{ij}[k+1] = \text{argmin } L_{\rho_1}(\mathbf{c}_i[k+1], \mathbf{z}_{ij}[k], \mathbf{y}_{ij}[k]). \tag{16}$$

- Lagrange multipliers update:

$$\upsilon_{ij}[k+1] = \upsilon_{ij}[k] + \rho_2(\bar{x}_i[k+1] - \mu_{ij}[k+1]). \tag{17}$$

$$\mathbf{y}_{ij}[k+1] = \mathbf{y}_{ij}[k] + \rho_1(\mathbf{c}_i[k+1] - \mathbf{z}_{ij}[k+1]). \tag{18}$$

Solving the sub-optimization problems (12) and (13), we respectively get:

$$\begin{aligned} \bar{x}_i[k+1] &= \frac{(1-\beta)x_i(0) + \rho_2 \sum_{j \in N_i} \mu_{ij}[k] - \sum_{j \in N_i} \upsilon_{ij}[k]}{1 - \beta + \rho_2 d_i}, \\ \hat{\mathbf{c}}_i &= \frac{\beta \sum_{j \in N_i} \mathbf{c}_j + \rho_1 \sum_{j \in N_i} \mathbf{z}_{ij} - \sum_{j \in N_i} \mathbf{y}_{ij}}{d_i(\beta + \rho_1)}. \end{aligned} \tag{19}$$

Then, solving the following the constrained optimization problem to obtain the estimated coefficients $\mathbf{c}_i[k+1] = \Omega_{C_i(\bar{x}_i[k+1])}[\hat{\mathbf{c}}_i[k+1]]$. Since the constraint is linear, we can apply the following equation to find out a feasible $\mathbf{c}_i[k+1]$:

$$\mathbf{c}_i[k+1] = \bar{x}_i[k+1]\tilde{\mathbf{Q}}_i\mathbf{1} + (\mathbf{I}_N - \tilde{\mathbf{Q}}_i\mathbf{Q}_i)\hat{\mathbf{c}}_i[k+1], \tag{20}$$

with $\tilde{\mathbf{Q}}_i = \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}$. Next, solving the sub-optimization problems (15) and (16), in the same way, yields

$$\mu_{ij}[k+1] = \frac{\bar{x}_i[k+1] + \bar{x}_j[k+1]}{2} + \frac{\upsilon_{ij}[k] + \upsilon_{ji}[k]}{2\rho_2}, \tag{21}$$

$$\mathbf{z}_{ij}[k+1] = \mathbf{c}_i[k+1] + \frac{1}{\rho_1}\mathbf{y}_{ij}[k], \text{ respectively.} \tag{22}$$

Then, we project $\mathbf{z}_{ij}[k+1]$ on the sign constraint as:

$$z_{ij}^t[k+1] = \begin{cases} \max(0, z_{ij}^t[k+1]) & \text{if } mod(t,2) = 0. \\ \min(0, z_{ij}^t[k+1]) & \text{if } mod(t,2) \neq 0. \end{cases} \tag{23}$$

The details of the derivation of update equations (19), (20), and (21) are given in Appendix A

Since the constraint set $C_i(\bar{x}_i)$ relies on the consensus value x_i, hence the parameter β can be chosen so that the optimization procedure can initially give more weight to the average consensus problem and then to the factorization one. Then, β is defined as

$$\beta(k) = \frac{1 - e^{-\gamma k}}{1 + e^{-\gamma k}} \quad (0 < \gamma < 1) \tag{24}$$

to keep it in the range $[0, 1]$ and to increase the importance of the factorization progressively while decreasing that of the average consensus.

The corresponding ADMM algorithm is then described as follows:

Algorithm 2 *ADMM-based joint estimate of average degree and polynomial coefficients of the factorization of the Laplacian matrix*

- *Inputs: Number of super-nodes N, transient measurements of the average consensus protocol $x_i(p)$, $i = 1, 2, \cdots, N$, $p = 0, 1, \cdots, N-1$.*
- *Outputs: scaled average degree $\bar{x}_i$ and polynomial coefficients vectors $\mathbf{c}_i$.*

1. *Initialization:*
    - *Penalty parameters ρ_1, ρ_2, and γ; vectors of random initial values for $\mathbf{c}_i[0] \in \Re^{N \times 1}$, $i = 1, \ldots, N$.*
    - *Each agent i forms a full row rank matrix $\mathbf{Q}_i$ with the available transient measurements and compute $\tilde{\mathbf{Q}}_i = \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}$, $\tilde{\mathbf{Q}}_i\mathbf{1}$ and $\mathbf{I}_N - \tilde{\mathbf{Q}}_i\mathbf{Q}_i$.*
    - *$k \leftarrow 0$;*
2. *Update Process:*
    - *(a) Calculate $\beta[k]$ using (24).*
    - *(b) Each agent i sends a message including its local value $\bar{x}_i[k]$, and local coefficients vector $\mathbf{c}_i[k]$ to its neighbors.*
      Then, each agent i carries out the update process.
    - *(c) Update equations:*
        - *Calculate $\bar{x}_i[k+1]$ by (19)*
        - *Calculate $\mathbf{c}_i[k+1]$ by (20).*
        - *Calculate $\mu_{ij}[k+1]$ by (21).*
        - *Calculate $\boldsymbol{z}_{ij}[k+1]$ by (22), (23).*
        - *Update the Lagrange multipliers $\upsilon_{ij}, \boldsymbol{y}_{ij}[k+1]$ using (17), (18).*
        - *$k \leftarrow k+1$;*
    - *(d) Return to 2a or stop the iterations if a stopping criterion is reached.*

This ADMM method is guaranteed to converge for all tuning parameters under mild condition [28]. The convergence analysis for distributed quadratic programming has been studied deeply in [27, 33] and results therein can be applied to our method. ADMM usually has a very good convergence speed when its parameters are appropriately chosen. There exist some works dealing with penalty parameter selection for accelerating the convergence rate of ADMM [27, 34].

