Automatica 93 (2018) 415–421

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Speeding up finite-time consensus via minimal polynomial of a weighted graph — A numerical approach[☆]

Zheming Wang, Chong Jin Ong [*]

Department of Mechanical Engineering, National University of Singapore, 117576, Singapore

ARTICLE INFO

Article history:
Received 5 March 2017
Received in revised form 21 August 2017
Accepted 3 January 2018

Keywords:
Laplacian matrix
Rank minimization
Minimal polynomial
Consensus algorithm

ABSTRACT

This work proposes an approach to speed up finite-time consensus algorithm using the weights of a weighted Laplacian matrix. It is motivated by the need to reach consensus among states of a multi-agent system in a distributed control/optimization setting. The approach is an iterative procedure that finds a low-order minimal polynomial that is consistent with the topology of the underlying graph. In general, the lowest-order minimal polynomial achievable for a network system is an open research problem. This work proposes a numerical approach that searches for the lowest order minimal polynomial via a rank minimization problem using a two-step approach: the first being an optimization problem involving the nuclear norm and the second a correction step. Convergence of the algorithm is shown and effectiveness of the approach is demonstrated via several examples.

1. Introduction

Achieving consensus of states is a well-known important feature for networked system, see for example Olfati-Saber and Murray (2004) and Ren and Beard (2007). Many distributed control/optimization problems over a network require a consensus algorithm as a key component. The most common consensus algorithm is the dynamical system defined by the Laplacian matrix for continuous time system and the Perron matrix for discrete-time system. Past works in the general direction of speeding up convergence of these algorithms exist. For example, the work of Xiao and Boyd (2004) proposes a semi-definite programming approach to minimize the algebraic connectivity over the family of symmetric matrices that are consistent with the topology of the network. Their approach, however, results in asymptotic convergence towards the consensus value and is most suitable for large networks. More recent works focus on finite-time convergence consensus algorithm (Hendrickx, Jungers, Olshevsky, & Vankeerberghen, 2014; Hendrickx, Shi, & Johansson, 2015; Sundaram & Hadjicostis, 2007; Wang & Xiao, 2010; Yuan, Stan, Shi, Barahona, & Goncalves, 2013; Yuan, Stan, Shi, & Goncalves, 2009) which is generally preferred for small to moderate sized networks. One important area in finite-time convergence literature is the determination of the asymptotic value of a consensus network using a finite number of state measurement. Typically, the approach adopted is based on the z-transform final-value theorem and on the finite-time convergence for individual node (Sundaram & Hadjicostis, 2007; Yuan et al., 2013, 2009) without knowledge of the full network. Other works in finite-time consensus include the design of a short sequence of stochastic matrices $A_k, \dots, A_0$ such that $z(k) = \Pi_{j=1}^{k} A_j z(0)$ reaches consensus after k steps (Hendrickx et al., 2015; Ko & Shi, 2009).

Unlike past works (Sundaram & Hadjicostis, 2007; Yuan et al., 2013, 2009) where the network is unknown, this work assumed a known network and proposes an approach to speed up finite-time convergence of consensus algorithm via the choices of the weights associated with the edges of the graph. Thus, it is similar in spirit to the work of Xiao and Boyd (2004) except that the intention is to find a low-order minimal polynomial. Ideally, the lowest-order minimal polynomial should be used. However, the lowest minimal polynomial achievable for a given graph with variable weights is an open research problem (Fallat & Hogben, 2007). They are only known for some special classes of graphs (full connected, star-shaped, strongly regular and others), van Dam and Haemers (1998) and van Dam, Koolen, and Tanaka (2014). For this reason, this paper adopts a computational approach towards finding a low-order minimal polynomial. The proposed approach achieves the lowest order minimal polynomial in many of the special classes of graphs and almost always yields minimal polynomial of order lower than those obtained from standard Perron matrices of general graphs. These are demonstrated by several numerical examples.

