

# Distributed Control of the Laplacian Spectral Moments of a Network

Victor M. Preciado, Michael M. Zavlanos, Ali Jadbabaie, and George J. Pappas

**Abstract**—It is well-known that the eigenvalue spectrum of the Laplacian matrix of a network contains valuable information about the network structure and the behavior of many dynamical processes run on it. In this paper, we propose a *fully decentralized* algorithm that iteratively modifies the structure of a network of agents in order to control the moments of the Laplacian eigenvalue spectrum. Although the individual agents have knowledge of their *local* network structure only (i.e., myopic information), they are collectively able to aggregate this local information and decide on what links are most beneficial to be added or removed at each time step. Our approach relies on gossip algorithms to distributively compute the spectral moments of the Laplacian matrix, as well as ensure network connectivity in the presence of link deletions. We illustrate our approach in nontrivial computer simulations and show that a good final approximation of the spectral moments of the target Laplacian matrix is achieved for many cases of interest.

## I. INTRODUCTION

A wide variety of distributed systems composed by autonomous agents are able to display a remarkable level of self-organization despite the absence of a centralized coordinator [1, 2]. For example, the structure of many “self-engineered” networks, such as social and economic networks, emerges from local interactions between agents aiming to optimize their local utilities [3]. Motivated by the implications of a network’s Laplacian spectrum on its structure (i.e., number of connected components) and behavior of dynamical processes implemented on it (i.e., speed of convergence of distributed consensus algorithms), we propose a distributed model of graph evolution in which autonomous agents can modify their local neighborhood in order to control a set of moments of the network Laplacian spectrum.

The eigenvalue spectra of a network provide valuable information regarding the behavior of many dynamical processes running within the network [4]. For example, the eigenvalue spectrum of the Laplacian matrix of a graph affects the mixing speed of Markov chains [5], or the stability of synchronization of a network of nonlinear oscillators [6, 7]. Similarly, the second smallest eigenvalue of the Laplacian matrix (also called spectral gap) is broadly considered a critical parameter that influences the stability and robustness properties of dynamical systems that are implemented over

information networks [8, 9]. Optimization of the spectral gap has been studied by several authors both in a centralized [10]–[12] and decentralized context [13]. In contrast, our approach focuses on controlling the moments of the Laplacian eigenvalue spectrum. In this way, we can influence the behavior of certain dynamical processes run within the network. As we show, the benefit of controlling the spectral moments, and especially the lower order ones, lies in lower computational cost and elegant distributed implementation.

A major challenge in our approach is to efficiently control the spectral moments of a network in a *fully distributed fashion* while maintaining *network connectivity* in the presence of link deletions. Our work is related to [14], where a fully distributed algorithm is proposed to compute the full set of eigenvalues and eigenvectors of a matrix representing the network topology. However, our approach is computationally cheaper since computation of the spectral moments does not require a complete eigenvalue decomposition, but can be performed distributively by averaging local network information, such as node degrees. On the other hand, control of the network structure to the desired set of spectral moments is based on greedy actions (link additions and deletions) that are the result of distributed agreement protocols between the agents. We show that our distributed topology control algorithm is stable and converges to a network with spectral moments “close” to the desired. The performance of our algorithm is illustrated in computer simulations.

The rest of this paper is organized as follows. In Section II, we formulate the problem under consideration and review some terminology. In Section III, we derive closed-form expressions for the first four moments of the Laplacian spectrum in terms of graph properties that can be measured locally. Based on these expressions, we introduce a distributed algorithm to compute these moments. In Section IV, we propose an efficient distributed algorithm to control of the spectral moments of a network. Finally, in Section V, we illustrate our approach with several computer simulations.

## II. PRELIMINARIES & PROBLEM DEFINITION

Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  denote a graph on  $n$  nodes, where  $\mathcal{V} = \{v_1, \dots, v_n\}$  denotes the set of nodes and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of edges. If  $(v_i, v_j) \in \mathcal{E}$  whenever  $(v_j, v_i) \in \mathcal{E}$  we say that  $\mathcal{G}$  is *undirected* and call nodes  $v_i$  and  $v_j$  *adjacent* (or neighbors), which we denote by  $v_i \sim v_j$ . The set of all nodes adjacent to node  $v$  constitutes the *neighborhood* of node  $v$ , defined by  $\mathcal{N}^v = \{w \in \mathcal{V} : (v, w) \in \mathcal{E}\}$ , and the number of those neighbors is called the *degree* of node  $v$ , denoted by  $\deg v$ . We now define two quantities related with the sequence of degrees in the graph, namely, the averaged power

This work is partially supported by the ONR MURI HUNT and AFOR MURI Complex Networks programs

Victor M. Preciado, Ali Jadbabaie, and George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA {preciado, jadbabai, pappasg}@seas.upenn.edu. Michael M. Zavlanos is with the Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA michael.zavlanos@stevens.edu.

sums of the degrees  $W_k = \frac{1}{n} \sum_{v \in \mathcal{V}} (\deg v)^k$ , and the empirical degree correlation  $C_G = \frac{1}{n} \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{N}^v} \deg v \deg u$ . In this paper, we consider finite *simple* graphs, where nodes are connected by at most one edge with no self-loops.