4.1 Laplacian eigenvalues estimation from the averaging matrix factorization.

Let $P_i(c_0, c_1, \cdots, c_{N-1}; \cdot)$ be the polynomial built with the coefficients vector $\mathbf{c}_i$ obtained with Algorithm 2. By computing the inverses of the roots of the $P_i(c_0, c_1, \cdots, c_{N-1}; \cdot)$, that is called set $\mathcal{A}$, we obtain a set $\mathbf{\Lambda}$ of values which includes the D nonzero Laplacian eigenvalues. Our objective is then to find the smallest subset of roots of $P_i^*(c_0, c_1, \cdots, c_{N-1}; \cdot)$ such that the factorization of the averaging matrix is still valid. Recall that the factorization is valid if $\mathbf{J}_N = P_i^*(c_0, c_1, \cdots, c_{N-1}; \mathbf{L})$ meaning that $\bar{x}_i = \sum_{t=0}^{N-1} c_t \mathbf{q}_i(t)$. We state the following proposition:

Proposition 2 *Let $\mathcal{A}$ be the set of roots $\{\alpha_n\}$ of the N^{th} order polynomial with coefficients vector $\mathbf{c}$. α_n is repeated m_n times in $\mathcal{A}$. We denote by $\mathbf{Q}_{i,N-m_n}$ the sub-matrix of the $N-m_n$ columns of $\mathbf{Q}_i$ and by $\tilde{\mathbf{c}}_{\alpha_i}$ the coefficients vector of the $(N-m_n)$-th degree polynomial with roots in $\mathcal{A} \backslash \{\alpha_n\}$. The estimated Laplacian eigenvalues at super-node i are given by:*

$$\Lambda_i = \left\{ \frac{1}{\alpha_i} | \mathbf{Q}_{i,N-m_n} \tilde{\mathbf{c}}_{\alpha_n} = \bar{x}\mathbf{1}, \quad \alpha_n \in \mathcal{A} \right\}. \tag{25}$$

Proof: The proposed process is a leave-one-out testing process. Indeed, if one inverse of the Laplacian eigenvalue is left out from $\mathcal{A}$, the factorization process is no longer valid, i.e. $\mathbf{Q}_{i,N-m_n} \tilde{\mathbf{c}}_{\alpha_n} \neq \bar{x}\mathbf{1}$. ■.

In [35], the authors have proposed a leave-one-out method to retrieve the non-zero Laplacian eigenvalues.

4.2 Estimation of the critical threshold

With the algorithms described above, we get the Laplacian eigenvalues, with their multiplicities, and the average degree. We can therefore compute the effective graph resistance. In order to compute the critical threshold we need to compute the second order moment of the degree distribution, or a scaled one as done for the average degree. For this purpose the finite-time average consensus protocol [20], initialized with $\frac{\epsilon_\ell \bar{d}_\ell^2}{M_l}$, $\ell = 1, 2, \cdots, M$, can be used in order to get the second order moment of the degree distribution in a number of steps equal to the number of nonzero distinct Laplacian eigenvalues. The overall process for collaborative network monitoring includes five main steps as summarized

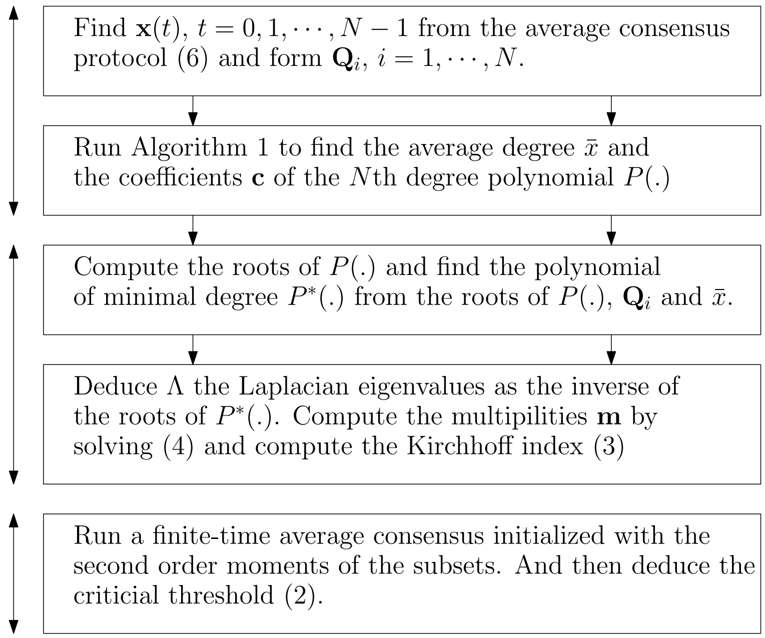
Find $\mathbf{x}(t)$, $t = 0, 1, \cdots, N-1$ from the average consensus protocol (6) and form $\mathbf{Q}_i$, $i = 1, \cdots, N$.
Run Algorithm 1 to find the average degree $\bar{x}$ and the coefficients $\mathbf{c}$ of the Nth degree polynomial $P(.)$
Compute the roots of $P(.)$ and find the polynomial of minimal degree $P^*(.)$ from the roots of $P(.)$, $\mathbf{Q}_i$ and $\bar{x}$.
Deduce Λ the Laplacian eigenvalues as the inverse of the roots of $P^*(.)$. Compute the multipilities $\mathbf{m}$ by solving (4) and compute the Kirchhoff index (3)
Run a finite-time average consensus initialized with the second order moments of the subsets. And then deduce the criticial threshold (2).