The choice of the weights is obtained via a rank minimization problem. In general, rank minimization is a well-known difficult

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Antonis Papachristodoulou under the direction of Editor Christos G. Cassandras.
[*] Corresponding author.
E-mail addresses: wang.zheming@u.nus.edu (Z. Wang), mpeongcj@nus.edu.sg (C.J. Ong).

https://doi.org/10.1016/j.automatica.2018.03.067
0005-1098/© 2018 Elsevier Ltd. All rights reserved.

problem (Fazel, Hindi, & Boyd, 2004; Recht, Fazel, & Parrilo, 2010). Various approaches have been proposed in the literature including the nuclear norm relaxation, bilinear projection and others. This work uses a unique two-step procedure: the first is a nuclear norm optimization problem and the second, which uses the results of the first, is a correction step based on a low rank approximation. While both steps of this two-step procedure have appeared in the literature, the use of the two in a two-step predictor–corrector procedure is novel, to the best of the authors' knowledge. Hence, the proposed rank minimization approach can be of independent interest, as well as the expression of finite-time convergence value obtained via a non z-transform mechanization.

The remainder of this paper is organized as follows. This section ends with a description of the notations used. Section 2 reviews features of the standard Laplacian and Perron matrices as well as minimal polynomial and its properties. Section 3 presents the procedure of obtaining the consensus value from the minimal polynomial and discusses, in detail, the key subalgorithm used in the overall algorithm including a convergence result. The overall algorithm is described in Section 4 and the performance of the approach is illustrated via several numerical examples in Section 5. Conclusions are given in Section 6.

The notations used in this paper are standard. Non-negative and positive integer sets are indicated by $\mathbb{Z}_0^+$ and $\mathbb{Z}^+$, respectively; whereas, $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n\times m}$ refer, respectively, to the sets of real numbers, n-dimensional real vectors and n by m real matrices. I_n is the $n \times n$ identity matrix with 1_n being the n-column vector of all ones (subscript omitted when the dimension is clear). Given a set C, $|C|$ denotes its cardinality. The transpose of matrix M and vector v are indicated by M' and v', respectively. For a square matrix Q, $Q \succ (\succeq) 0$ means Q is positive definite (semi-definite), $spec(Q)$ refers to the set of its eigenvalues, and $vec(Q)$ is the representation of elements of Q as a vector. The cones of symmetric positive semi-definite and symmetric and positive definite matrices are $\mathcal{S}_{0+}^n = \{M \in \mathbb{R}^{n\times n} | M = M', M \succeq 0\}$ and $\mathcal{S}_+^n = \{M \in \mathbb{R}^{n\times n} | M = M', M \succ 0\}$, respectively. The ℓ_p-norm of $x \in \mathbb{R}^n$ is $\|x\|_p$ for $p = 1, 2, \infty$ while $\|M\|_*, \|M\|_2, \|M\|_F$ are the nuclear, operator (induced) and Frobenius norm of matrix M. Diagonal matrix is denoted as $diag\{d_1, \ldots, d_n\}$ with diagonal elements d_i. Additional notations are introduced when required.

2. Preliminaries and problem formulation

This section begins with a review of standard consensus algorithm and sets up the notations needed for the sequel. The network of n nodes is described by an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The pair $(i, j) \in \mathcal{E}$ if i is a neighbor of j and vice versa since G is undirected. The set of neighbors of node i is $N_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}, i \neq j\}$. The standard adjacency matrix $\mathcal{A}_s$ of G is the $n \times n$ matrix whose (i, j) entry is 1 if $(i, j) \in \mathcal{E}$, and 0 otherwise.

The implementation of the proposed consensus algorithm is a discrete-time system of the form $z(k + 1) = \mathcal{P}z(k)$ where $\mathcal{P}$ is the Perron matrix. However, for computational expediency, the working algorithm uses the weighted Laplacian matrix $\mathcal{L} \in \mathcal{S}_{0+}^n$. The conversion of $\mathcal{L}$ to $\mathcal{P}$ is standard and is discussed later, together with desirable properties of $\mathcal{P}$ and $\mathcal{L}$. The properties of standard (non-weighted) $\mathcal{L}$ are first reviewed.

The standard Laplacian matrix $\mathcal{L}_s$ of a given G is

$$[\mathcal{L}_s]_{i,j} = \begin{cases} -1, & \text{if } j \in N_i; \\ |N_i|, & \text{if } i = j\,; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In this form, it is easy to verify that (i) eigenvalues of $\mathcal{L}_s$ are real and non-negative, (ii) eigenvectors corresponding to different eigenvalues are orthogonal, (iii) $\mathcal{L}_s$ has at least one eigenvalue 0 with eigenvector 1_n. Properties (i) and (ii) follow from the fact that $\mathcal{L}_s$ is symmetric and positive semi-definite while property (iii) is a result of the row sum of all rows being 0. Suppose the assumption (**A1**): G is connected is made. Then, it is easy to show that the eigenvalue of 0 is simple with eigenvector 1_n. Consequently, the consensus algorithm of $\dot{x}(t) = -\mathcal{L}_s x(t)$ converges to $\frac{1}{n}1_n(1_n'x(0))$.