We define a *walk* from  $v_0$  to  $v_k$  of length  $k$  to be an ordered sequence of nodes  $(v_0, v_1, \dots, v_k)$  such that  $v_i \sim v_{i+1}$  for  $i = 0, 1, \dots, k-1$ . We say that a graph  $\mathcal{G}$  is *connected* if there exists a walk between every pair of nodes. If  $v_0 = v_k$ , then the walk is closed. A closed walk with no repeated nodes (with the exception of the first and last nodes) is called a *cycle*. *Triangles* and *quadrangles* are cycles of length three and four, respectively. Let  $d(v, w)$  denote the *distance* between two nodes  $v$  and  $w$ , i.e., the minimum length of a walk from  $v$  to  $w$ . We say that  $v$  and  $w$  are  $k$ -th order neighbors if  $d(v, w) = k$  and define the  $k$ -th order neighborhood of a node  $v$  as the set of nodes within a distance  $k$  from  $v$ , i.e.,  $\mathcal{N}_k^v = \{w \in \mathcal{V} : d(v, w) \leq k\}$ . A  $k$ -th order neighborhood, induces a subgraph  $\mathcal{G}_k^v \subseteq \mathcal{G}$  with node-set  $\mathcal{N}_k^v$  and edge-set  $\mathcal{E}_k^v$  defined as the set of edges in  $\mathcal{E}$  that connect two nodes in  $\mathcal{N}_k^v$ . We say that a graphical property  $P_G$  is *locally measurable within a radius  $k$  around a node  $v$*  if  $P_G$  is exclusively a function of the neighborhood subgraph, i.e.,  $P_G = f(\mathcal{G}_k^v)$ . For example, both the degree and the number of triangles touching a node are locally measurable within a radius 1. Same is true for the number of quadrangles touching a node within a radius 2.

Graphs can be algebraically represented via the *adjacency* and *Laplacian* matrices. The *adjacency matrix* of an undirected graph  $\mathcal{G}$ , denoted by  $A = [a_{ij}]$ , is an  $n \times n$  symmetric matrix defined entry-wise as  $a_{ij} = 1$  if nodes  $v_i$  and  $v_j$  are adjacent and  $a_{ij} = 0$  otherwise.<sup>1</sup> The powers of the adjacency matrix is related to walks in a graph. In particular we have the following results [15]:

*Lemma 2.1:* The number of closed walks of length  $\alpha$  joining node  $v_i$  to itself is given by the  $i$ -th diagonal entry of the matrix  $A^\alpha$ .

*Corollary 2.2:* Let  $\mathcal{G}$  be a simple graph. Denote by  $T_i$  and  $Q_i$  the number of triangles and quadrangles touching node  $v_i$ , respectively. Then  $(A)_{ii} = 0$ ,  $(A^2)_{ii} = \deg v_i$ ,  $(A^3)_{ii} = 2T_i$  and  $(A^4)_{ii} = 2Q_i + (\deg v_i)^2 + \sum_{j \in \mathcal{N}_i} (\deg v_j - 1)$ .

Arranging the node degrees on a diagonal matrix yields the degree matrix  $D = \text{diag}(\deg v_i)$ . Then, the *Laplacian matrix*  $L$  of a graph  $\mathcal{G}$  can be defined by  $L = D - A$ . Let  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  be the eigenvalues of  $L$ , where  $\mathbf{1}$  is the vector of all ones. One can prove that  $L$  is positive semidefinite and  $\lambda_1 = 0$ . Furthermore,  $\mathcal{G}$  is connected if and only if  $\lambda_2 > 0$ , or equivalently, if  $\ker L = \text{span}\{\mathbf{1}\}$  [15]. As a result, we have the following well-known result [16]:

*Theorem 2.3:* Consider a fixed undirected graph  $\mathcal{G}$  on  $n$  nodes and let  $\theta_i(t) \in \mathbb{R}$  denote the state variable of node  $i$ . Let  $\theta(t) = [\theta_i(t)] \in \mathbb{R}^n$  be the vector of all states and assume  $\dot{\theta}(t) = -L\theta(t)$ . Then the network  $\mathcal{G}$  is connected if and only if,

$$\lim_{t \rightarrow \infty} \theta(t) = \frac{1}{n} \sum_{i=1}^n \theta_i(0) \mathbf{1} \in \text{span}\{\mathbf{1}\}. \quad (1)$$

<sup>1</sup>For simple graphs with no self-loops,  $a_{ii} = 0$  for all  $i$ .

for all initial conditions  $\theta(0) \in \mathbb{R}^n$ .

Theorem 2.3 says that the graph  $\mathcal{G}$  is connected if and only if all nodes eventually reach a consensus on their state values  $\theta_i(t)$ , for all initial conditions. Therefore, connectivity of a network  $\mathcal{G}$  can be verified almost surely by comparing the asymptotic state values (1) of all agents, for any random initialization. Note that a similar result can be obtained by application of a *finite-time* maximum (or minimum) consensus [17].