Figure 2: Functional architecture of the proposed collaborative network monitoring scheme.

in Fig. 2. The two first steps need the collaboration between the super-nodes while the two subsequent steps are purely local using consensus values of the average degree and of the polynomial giving rise to the factorization of the average matrix. The last one needs also collaboration to compute the second order moment of the degree distribution.

5 Simulation Results

In this section, we consider a large network $G^*(V^*, E^*)$ partitioned into 6 disjoint sub-networks. Here, the network can be social network, power system network, molecular network, etc.. Each sub-network has only one super-node. These super-nodes interact to each other by a new graph $G(\mathcal{V}, \mathcal{E})$, showed in Fig. 3. The parameter of each super-node can be various to estimate the Kirchhoff index. However, with the purpose of the critical threshold estimation , it is chosen to be each average degree of each sub-network, $\bar{d}_\ell$. It can be easily executed by a finite-time average consensus protocol in [20].

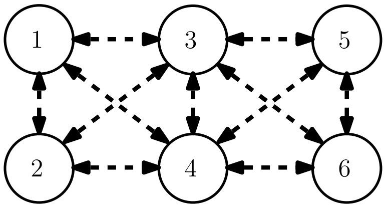

Figure 3: A new network constituted by 6 super-nodes

The parameters of each super-node are given as in Table 1.

Table 1: Parameters of given super-nodes

Super-node	1	2	3	4	5	6
fraction of nodes ϵ_ℓ	1/12	1/12	1/6	1/12	1/6	1/6
average degree $\bar{d}_\ell$	10.08	9.12	3.6	2	3.18	2.76

Let define the initial information state at time $t = 0$: $x_i(0) = \frac{\epsilon_\ell \bar{d}_\ell}{M_l}$, $\ell = 1, 2, \cdots, 6$ and $M_l = 1$, hence $\mathbf{x}[0] = \{0.84, 0.76, 0.6, 0.1667, 0.53, 0.46\}$.

5.1 Original topology

One thing to note is that, in [18], the authors have carried out the comparison between ADMM-based method with two other methods to infer that ADMM-based method is the fastest method. Therefore, in this section, we just focus on the performance of the proposed procedure and the choice of γ to reach the better solution.

The set of nonzero distinct Laplacian eigenvalues is $\mathbf{\Lambda} = \{2, 4, 6\}$ and we will compare with the output of the Algorithm 2 to validate the Algorithms.

According to proposed Algorithm 2, the super-nodes have to achieve agreement on the values of the polynomial coefficients c_t and that of the average consensus value $\bar{x}$, which is also the first order moment of the degree distribution, simultaneously.

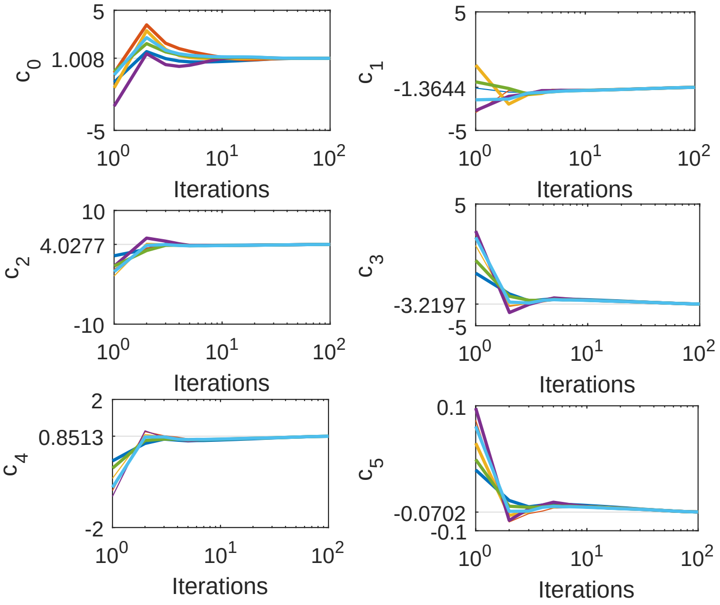

Figure 4: Nodes trajectories for the estimation of the coefficients c_t.

Fig. 4 and Fig. 5 depict the trajectories of the nodes for the estimation of the coefficients $c_{i,t}, t = 0, \ldots, N-1$, and that of the consensus value $\bar{x}$. Moreover, the average consensus value is obtained much more faster than the common values of the coefficients c_t.