Unlike (1), this work uses the weighted Laplacian

$$\mathcal{L}(W, G) = \mathcal{D}(G) - \mathcal{A}(G, W) \tag{2}$$

where $\mathcal{A}(G, W)$ is the weighted adjacency matrix with $[\mathcal{A}(G, W)]_{ij} = w_{ij}$ when $(i, j) \in \mathcal{E}$, $\mathcal{D}(G) = diag\{d_1, d_2, \ldots, d_n\}$ with $d_i := \sum_{j\in N_i} w_{ij}$ and $W := \{w_{ij} \in \mathbb{R} | (i, j) \in \mathcal{E}\}$. The intention of this work is to compute algorithmically the minimal polynomial of $\mathcal{L}(W, G)$ over variable W for a given G. However, since the minimal polynomial attainable for a given network G is a well-known difficult problem (Fallat & Hogben, 2007), the output of the algorithm can be seen as an upper bound on the order of the achievable minimal polynomials of $\mathcal{L}(W, G)$ over all W. Note that the value of w_{ij} is arbitrary including the possibility that $w_{ij} = 0$ and $w_{ij} < 0$ for $(i, j) \in \mathcal{E}$. This relaxation allows for a larger W search space but brings about the possibility of losing connectedness of $\mathcal{L}(W, G)$ even when G is connected. Additional conditions are therefore needed to preserve connectedness, as discussed in the sequel. Since G is fixed, its dependency in $\mathcal{L}(\cdot), \mathcal{D}(\cdot)$ and $\mathcal{A}(\cdot)$ is dropped for notational convenience unless required.

The desirable properties of $\mathcal{L}(W)$ are as follows:

(L1) All eigenvalues are non-negative.
(L2) 0 is a simple eigenvalue with eigenvector 1_n.
(L3) $[\mathcal{L}(W)]_{ij} = 0$ when $(i, j) \notin \mathcal{E}$.
(L4) $\mathcal{L}(W)$ has a low-order minimal polynomial.

Properties (L1) and (L2) are needed for $x(t)$ of the continuous time system $\dot{x}(t) = -\mathcal{L}x(t)$ to reach consensus while (L3) is a hard constraint imposed by the structure of G. Property (L4) determines the finite-time convergence towards consensus and is the objective of this work. With these properties, the corresponding Perron matrix is obtained from $\mathcal{P} := e^{-\epsilon\mathcal{L}}$ or $\mathcal{P} := I_n - \epsilon\mathcal{L}(W)$, with $0 < \epsilon < \frac{1}{max_i\{d_i\}}$). Then, it is easy to verify that $\mathcal{P}$ inherits from (L1)–(L4) the following properties:

(P1) All eigenvalues of $\mathcal{P}$ lie within the interval $(-1, 1]$.
(P2) 1 is a simple eigenvalue of $\mathcal{P}$ with eigenvector 1_n.
(P3) $[\mathcal{P}]_{ij} = 0$ when $(i, j) \notin \mathcal{E}$.
(P4) $\mathcal{P}$ has a low-order minimal polynomial.

The discrete-time consensus algorithm via $\mathcal{P}$ follows

$$z(k + 1) = \mathcal{P}z(k) \tag{3}$$

for discrete variable $z \in \mathbb{Z}_0^+$. From (P1), (P2) and (A1), it is easy to show, with (σ_i, ξ_i) being the ith eigenpair of $\mathcal{P}$, that $\lim_{k\to\infty} z(k) = \lim_{k\to\infty}(\sum_{i=1}^n \xi_i\xi_i'\sigma_i^k)z(0) = \frac{1}{n}1_n$. Hence, $\lim_{k\to\infty} z(k)$ reaches consensus among all its elements. The above shows that finding a $\mathcal{P}$ that possesses properties (P1)–(P4) is equivalent to finding an $\mathcal{L}(W)$ having properties (L1)–(L4). The remaining paragraphs of this section review definitions and properties of the minimal and characteristic polynomials.