### A. Problem Definition

Consider a discrete-time sequence of graphs  $\{\mathcal{G}(s)\}_{s \geq 1}$  where  $s \in \{1, 2, \dots\}$  is the discrete time index. We denote by  $\{\lambda_i(s)\}_{s \geq 1}$  the set of Laplacian eigenvalues of  $\mathcal{G}(s)$ . We define the  $k$ -th spectral moment of the Laplacian matrix of  $\mathcal{G}(s)$  at time  $s$  as  $m_k(s) \triangleq \frac{1}{n} \sum_{i=1}^n \lambda_i^k(s)$ . The  $k$ -th centralized spectral moment of the Laplacian are  $\bar{m}_k(s) \triangleq \sum_{r=0}^k \binom{k}{r} (-1)^{k-r} m_r(s) m_1^{k-r}(s)$ . The first four centralized spectral moments of the Laplacian corresponds to the mean, variance, skewness and kurtosis of the eigenvalue spectrum and they play a central role in this paper. Define further the error function:

$$\text{CME}(\mathcal{G}(s)) = \sum_{k=0}^4 \left[ (\bar{m}_k(s))^{1/k} - (\bar{m}_k^*)^{1/k} \right]^2, \quad (2)$$

where  $\bar{m}_k^*$  denotes a given set of desired centralized moments. Since the  $k$ -th moment is the  $k$ -th power-sum of the Laplacian eigenvalues, we include the exponents  $1/k$  in the above error function with the purpose of assigning the same dimension to the summands in (2). Then, the problem addressed in this paper is:

*Problem 1:* Given an initially connected graph  $\mathcal{G}(0)$ , design a distributed algorithm that iteratively adds or deletes links from  $\mathcal{G}(s)$ , so that the connectivity of  $\mathcal{G}(s)$  is maintained for all time  $s$  and the error function  $\text{CME}(\mathcal{G}(s))$  is locally minimized for large enough  $s$ .

In what follows, we first propose a distributed algorithm to efficiently compute and update  $\text{CME}(\mathcal{G}(s))$  without any explicit eigendecomposition (Section III). Then, in Section IV, we propose a greedy algorithm where the most beneficial edge addition/deletion is determined based on a distributed agreement over all possible actions that satisfy network connectivity (Theorem 2.3). In this framework, the time variable  $s$  increases by one whenever an action is taken (i.e., an addition or deletion of a link). For simplicity, we assume that actions are taken one at a time, although this assumption can be relaxed to accommodate more complex action schemes.

## III. DISTRIBUTED COMPUTATION OF SPECTRAL MOMENTS

In what follows, we assume that the agents in the network have very limited knowledge of the network topology. In particular, we assume that every agent  $v$  only knows the topology of the second-order neighborhood subgraph around it,  $\mathcal{G}_2^v$ . (This is the case, for example, for many online social networks, where each individual can retrieve a list of his

friends' friends.) Then, computing the first four Laplacian spectral moments relies on counting the presence of certain subgraphs, such a triangles and quadrangles, in every agent's neighborhood and averaging these quantities via distributed average consensus. In particular, since the matrix trace operator is conserved under diagonalization (in general, under any similarity transformation) the first three spectral moments of the Laplacian matrix of a graph can be written as

$$m_k(L) = \frac{1}{N} \text{tr} L^k = \frac{1}{n} \sum_{p=0}^k \binom{k}{p} (-1)^p \text{tr} (A^p D^{k-p}), \quad (3)$$

for  $k \leq 3$ , where we have used the fact that the trace is preserved under cyclic permutations (i.e.,  $\text{tr} ADD = \text{tr} DAD = \text{tr} DDA$ ). We cannot use Newton's binomial expansion for the forth moment; nevertheless, we may still obtain the following closed form solution:

$$m_4(L) = \frac{1}{n} \text{tr} (D - A)^4 = \frac{1}{n} [\text{tr} D^4 - 4\text{tr} D^3 A + 4\text{tr} D^2 A^2 + 2\text{tr} DADA - 4\text{tr} DA^3 + \text{tr} A^4]. \quad (4)$$

Expanding the traces that appear in (3) and (4) we get  $\text{tr}(D^q A^p) = \sum_{i=1}^n (\text{deg } v_i)^q (A^p)_{ii}$  and  $\text{tr}(DADA) = \sum_{i=1}^n \sum_{j=1}^n (d_i a_{ij})(d_j a_{ji}) = 2C_G$ , which substituted back in equations (3) and (4) give the following expression for  $k \leq 3$

$$m_k(L) = \frac{1}{n} \sum_{i=1}^n \sum_{r=0}^k \binom{k}{r} (-1)^r (d_i)^{k-r} (A^r)_{ii}. \quad (5)$$

For  $k = 4$ , we can also simplify the Laplacian spectral moment, which now becomes

$$m_4(L) = \frac{1}{n} \sum_{i=1}^n [d_i^4 - 4d_i^3 (A)_{ii} + 4d_i^2 (A)_{ii} + 2d_i \sum_{j \in \mathcal{N}_i} d_j - 4d_i (A^3)_{ii} + (A^4)_{ii}]. \quad (6)$$

Substituting the expressions for  $(A^r)_{ii}$  from Lemma 2.1 and Corollary 2.2 in equations (5) and (6) we obtain the first four spectral moments of the Laplacian matrix  $L_G$  as follows

$$m_1(L) = W_1, \quad (7a)$$

$$m_2(L) = W_2 + W_1, \quad (7b)$$

$$m_3(L) = W_3 + 3W_2 - \frac{1}{n} \sum_{i=1}^n 2T_i, \quad (7c)$$

$$m_4(L) = W_4 + 4W_3 + W_2 + 2C_G - \frac{1}{n} \sum_{i=1}^n (8T_i d_i + 2Q_i). \quad (7d)$$

Note that the expressions for the spectral moments in equations (7) are all *averages* of locally measurable quantities (within a 2-hop neighborhood), namely, node degrees, triangles and quadrangles touching the node. Hence, we can apply consensus and use the result of Theorem 2.3 to obtain the first four moments in a distributed way.