Fig. 6 describes the performance in terms of the mean square error between the estimated values and the actual values for the polynomial coefficients $\mathbf{c}_i$ and the average degree $\bar{x}_i$ as:

$$MSE_c = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\mathbf{c}_i - \tilde{\mathbf{c}}\|^2}{\|\tilde{\mathbf{c}}\|^2}, \quad \text{and} \quad MSE_{\bar{x}} = \frac{1}{N} \sum_{i=1}^{N} \frac{(\bar{x}_i - \tilde{x})^2}{\tilde{x}^2},$$

where $\tilde{x}, \tilde{\mathbf{c}}$ stand for the actual average consensus value and actual polynomial coefficients respectively.

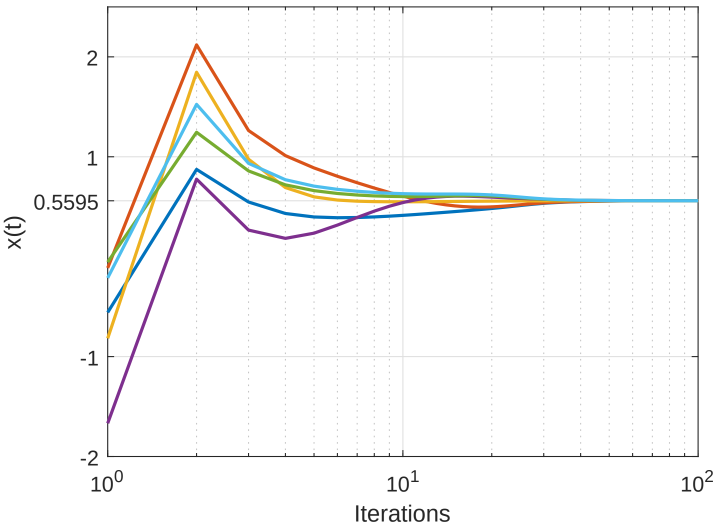

Figure 5: Nodes trajectories converging to the average of the initial condition.

Fig. 6 indicates the performance in terms of MSE_c, MSE_x with respect to the variation of $\beta(k)$. As can be seen from this Figure that when $\beta(k)$ is near to 0, the convergence speed of the average consensus problem is shown to converge faster. However, when $\beta(k)$ tends to 1, the performance in terms of MSE_c becomes faster. Table 2 depicts the results obtained for three values of γ. The choice of this parameter affects mainly the convergence speed of the algorithm. The Kirchhoff index is obtained after deploying Proposition 2 and Algorithm 1. Moreover, with the same choice of penalty parameter $\rho = 0.1$ and threshold (stopping criterion), all of the solutions of the polynomial coefficients c_t give the same assessment of the network robustness. Regarding to the speed of the convergence, $\gamma = 0.1$ seems to be the good choice in this example. We can see that, with $\gamma = 0.1$, the procedure is implemented with the fastest rate.

After achieving the Laplacian spectrum $sp(\mathbf{L})$, we can calculate the critical threshold f_c of the graph by applying a finite-time average consensus protocol again to define the second order moment of the degree distribution (1).

$$\begin{aligned} < k^2 > = & \quad \frac{1}{6}\sum_{i=1}^{6}\frac{\epsilon_\ell \bar{d}_\ell^2}{M_l} = 3.47 \\ f_c = & \quad 1 - \frac{1}{\frac{3.47}{0.5595} - 1} = 0.81 \end{aligned}$$

One thing to remark here is that a high value of f_c means that a higher fraction of nodes have to be removed to disconnect the giant component of the network. If $f_c = 1$, meaning that we have to remove all of its nodes to fragment

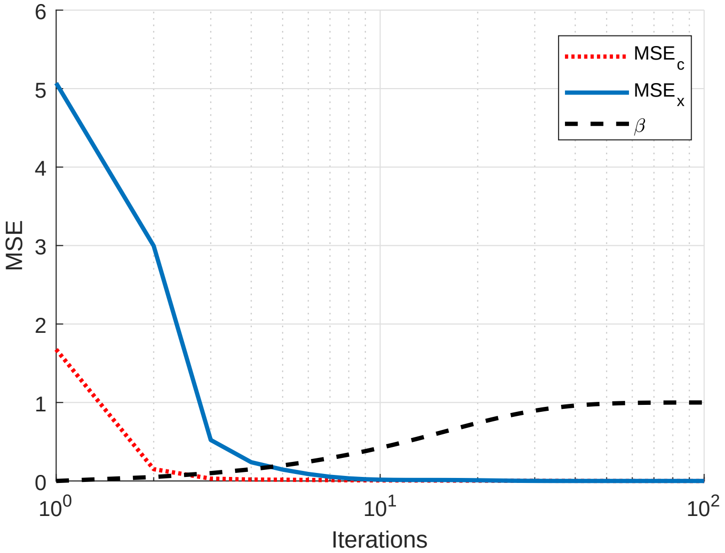

Figure 6: Mean square error between the estimated values and actual values with respect to $\mathbf{c}_i$ (MSE_c) and $\bar{x}_i$ (MSE_x).

Table 2: The achievement of the proposed algorithms with respect to the variation of γ

γ	0.01	0.06	0.1	0.5
$\mathbf{\Lambda}$	$\{2,4,6\}$			
c_t	1 -0.8386 5.0498 -4.4667 1.2162 -0.1015	1.0003 -1.3564 4.3846 -3.5514 0.9421 -0.0777	1.0008 -1.3644 4.0277 -3.2197 0.8513 -0.0702	1.0609 -0.9959 2.4466 -2.0079 0.5405 -0.0450
Iterations	1104	215	143	235
$\mathbf{m}$	$\{1,2,2\}$			
$\mathcal{R}_{\mathbf{L}}$	8			
f_c	0.81			

the large network.