Definition 1. The minimal polynomial $m_M(t)$ of a square matrix M is the monic polynomial of the lowest order such that $m_M(M) = 0$.

Several well known properties of characteristic and minimal polynomial (see for example, Friedberg, Insel, and Spence (2003) or others) are collected in the next lemma.

Lemma 1. *Given a square matrix M with minimal polynomial $m_M(t)$ and characteristic polynomial $p_M(t)$. Then (i) λ is an eigenvalue of M if and only if λ is a root of $m_M(t)$. (ii) A root of $m_M(t)$ is a root of $p_M(t)$. (iii) The distinct roots of $m_M(t)$ are equivalent to the distinct roots of $p_M(t)$. Suppose M is symmetric, then (iv) the algebraic multiplicity of every eigenvalue in M equals its geometric multiplicity and (v) each distinct root of $m_M(t)$ has a multiplicity of one.*

Since G is undirected, the desirable properties of (L4) (or equivalently (P4)) is achieved by having as few distinct roots of $m_{\mathcal{L}}(t)$ as possible, as given in properties (iv) and (v) of Lemma 1. The next result shows the connection of the minimal polynomial and the computations needed to obtain the consensus value of (3) using only $s_{\mathcal{P}}$ number of states where $s_{\mathcal{P}}$ is the order of the minimal polynomial, $m_{\mathcal{P}}(t)$. It also indicates how $m_{\mathcal{P}}(t)$ can be computed when only *spec*$(\mathcal{P})$ is known and not the full $\mathcal{P}$.

3. Main approach

Lemma 2. *Given a nth order symmetric matrix $\mathcal{P}$ with minimal polynomial $m_{\mathcal{P}}(t)$ of order $s_{\mathcal{P}}$. Any matrix polynomial $h(\mathcal{P})$ can be expressed as $h(\mathcal{P}) = r(\mathcal{P})$ where $r(t)$ is a $(s_{\mathcal{P}} - 1)$ order polynomial. The coefficients of $r(t)$ can be obtained from solution of the following equation involving the Vandermonde matrix.*

$$\begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{s_{\mathcal{P}}-1} \\ \vdots & & \ddots & & \vdots \\ 1 & \lambda_{s_{\mathcal{P}}} & \lambda_{s_{\mathcal{P}}}^2 & \cdots & \lambda_{s_{\mathcal{P}}}^{s_{\mathcal{P}}-1} \end{pmatrix} \begin{pmatrix} \pi_0 \\ \vdots \\ \pi_{s_{\mathcal{P}}-1} \end{pmatrix} = \begin{pmatrix} h(\lambda_1) \\ \vdots \\ h(\lambda_{s_{\mathcal{P}}-1}) \end{pmatrix} \tag{4}$$

where $\lambda_1, \ldots, \lambda_{s_{\mathcal{P}}}$ are the distinct eigenvalues of $\mathcal{P}$ and π_i is the coefficient of t^i in $r(t)$.

Proof. The polynomial $h(t)$ can be expressed, via long division by $m_{\mathcal{P}}(t)$, as $h(t) = \phi(t)m_{\mathcal{P}}(t) + r(t)$, where $r(t)$ is the remainder of the division and, hence, is of order $s_{\mathcal{P}} - 1$ (including the possibility that some coefficients are zero). Since the above holds for all values of t, it holds particularly when $t = \lambda_i$, the ith distinct eigenvalue of $\mathcal{P}$. This fact, together with $m_{\mathcal{P}}(t)|_{t=\lambda_i} = 0$ from property (P1) of Lemma 1, establishes each row of (4). From property (P3) of Lemma 1, there are $s_{\mathcal{P}}$ distinct eigenvalues of $\mathcal{P}$ and there are $s_{\mathcal{P}}$ unknown coefficients in $r(t)$ whose values can be obtained from solving (4). To show that (4) always admits a solution, note that the determinant of the Vandermonde matrix is $\Pi_{1\le i\le j\le s_{\mathcal{P}}-1}(\lambda_j - \lambda_i)$ and is non-zero since λ_i are distinct. The above equation also holds for its corresponding matrix polynomial or $h(\mathcal{P}) = \phi(\mathcal{P})m_{\mathcal{P}}(\mathcal{P}) + r(\mathcal{P})$. That $h(\mathcal{P}) = r(\mathcal{P})$ follows because $m_{\mathcal{P}}(\mathcal{P}) = 0$. □