## IV. DISTRIBUTED CONTROL OF SPECTRAL MOMENTS

### A. Possible Local Actions

The possible actions (or control variables) we consider are local link *additions* and local link *deletions*. A link addition is *local* if it connects a node with another node within its second-order neighborhood. Let  $\mathcal{N}_1^i(s)$  and  $\mathcal{N}_2^i(s)$  denote the sets of neighbors and two-hop neighbors of node  $i$  at time  $s \geq 0$ , respectively. Since any of the two nodes adjacent to a link can take an action to delete that link, we need to decide which of the two nodes has the authority to delete the link. To avoid ambiguities, we define the set of edges that node  $i$  has authority to remove as:  $\mathcal{E}_d^i(s) \triangleq \{(i, j) \in \mathcal{E}(s) \mid j \in \mathcal{N}_1^i(s), i > j\}$ . Similarly, to disambiguate between nodes adding a (still non-existing) link between them, we define the set of potential edges that node  $i$  can create as:  $\mathcal{E}_a^i(s) \triangleq \{(i, j) \in \mathcal{E}(s) \mid j \in \mathcal{N}_2^i(s) \setminus \mathcal{N}_1^i(s), i > j\}^2$ . Note that link deletions may violate network connectivity. In this case, those link deletions should be excluded from the set of allowable actions.

1) *Link Deletions*: To infer network connectivity we employ finite-time-maximum consensus which is a distributed algorithm and converges to equal values on nodes belonging to the same connected component of a graph (Theorem 2.3). According to this idea, if deletion of a link violates connectivity, both nodes adjacent to that link will lie in different connected components and will have different consensus values. In particular, consider node  $j$  that has authority to remove any of the links in the set  $\mathcal{E}_d^j(s)$ . Each one of these links needs to be checked with respect to connectivity and each connectivity verification relies on a scalar consensus update, according to Theorem 2.3. Therefore, checking all links in  $\mathcal{E}_d^j(s)$  requires  $|\mathcal{E}_d^j(s)|$  consensus updates.<sup>3</sup> We associate with every link in  $\mathcal{E}_d^j(s)$  a consensus variable, and stacking all these variables in a vector we obtain the state vector  $\mathbf{x}_{jj}(s) \in \mathbb{R}^{|\mathcal{E}_d^j(s)|}$ . Running a distributed consensus over the network, requires participation of all other nodes  $i \neq j$ . This is possible by defining the state variables  $\mathbf{x}_{ij}(s) \in \mathbb{R}^{|\mathcal{E}_d^j(s)|}$ . All vectors  $\mathbf{x}_{ij}(s)$  are initialized randomly and are updated by node  $i$  according to the following maximum consensus: *Case I*: If  $(i, j) \notin \mathcal{E}_d^i(s) \cup \mathcal{E}_d^j(s)$ , i.e., if nodes  $i$  and  $j$  are not neighbors, then

$$\mathbf{x}_{ij}(s) := \max_{k \in \mathcal{N}_1^i(s)} \{\mathbf{x}_{ij}(s), \mathbf{x}_{kj}(s)\}, \quad (8)$$

with the maximum taken elementwise on the vectors,

*Case II*: If  $(i, j) \in \mathcal{E}_d^j(s)$ , i.e., if nodes  $i$  and  $j$  are neighbors and node  $j$  has authority over link  $(i, j)$ , then

$$[\mathbf{x}_{ij}(s)]_{(i,j)} := \max_{k \in \mathcal{N}_1^i(s) \setminus \{j\}} \{[\mathbf{x}_{ij}(s)]_{(i,j)}, [\mathbf{x}_{kj}(s)]_{(i,j)}\}, \quad (9)$$

and

$$[\mathbf{x}_{ij}(s)]_{(l,j)} := \max_{k \in \mathcal{N}_1^i(s)} \{[\mathbf{x}_{ij}(s)]_{(l,j)}, [\mathbf{x}_{kj}(s)]_{(l,j)}\}, \quad (10)$$

<sup>2</sup>Note that since the indices of all nodes in the network are distinct, this definition results in a unique assignment of links to nodes.

<sup>3</sup>We define by  $|X|$  the cardinality of the set  $X$ .