With this topology, the network $G^*(V^*, E^*)$ is robust since the sub-graph $G(\mathcal{V}, \mathcal{E})$ is so-called strongly connected as well as the f_c is high enough. However, the f_c obviously depends only on the two first moments of the degree distribution, then if the topology of the sub-graph $G(\mathcal{V}, \mathcal{E})$ is changed, then the robustness of the large network have to be assessed again. Since, $M_\ell = 1$, the we can neglect the influence of the modification of the sub-graph's topology on the f_c. Now, let see the change of the robustness index in unexpected cases in the same context.

5.2 A Vulnerable topology

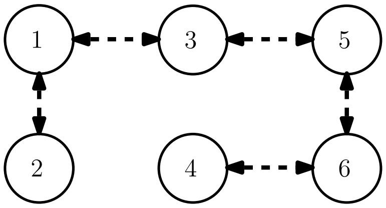

Figure 7: A tree-topology of 6-super-node network

Fig. 7 shows the case that just one arbitrary communication link is removed, then the connection of the large network is breakdown into independently subnetworks.

Theoretically, the Laplacian spectrum of this kind of topology is $\{0, 0.2679, 1, 2, 3, 3.7321\}$. Now, let see how the proposed procedure works via Table below.

Table 3 is obtained after executing the proposed procedure.

Table 3: The achievement of the proposed procedure in the sense of a-tree topology

γ	0.1
$\mathbf{\Lambda}$	$\{0.2679, 1, 2, 3, 3.7320\}$
c_t	$\{0.9999; -5.8278; 9.3245; -5.9943; 1.6651; -0.1665\}$
$\mathbf{m}$	$\{1, 1, 1, 1, 1\}$
$\mathcal{R}_{\mathbf{L}}$	35

The performance of the proposed Algorithms is described by Fig. 8 and Fig. 9. As can be seen that if the robustness index is smaller than 35, then the connection of the graph is still acceptable to ensure the performance of the system. We can see it in the next case.

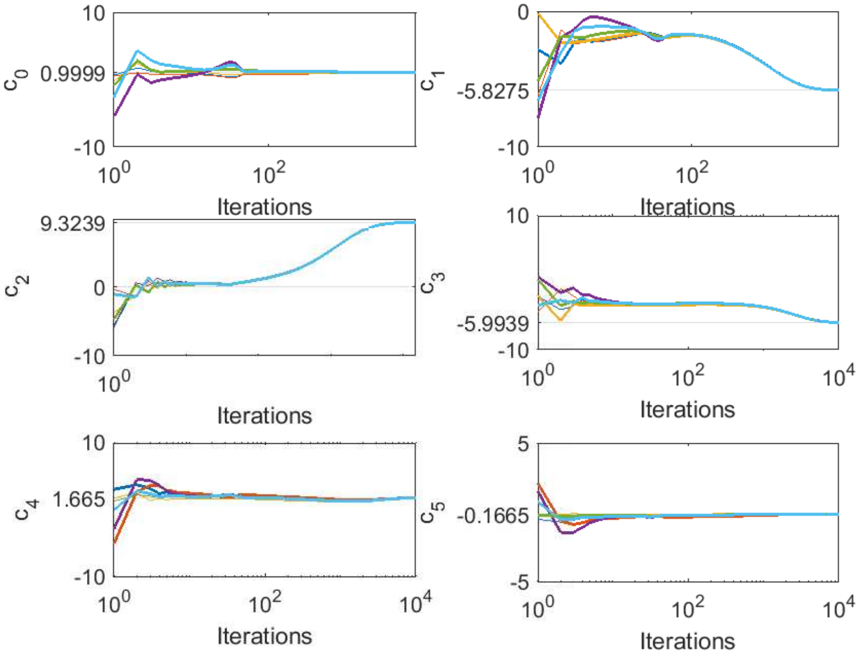

Figure 8: Nodes trajectories for the estimation of the coefficients c_t.

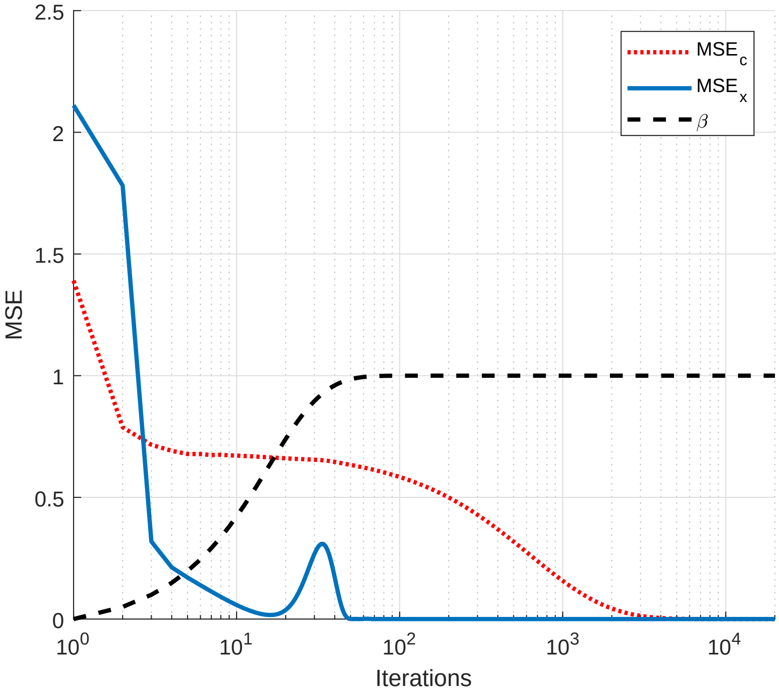

Figure 9: Nodes trajectories converging to the average of the initial condition.

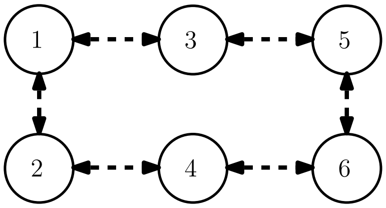

Figure 10: Network topology losts 5 links

5.3 Lost-five-link topology

Run the procedure and calculate the robustness index. Table 4 illustrates the result of the monitoring procedure. Theoretically, the eigenvalue of this topology is $\{0, 1, 1, 3, 3, 4\}$.

Table 4: The achievement of the proposed procedure in sense of a-circle topology

γ	0.1
$\mathbf{\Lambda}$	$\{1, 3, 4\}$
c_t	$\{1, -1.1519, 1.7715, -2.6268, 1.1561, -0.1490\}$
$\mathbf{m}$	$\{2, 2, 1\}$
$\mathcal{R}_{\mathbf{L}}$	17.5

And, we can see from the Table 4 that, the new robustness index is still in the safety zone ($\mathcal{R}_{\mathbf{L}}$¡35), that mean all nodes are still connected. The closer to 35 $\mathcal{R}_{\mathbf{L}}$ is, the less robust the large network is. However, if this topology is continuously lost one more link, then, it becomes a vulnerable topology. Then, it is necessarily to be considered to be improved.

6 Conclusion

In this paper, we have proposed a collaborative scheme for monitoring a large network which is already organized into sub-networks. Schematically, the process is divided into five main steps. The first two steps require the collaboration between super-nodes of corresponding sub-networks. Precisely, an ADMM-based algorithm is to perform the factorization of the averaging matrix and to compute the average degree of the network concurrently. In more detail, the factorization of the averaging matrix results on the estimation of a polynomial whose set of roots includes the inverses of Laplacian eigenvalues. Then a linear programming allows estimating the multiplicities of the Laplacian eigenvalues which are used for the computation of the Kirchhoff index. Through estimating the second order moment of the degree distribution, the critical threshold can be inferred. The proposed approach can easily be extended when one desires to estimate also the degree distribution of the overall network. In the future works, we intend to consider imperfect data exchanges for a more realistic scenario.

A Derivation of update equations

A.1 Update of the average value (19)

Setting to zero the derivative of $\mathbf{L}$ with respect to $\bar{x}_i$:

$$\frac{\partial \mathbf{L}_{\rho_2}}{\partial \bar{x}_i} = (1-\theta)(\bar{x}_i - x_i(0)) + \sum_{j \in N_i} \upsilon_{ij} + \sum_{j \in N_i} \rho_2(\bar{x}_i - \mu_{ij}) = 0$$

$$
\begin{aligned}
\Leftrightarrow \quad & (1-\theta+\rho_2 d_i)\bar{x}_i = (1-\theta)x_i(0) + \rho_2 \sum_{j\in N_i} \mu_{ij} - \sum_{j\in N_i} \upsilon_{ij} \\
\Leftrightarrow \quad & \bar{x}_i[k+1] = \frac{(1-\theta)x_i(0) + \rho_2 \sum_{j\in N_i} \mu_{ij}[k] - \sum_{j\in N_i} \upsilon_{ij}[k]}{1-\theta+\rho_2 d_i}
\end{aligned}
$$

A.2 Update of the polynomial coefficients (20)

Setting to zero the derivative of $\mathbf{L}_{\rho_1}$ with respect to $\mathbf{c}_i$:

$$
\frac{\partial \mathbf{L}_{\rho_1}}{\partial \mathbf{c}_i} = \theta \sum_{j\in N_i} (\mathbf{c}_i - \mathbf{c}_j) + \sum_{j\in N_i} y_{ij} + \sum_{j\in N_i} \rho_1(\mathbf{c}_i - \mathbf{z}_{ij}) = 0
$$

$$
\Leftrightarrow \hat{\mathbf{c}}_i = \frac{\theta \sum_{j\in N_i} \mathbf{c}_j + \rho_1 \sum_{j\in N_i} \mathbf{z}_{ij} - \sum_{j\in N_i} \mathbf{y}_{ij}}{d_i(\theta+\rho_1)}.
$$

In order to compute the projection,we know that:

$$
\begin{aligned}
\Omega_{C_i}[\hat{\mathbf{c}}_i] = & \underset{\mathbf{c}_i\in\Re^N, i=1,\dots,N}{\arg\min} \ \tfrac{1}{2}\|\mathbf{c}_i - \hat{\mathbf{c}}_i\|^2 \\
\text{s.t} \qquad & \mathbf{Q}_i \mathbf{c}_i = \bar{x}_i \mathbf{1}, \ \mathbf{Q}_i \in \Re^{(d_i+1)\times N}.
\end{aligned}
\tag{26}
$$