---

**Algorithm 1** Connectivity verification

---

**Require:**  $\mathbf{x}_{ij} \in \mathbb{R}^{|\mathcal{E}_d^i|}$  for all  $j \in \mathcal{V}$  ;  
**Require:**  $T_{ij} = [0 \dots 1_i \dots 0]^T, \forall j \in \mathcal{V}$ ;  
1: **if**  $\exists j \in \mathcal{V}$  such that  $\min\{T_{ij}\} = 0$  **then**  
2:   Update  $\mathbf{x}_{ij}$  by (8)–(11);  
3:    $T_{ij} := \max_{k \in \mathcal{N}_i^1} \{T_{ij}, T_{kj}\}$ ;  
4: **else if**  $\min\{T_{ij}\} = 1$  for all  $j \in \mathcal{V}$  **then**  
5:   Update  $\mathcal{E}_\sigma^i$  by (12);  
6: **end if**

---

for  $l \neq i$ , where  $[\mathbf{x}_{kj}(s)]_{(l,j)} \in \mathbb{R}$  denotes the entry of  $\mathbf{x}_{kj}(s)$  corresponding to the link  $(l, j)$ ,

*Case III:* If  $(i, j) \in \mathcal{E}_d^i(s)$ , i.e., if nodes  $i$  and  $j$  are neighbors and node  $i$  has authority over link  $(i, j)$ , then

$$[\mathbf{x}_{ii}(s)]_{(i,j)} := \max_{k \in \mathcal{N}_i^1(s) \setminus \{j\}} \{[\mathbf{x}_{ii}(s)]_{(i,j)}, [\mathbf{x}_{ki}(s)]_{(i,j)}\}. \quad (11)$$

Once consensus (8)–(11) has converged, node  $i$  compares the entries  $[\mathbf{x}_{ii}(s)]_{(i,j)}$  and  $[\mathbf{x}_{ji}(s)]_{(i,j)}$  for all  $(i, j) \in \mathcal{E}_d^i(s)$ . Since, violation of connectivity due to deletion of the link  $(i, j)$  would result in nodes  $i$  and  $j$  being in different connected components of the network,  $[\mathbf{x}_{ii}(s)]_{(i,j)} = [\mathbf{x}_{ji}(s)]_{(i,j)}$  implies that the reduced network is still connected. Hence, we can define a set

$$\mathcal{E}_\sigma^i(s) \triangleq \{(i, j) \in \mathcal{E}_d^i(s) \mid [\mathbf{x}_{ii}(s)]_{(i,j)} = [\mathbf{x}_{ji}(s)]_{(i,j)}\}, \quad (12)$$

containing the *safe* links adjacent to node  $i$  that if deleted, connectivity is maintained.

2) *Connectivity Verification:* The connectivity verification of link deletions, discussed in Section IV-A.1, is illustrated in Alg. 1. Convergence of the finite-time consensus (8)–(11) is captured by a vector of tokens  $T_{ij}(s) \in \{0, 1\}^n$ , initialized as  $T_{ij}(s) := [0 \dots 1_i \dots 0]^T$  for all  $j \in \mathcal{V}$  and indicating that node  $i$  has initialized the consensus variables for link deletions for which node  $j$  is responsible. In particular, when all tokens of all nodes have been collected (line 4, Alg. 1), then consensus has converged and the set of safe link deletions  $\mathcal{E}_\sigma^i(s)$  can be computed (line 5, Alg. 1).

### B. Most Beneficial Local Action

As discussed in Problem 1, the objective of this work is to minimize the error function  $\text{CME}(\mathcal{G}(s))$ . For this we propose a greedy algorithm, which for every time  $s$  selects the action that maximizes the quantity  $\text{CME}(\mathcal{G}(s)) - \text{CME}(\mathcal{G}(s+1))$ , if such an action exists, and terminates if no such action exists. By construction, such an algorithm converges to a network that *locally* minimizes  $\text{CME}(\mathcal{G}(s))$ , while in Section V, we show that it performs well in practice too. In particular, let

$$\begin{aligned} \Delta m_1^{\pm(i,j)} &= \pm \frac{2}{n}, \\ \Delta m_2^{\pm(i,j)} &= \frac{2}{n} [1 \pm (d_i + d_j + 1)], \\ \Delta m_3^{\pm(i,j)} &= \frac{1}{n} [(3 \pm 6)(d_i + d_j) \pm 3(d_i^2 + d_j^2) + (6 \pm 2) \mp 6T_{ij}] \end{aligned}$$

denote the increments in the first three moments, where the notation  $\pm(i, j)$  indicates a link addition (+) or deletion (−) and the dependence on time  $s$  has been dropped for simplicity. (Similarly, one can obtain a complicated closed-form expression for  $\Delta m_4^{\pm(i,j)}$ , which we omit due to space limitations.) Then, agent’s  $i$  copy of the  $k$ -th spectral moment  $m_k^i(s)$  becomes

$$m_k^{\pm(i,j)}(s) = m_k^i(s) + \Delta m_k^{\pm(i,j)}(s),$$

and the associated centralized moment  $\bar{m}_k^{\pm(i,j)}(s)$  can be computed as in Section II. Then, for all possible actions discussed in Section IV-A, agent  $i$  computes the error function

$$\text{CME}_{\pm(i,j)}(s) = \sum_{k=0}^4 \left[ \left( \bar{m}_k^{\pm(i,j)}(s) \right)^{1/k} - (\bar{m}_k^*)^{1/k} \right]^2.$$

Then, the local *most beneficial action* to the target centralized moments, namely, the action that results in the maximum decrease in the error function  $\text{CME}(\cdot)$ , can be defined by<sup>4</sup>

$$\nu_i(s) \triangleq \max \left\{ \underset{j}{\text{argmin}} \left( \text{CME}_{\pm(i,j)}(s) - \text{CME}_i(s) \right) \right\},$$

where  $\text{CME}_i(s)$  denotes agent  $i$ ’s copy of  $\text{CME}(s)$ , and the largest decrease in the error associated with action  $\nu_i(s)$  becomes:

$$\text{CME}_i(s) = \text{CME}_{\pm(i,\nu_i(s))}(s)$$

if  $\min_j (\text{CME}_{\pm(i,j)}(s) - \text{CME}(s)) \leq 0$  and  $\text{CME}_i(s) = D$ , otherwise. Note that  $\text{CME}_i(s)$  is nontrivially defined only if there exists a link adjacent to node  $i$  that if added or deleted decreases the error function  $\text{CME}(\cdot)$ . Otherwise, a large value  $D > 0$  is assigned to  $\text{CME}_i(s)$  to indicate that this action is not beneficial to the final objective. We can include all information of a best local action in the vector

$$\mathbf{v}_i(s) \triangleq [i \ \nu_i(s) \ \text{CME}_i(s) \ \bar{\mathbf{m}}_{\pm(i,\nu_i(s))}(s)]^T \in \mathbb{R}^7$$

containing the local best action  $(i, \nu_i(s))$ , the associated distance to the desired moments  $\text{CME}_i(s)$ , and the vector of centralized moments  $\bar{\mathbf{m}}_{\pm(i,\nu_i(s))}(s)$  due to this action. In the following section we discuss how to compare all local actions  $\mathbf{v}_i(s)$  for all nodes  $i \in \mathcal{V}$  to obtain the one that decreases the distance to the desired moments the most.

### C. From Local to Global Action

We propose a control scheme where the desired local actions  $\mathbf{v}_i(s)$  are propagated in the network, along with vectors of tokens  $T_i(s) \in \{0, 1\}^n$ , initialized as  $T_i(s) := [0 \dots 1_i \dots 0]^T$ , indicating that node  $i$  has transmitted its desired action. During every iteration of the algorithm, every node  $i$  communicates with its neighbors and updates its vector of tokens  $T_i(s)$  (line 3, Alg. 2), as well as its desired action  $\mathbf{v}_i(s)$  with the action  $\mathbf{v}_j(s)$  corresponding to the node  $j$  that contains the smallest distance to the target moments  $[\mathbf{v}_j(s)]_3$ , i.e.,

$$j \in \underset{k \in \mathcal{N}_1 i(s)}{\text{argmin}} \{[\mathbf{v}_i(s)]_3, [\mathbf{v}_k(s)]_3\}.$$

<sup>4</sup>The max in the expression below indicates that in case of ties in the min, the highest index node wins.

---

**Algorithm 2** Globally most beneficial action
 

---

**Require:**  $\mathbf{v}_i \triangleq [i \ \nu_i \ \text{CME}_i \ \bar{\mathbf{m}}_{\pm(i,\nu_i)}]^T$ ;

**Require:**  $T_i := [0 \dots 1_i \dots 0]^T$ ;

- 1: **if**  $\min\{T_i\} = 0$  **then**
  - 2:    $\mathbf{v}_i := \mathbf{v}_j$ , with  $j = \max\{\text{argmin}_{k \in \mathcal{N}_1^i} \{[\mathbf{v}_i]_3, [\mathbf{v}_k]_3\}\}$ ;
  - 3:    $T_i := \max_{j \in \mathcal{N}_1^i} \{T_i, T_j\}$ ;
  - 4: **else if**  $\min\{T_i\} = 1$  and  $[\mathbf{v}_i]_3 < D$  **then**
  - 5:   Update  $\mathcal{N}_i$ ,  $\bar{\mathbf{m}}_i$  and  $\text{CME}_i$  according to (13)–(16);
  - 6: **else if**  $\min\{T_i\} = 1$  and  $[\mathbf{v}_i]_3 = D$  **then**
  - 7:   No beneficial action. Algorithm has converged;
  - 8: **end if**
- 

In case of ties in the distances to the targets  $[\mathbf{v}_j(s)]_3$ , i.e., if  $\text{argmin}_{k \in \mathcal{N}_1^i(s)} \{[\mathbf{v}_i(s)]_3, [\mathbf{v}_k(s)]_3\}$  contains more than one node, then the node  $j$  with the largest label is selected (line 2, Alg. 2). Note that line 2 of Alg. 2 is essentially a minimum consensus update on the entries  $[\mathbf{v}_i(s)]_3$  and will converge to a common outcome for all nodes when they have all been compared to each other, which is captured by the condition  $\min_{j=1}^n T_{ij}(s) = 1$  (lines 4 and 6, Alg. 2). When the consensus has converged, if there exists a node whose desired action decreases the distance to the target moments, i.e., if  $[\mathbf{v}_i(s)]_3 < D$  (line 4, Alg. 2), then Alg. 2 terminates with a greedy action and node  $i$  updates its set of neighbors  $\mathcal{N}_1^i(s)$  and vector of centralized moments  $\bar{\mathbf{m}}_i(s)$  (line 5, Alg. 2). If the optimal action is a link addition, i.e., if  $[\mathbf{v}_i(s)]_2 \notin \mathcal{N}_1^i(s)$ , then

$$\mathcal{N}_1^i(s+1) := \mathcal{N}_1^i(s) \cup \{[\mathbf{v}_i(s)]_2\}. \quad (13)$$