The Lagrangian is defined as follows:

$$
L(\mathbf{c}_i, \mathbf{y}) = \frac{1}{2}\|\mathbf{c}_i - \hat{\mathbf{c}}_i\|^2 + \mathbf{y}^T(\mathbf{Q}_i\mathbf{c}_i - \bar{x}_i\mathbf{1}).
$$

Following the KKT (Karush-Kuhn-Tucker) condition, we have to solve:

$$
\nabla_{\mathbf{c}_i} L(\mathbf{c}_i, \mathbf{y}) = 0. \tag{27}
$$

$$
\nabla_{\mathbf{y}} L(\mathbf{c}_i, \mathbf{y}) = 0. \tag{28}
$$

Let solve (27) first:

$$
\mathbf{c}_i - \hat{\mathbf{c}}_i + \mathbf{Q}_i^T \mathbf{y} = 0 \quad \Leftrightarrow \quad \mathbf{c}_i = \hat{\mathbf{c}}_i - \mathbf{Q}_i^T \mathbf{y} \tag{29}
$$

Then, we substitue $\mathbf{c}_i$ into (28):

$$
\begin{aligned}
& \mathbf{Q}_i\mathbf{c}_i - x_i\mathbf{1} = 0 \Leftrightarrow \mathbf{Q}_i(\hat{\mathbf{c}}_i - \mathbf{Q}_i^T\mathbf{y}) = \bar{x}_i\mathbf{1} \\
\Leftrightarrow \quad & \mathbf{Q}_i\hat{\mathbf{c}}_i - \mathbf{Q}_i\mathbf{Q}_i^T\mathbf{y} = \bar{x}_i\mathbf{1} \Leftrightarrow \mathbf{y} = (\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}(\mathbf{Q}_i\hat{\mathbf{c}}_i - \bar{x}_i\mathbf{1})
\end{aligned}
\tag{30}
$$

Now, we substitute (30) into (29), we get:

$$
\begin{aligned}
& \mathbf{c}_i = \hat{\mathbf{c}}_i - \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}(\mathbf{Q}_i\hat{\mathbf{c}}_i - \bar{x}_i\mathbf{1}) \\
\Leftrightarrow \quad & \mathbf{c}_i = \hat{\mathbf{c}}_i - \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}\mathbf{Q}_i\hat{\mathbf{c}}_i + \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}\bar{x}_i\mathbf{1} \\
\Leftrightarrow \quad & \mathbf{c}_i = (\mathbf{I}_N - \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}\mathbf{Q}_i)\hat{\mathbf{c}}_i + \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}\bar{x}_i\mathbf{1}.
\end{aligned}
$$

or equivalently,

$$
\mathbf{c}_i[k+1] = \tilde{\mathbf{Q}}_i(\bar{x}_i[k+1]\mathbf{1}) + (\mathbf{I}_{h+1} - \tilde{\mathbf{Q}}_i\mathbf{Q}_i)\hat{\mathbf{c}}_i[k+1],
$$

with $\tilde{\mathbf{Q}}_i = \mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{Q}_i^T)^{-1}$.

A.3 Update of the auxiliary variables (21)

Setting to zero the derivative of $\mathbf{L}_{\rho_2}$ with respect to μ_{ij}, and taking into account the constraint $\mu_{ij} = \mu_{ji}$:

$$\frac{\partial \mathbf{L}_{\rho_2}}{\partial \mu_{ij}} = -\upsilon_{ij} - \upsilon_{ji} - \rho_2(\bar{x}_i + \bar{x}_j) + 2\rho_2\mu_{ij} = 0$$

$$\Leftrightarrow \mu_{ij}[k+1] = \frac{\bar{x}_i[k+1] + \bar{x}_j[k+1]}{2} + \frac{\upsilon_{ij}[k] + \upsilon_{ji}[k]}{2\rho_2}.$$

References

[1] W. Abbas and M. Egersredt, "Robust graph Topologies for networked systems," *Proc. of the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NeCSYS)*, pp. 85–90, September 2012.

[2] R. Cohen, K. Erez and S. Havlin, "Resilience of the Internet to random breakdowns," *Phys. Rev. Lett.*, vol. 85:4626, 2000.

[3] A. Barabasi and M. Posfai, *Network science*, Cambridge University Press, Newyork, 2016.

[4] J. Wu, M. Barahona, Y. Tan and H. Deng, "Spectral Measure of Structural Robustness in Complex Networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 41, no. 6, pp. 1244-1252, November 2011.

[5] M. Friedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, pp. 298–305, 1973.

[6] W. Ellens and R. E. Kooij, "Graph measures and network robustness," *e-version, arXiv*, 2013.

[7] D.J. Klein and Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, pp. 81-95, 1993.

[8] W. Ellens, F.M. Spieksma, P. Van Mieghem, A. Jamakovic and R.E. Kooij, "Effective graph resistance," *Journal of Linear Algebra and its Applications*, vol. 435, no. 10, pp. 2491 - 2506, 2011.

[9] G. Young, L. Scardovi and N. Leonard, "Robustness of noisy consensus dynamics with directed communications," *Proc. of the American Control Conf. (ACC),Minenesota, USA*, pp. 6312–6317, 2010.

[10] H. Bevrani, M. R. Feizi and S. Ataee, "Robust Frequency Control in an Islanded Microgrid:H_∞andμ-Synthesis Approaches," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 706-717, March 2016.

[11] Y. Han, P. M. Young, A. Jain and D. Zimmerle,"Robust Control for Microgrid Frequency Deviation Reduction With Attached Storage System," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 557-565, March 2015.

[12] M. J. Hossain, H. R. Pota, M. A. Mahmud and M. Aldeen,"Robust Control for Power Sharing in Microgrids With Low-Inertia Wind and PV Generators," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 1067-1077, July 2015.

[13] H. Bevrani, F. Habibi, P. Babahajyani, M. Watanabe and Y. Mitani,"Intelligent Frequency Control in an AC Microgrid: Online PSO-Based Fuzzy Tuning Approach," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1935-1944, December 2012.

[14] W. Gu, W. Liu, Z. Wu, B. Zhao, and W. Chen,"Cooperative control to enhance the frequency stability of islanded microgrids with DFIG-SMES," *Journal of Energies*, vol. 6, no. 8, pp. 3951-3971, August 2013.

[15] M. Franceschelli, A. Gasparri, A. Giua and C. Seatzu,"Decentralized Estimation of Laplacian Eigenvalues in Multi-Agent Systems," *Journal of Automatica*, vol. 49, no. 4, pp. 1031-1036, April 2013.

[16] T. Sahai, A. Speranzon and A. Banaszuk,"Hearing the cluster of a graph: A distributed algorithm," *Journal of Automatica*, vol. 48, no. 1, pp. 15-24, January 2012.

[17] A. Y. Kibangou and C. Commault,"Decentralized Laplacian Eigenvalues Estimation and Collaborative Network Topology Identification," *Pro. of 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12),Santa Barbara, USA*, pp. 7-12, 2012.

[18] T.M.D Tran and A.Y. Kibangou,"Distributed estimation of Laplacian eigenvalues via constrained consensus optimization problems," *Systems & Control Letters*, vol. 80, pp. 56 - 62, 2015.

[19] N. Martin, P. Frasca and C. Canudas-De-Wit,"Large-scale network reduction towards scale-free structure," *IEEE Transactions on Network Science and Engineering*, vol. 14, no. 8, pp. 1-12, 2018.

[20] A.Y. Kibangou,"Graph Laplacian based matrix design for finite-time distributed average consensus," *Proc. of American Control Conference (ACC), 2012*, pp. 1901-1906, June 2012.

[21] A.Y. Kibangou,"Finite-time average consensus based protocol for distributed estimation over awgn channels," *Proc. of the IEEE Conference on Decision and Control (CDC)*, December 2011.

[22] T.M.D Tran and A.Y. Kibangou,"Consensus-based Distributed Estimation of Laplacian Eigenvalues of Undirected Graphs," *Proc. of European Control Conference (ECC)*, pp. 227-232, 2013.

[23] C. Cooper, T. Radzik and Y. Siantos,“Estimating network parameters using random walks,” *Proc. of 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, pp. 33-40, November 2012.

[24] R. Merris,“Laplacian matrices of a graph: a survey,” *Journal of Linear Algebra and its Applications*, vol. 197, pp. 143-176, 1994.

[25] J. W. Chinneck, *Practical optimization: A gentle introduction*, Systems and Computer Engineering. Carleton University. Ottawa, Canada, 2012.

[26] L. Xiao and S. Boyd,“Fast linear iterations for distributed averaging,” *Journal of Systems Control Letters*, vol. 53, pp. 65-78, 2004.

[27] D. Boley,“Local Linear Convergence of ADMM on Quadratic or Linear Programs,” *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2183-2207, 2013.

[28] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein,“Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 4, pp. 1-122, January 2011.

[29] T. Erseghe, D. Zennaro, E. Dall’Anese and L. Vangelista,“Fast Consensus by the Alternating Direction Multipliers Method,” *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5523-5537, 2011.

[30] L. Chen, D. Sun and K.C. Toh,“A note on the convergence of ADMM for linearly constrained convex optimization problems,” *Journal of Computational Optimization and Applications*, vol. 66, pp. 327-343, 2017.

[31] Y. Wei, J. Qiu and H. Lam,“A Novel Approach to Reliable Output Feedback Control of Fuzzy-Affine Systems With Time Delays and Sensor Faults,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1808-1823, December 2017.

[32] Y. Wei, J. H. Park, H. R. Karimi, Y. Tian and H. Jung, “Improved Stability and Stabilization Results for Stochastic Synchronization of Continuous-Time Semi-Markovian Jump Neural Networks With Time-Varying Delay,” *EEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2488-2501, June 2018.

[33] F. Iutzeler, P. Bianchi, Ph. Ciblat, and W. Hachem,“Explicit Convergence Rate of a Distributed Alternating Direction Method of Multipliers,” *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 892-904, April 2016.

[34] A. U. Raghunathan and S. D. Cairano,“Optimal Step-Size Selection in Alternating Direction Method of Multipliers for Convex Quadratic Programs and Model Predictive Control,” *In the 21st International Symposium on Mathematical Theory of Networks and Systems*, pp. 807-811, July 2014.

[35] T.M.D Tran and A.Y. Kibangou,"Distributed Estimation of Graph Laplacian Eigenvalues by the Alternating Direction of Multipliers Method," *Proc. of The 19th World Congress of the International Federation of Automatic Control*, 2014.