On the other hand, if the optimal action is a link deletion, i.e., if  $[\mathbf{v}_i(s)]_2 \in \mathcal{N}_1^i(s)$ , then

$$\mathcal{N}_1^i(s+1) := \mathcal{N}_1^i(s) \setminus \{[\mathbf{v}_i(s)]_2\}. \quad (14)$$

In all cases, the centralized moments and error function are updated by

$$\bar{\mathbf{m}}_i(s+1) := [\mathbf{v}_i(s)]_{4:7} \quad (15)$$

and

$$\text{CME}_i(s+1) := [\mathbf{v}_i(s)]_3, \quad (16)$$

respectively, where  $[\mathbf{v}_i(s)]_{4:7} = [[\mathbf{v}_i(s)]_4 \dots [\mathbf{v}_i(s)]_7]^T$ . Finally, if all local desired actions increase the distance to the target moments, i.e., if  $[\mathbf{v}_i(s)]_3 = D$  (line 6, Alg. 2), then no action is taken and the algorithm terminates with a network topology with almost the desired spectral properties. This is because no action exists that can further decrease the distance to the target moments.

#### D. Synchronization

Synchronization of all processes in the absence of a common clock is necessary for correctness of the proposed control scheme. Here, we employ the approach proposed in [18]. Details are omitted due to space limitations.

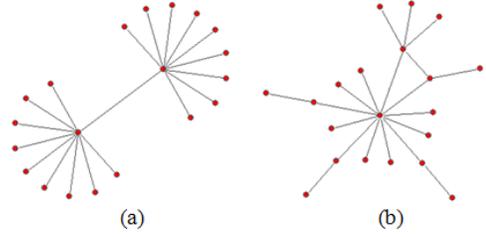


Fig. 1. Structures of the two-stars network (a) and the network returned by our algorithm (b).

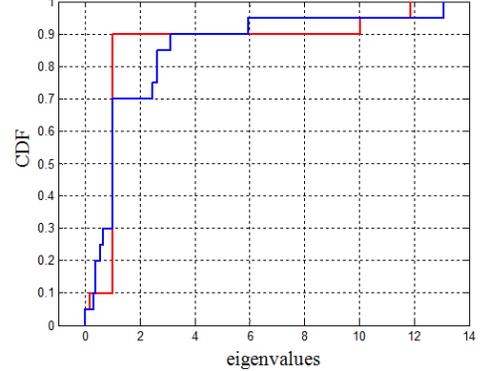


Fig. 2. Empirical cumulative distribution functions for the eigenvalues of the two-stars graph (blue) and the graph returned by our algorithm (red).

## V. NUMERICAL SIMULATIONS

*Example 1 (Two-stars network):* We consider a two-stars graph on 20 nodes (Fig. 1(a)). We initialize our algorithm with a random graph on 20 nodes and try to approximate the first four central moments of the two-stars graph. In our simulations, we observe that the error function quickly reaches a neighborhood of zero but does not reach zero exactly. Instead of obtaining the two-stars graph as a final result, our algorithm returns the network shown in Fig. 1(b). Although both graphs are different, the eigenvalue spectra of the desired two-star network and the network in Fig. 1(b) are still very similar, as shown in Fig. 2.

*Example 2 (Chain vs. ring networks):* The objective of this example is to illustrate how two structurally similar target graphs, such as a chain and a ring, may affect the performance of our algorithm. In particular, if we run our algorithm to control the moments of an initially random graph towards the moments of a chain graph, we observe how the error function converges exactly to zero in finite time. Furthermore, the final result of our algorithm is an exact reconstruction of the chain graph. Nevertheless, when transforming the target graph from a chain graph into a ring graph (by adding a single link), an exact reconstruction is very difficult. In Fig. 3, we illustrate some graphs returned by our algorithm for different initial conditions when we control the set of moments toward the moments of a ring network on 20 nodes. Observe that, although the algorithm tends to create long cycles and the majority of nodes have degree two, it fails to recreate the exact structure of the ring graph

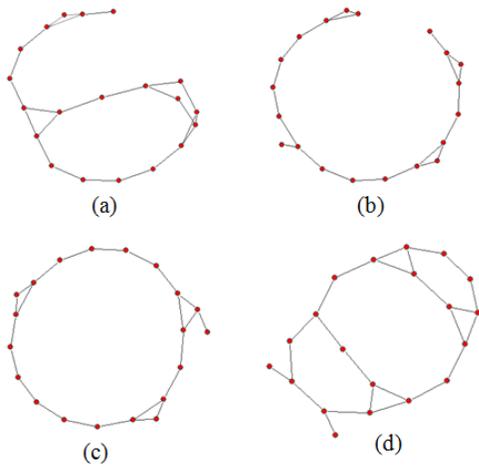


Fig. 3. Networks returned by our algorithm when trying to match the first four central moments of a ring on 20 nodes.

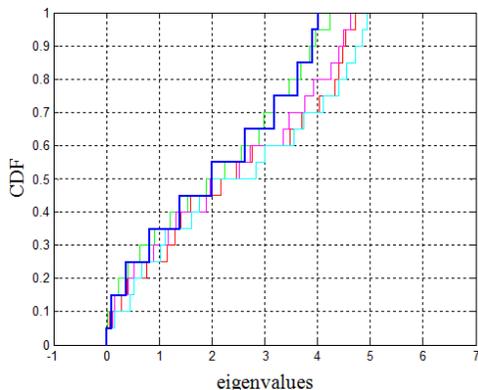


Fig. 4. Empirical cumulative distribution of eigenvalues for the ring graph with 20 nodes (blue plot) and the graphs in Fig. 3(a) (red), Fig. 3(b) (green), Fig. 3(c) (magenta) and Fig. 3(d) (cyan).

due to the local nature of the algorithm (as in Example 1). However, although the structure of the resulting networks is not exactly the desired ring graph, their spectral properties are remarkably close to those of a ring. In Fig. 4, we illustrate the empirical cumulative distribution functions of the eigenvalues of the ring graph (blue plot), versus the four empirical cumulative distribution functions corresponding to the graphs in Fig. 3.

## VI. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have described a fully decentralized algorithm that iteratively modifies the structure of a network of agents with the objective of controlling the spectral moments of the Laplacian matrix of the network. Although we assume that each agent has access to local information regarding the graph structure, we show that the group is able to collectively aggregate their local information to take a global optimal decision. This decision corresponds to the most beneficial link addition/deletion in order to minimize an error function that involves the first four Laplacian spectral moments of the network. The aggregation of the local information is achieved

via gossip algorithms, which are also used to ensure network connectivity throughout the evolution of the network.

Future work involves identifying sets of spectral moments that are reachable by our control algorithm. (We say that a sequence of spectral moments is reachable if there exists a graph whose moments match the sequence of moments.) Furthermore, we observed that fitting a set of low-order moments does not guarantee a good fit of the complete distribution of eigenvalues. In fact, there are important spectral parameters, such as the algebraic connectivity, that are not captured by a small set of spectral moments. Nevertheless, we observed in numerical simulations that fitting the first four moments of the eigenvalue spectrum often achieves a good reconstruction of the complete spectrum. Hence, a natural question is to describe the set of graphs most of whose spectral information is contained in a relatively small set of low-order moments.

## REFERENCES

- [1] N. Wiener, *The Mathematics of Self-Organising Systems. Recent Developments in Information and Decision Processes*, Macmillan, 1962.
- [2] H. Haken, *Synergetics: An Introduction*, 3<sup>rd</sup> Edition, Springer-Verlag, 1983.
- [3] M.O. Jackson, *Social and Economic Networks*, Princeton University Press, 2008.
- [4] V.M. Preciado, *Spectral Analysis for Stochastic Models of Large-Scale Complex Dynamical Networks*, Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., MIT, Cambridge, MA, 2008.
- [5] D.J. Aldous, *Some Inequalities for Reversible Markov Chains*, J. London Math. Soc. (2)25, pp. 564-576, 1982.
- [6] L.M. Pecora, and T.L. Carroll, "Master Stability Functions for Synchronized Coupled Systems," *Physics Review Letters*, vol. 80(10), pp. 2109-2112, 1998.
- [7] V.M. Preciado, and G.C. Verghese, "Synchronization in Generalized Erdős-Rényi Networks of Nonlinear Oscillators," *Proc. of the 44th IEEE Conference on Decision and Control*, pp. 4628-4633, 2005.
- [8] A. Fax, and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465-1476, 2004.
- [9] R. Olfati-Saber, and R. M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520-1533, 2004.
- [10] A. Ghosh, and S. Boyd, "Growing Well-Connected Graphs," *Proc. of the 45th IEEE Conference on Decision and Control*, pp. 6605-6611, 2006.
- [11] R. Grone, R. Merris, and V.S. Sunder, "The Laplacian Spectrum of a Graph," *SIAM Journal Matrix Analysis and Applications*, vol. 11, pp. 218-238, 1990.
- [12] Y. Kim, and M. Mesbahi, "On Maximizing the Second-Smallest Eigenvalue of a State Dependent Graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, pp. 116-120, 2006.
- [13] M.C. DeGennaro, and A. Jadbabaie, "Decentralized Control of Connectivity for Multi-Agent Systems," *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, CA, Dec. 2006, pp. 3628-3633.
- [14] D. Kempe, and F. McSherry, "A Decentralized Algorithm for Spectral Analysis," *Journal of Computer and System Science*, vol. 74, pp. 70-83, 2008.
- [15] N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, 2<sup>nd</sup> Edition, 1993.
- [16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Transactions on Automatic Control*, vol. 48(6), pp. 988-1001, 2003.
- [17] J. Cortes, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44(3), pp. 726-737, 2008.
- [18] M. M. Zavlanos, and G. J. Pappas, "Distributed Connectivity Control of Mobile Networks," *IEEE Transactions on Robotics*, vol. 24(6), pp. 1416-1428, 2008.
- [19] D.J. Watts, and S. Strogatz, "Collective Dynamics of Small World Networks," *Nature*, vol. 393, pp. 440-42, 1998.