Consider a given polynomial of $h(t) = \lim_{k\to\infty} t^k := t^\infty$ and that $\mathcal{P}$ satisfies properties (P1)–(P4). Lemma 2 states that there exists a polynomial $r(t)$ or order $s_{\mathcal{P}} - 1$ such that $t^\infty = \pi_{s_{\mathcal{P}}-1}t^{s_{\mathcal{P}}-1} + \cdots + \pi_1 t + \pi_0$ and the values of $\{\pi_{s_{\mathcal{P}}-1}, \ldots, \pi_0\}$ can be obtained from the $s_{\mathcal{P}}$ equations of (4). The right hand side of (4) has the properties that $h(t)|_{t=\lambda_i} = \lambda_i^\infty = 0$ for all but one eigenvalues of $\mathcal{P}$ since $|\lambda_i| < 1$ (property P1). The remaining eigenvalue of $\mathcal{P}$ is $\lambda_i = 1$ and it yields $h(t)|_{t=\lambda_i} = \lambda_i^\infty = 1$, following property (P2). Hence,

$$\mathcal{P}^\infty = \pi_{s_{\mathcal{P}}-1}\mathcal{P}^{s_{\mathcal{P}}-1} + \cdots + \pi_1\mathcal{P} + \pi_0 I_n. \tag{5}$$

Rewriting (3) as $z(k) = \mathcal{P}^k z(0)$, one gets

$$\lim_{k\to\infty} z(k) = \mathcal{P}^\infty z(0) = (\sum_{\ell=0}^{s_{\mathcal{P}}-1} \pi_\ell \mathcal{P}^\ell) z(0) = \sum_{\ell=0}^{s_{\mathcal{P}}-1} \pi_\ell z(\ell). \tag{6}$$

The application of Lemma 2 for distributed consensus algorithm is now obvious. Each agent i stores the parameters $\{\pi_{s_{\mathcal{P}}-1}, \ldots, \pi_0\}$ in its memory as well as $\{z_i(0), \ldots, z_i(s_{\mathcal{P}} - 1)\}$, obtained from $z_i(k+1) = \sum_{j\in N_i} \mathcal{P}_{ij} z_j(k)$, for $k = 0, \ldots, s_{\mathcal{P}} - 2$. At the end of $k = s_{\mathcal{P}} - 2$, agent i takes the sum of $\pi_0 z_i(0) + \cdots + \pi_{s_{\mathcal{P}}-1} z(s_{\mathcal{P}} - 1)$ which yields $\lim_{k\to\infty} z_i(k)$.

Remark 1. Clearly, the values of $\pi_0, \ldots, \pi_{s_{\mathcal{P}}-1}$ can be obtained from (4) when the spectrum of $\mathcal{L}(W)$ is known using the procedure described in the sequel. Numerically stable routine for (4) involving only $O(n^2)$ operations instead of $O(n^3)$ requirement of standard matrix inversion is also available (Björck & Pereyra, 1970).

3.1. Key subalgorithms

With the discussion above, it is clear that the matrix $\mathcal{L}$ (or $\mathcal{P}$) should be chosen such that $m_{\mathcal{L}}(t)$ is the lowest-order minimal polynomial consistent with the network. The numerical approach proposed here attempts to find the lowest order minimal polynomial by minimizing the number of distinct eigenvalues of $\mathcal{L}(W, G)$ over variable W, since the order of the minimal polynomial $m_{\mathcal{L}}(t)$ equals to the number of distinct eigenvalues of $\mathcal{L}$ from Lemma 1.

The overall scheme of the proposed algorithm is now described in loose terms for easier appreciation. The numerical algorithm is iterative and at each iteration k, two subalgorithms are invoked producing two possible choices of W. The W that results in $\mathcal{L}(W, G)$ having a lower order of minimal polynomial is then chosen as $\mathcal{L}(W_k, G)$. The two subalgorithms, OPA and OPB, are very similar in structure but serve different purposes: OPA searches for a new eigenvalue of $\mathcal{L}(W_k, G)$ with multiplicity of 2 or higher while OPB searches for additional multiplicity of eigenvalues that are already present in $\mathcal{L}(W_k, G)$. To accomplish this, two sets are needed: $\mathcal{C}_k$ containing the zero simple eigenvalue and distinct eigenvalues with multiplicities of 2 or higher in $\mathcal{L}(W_k, G)$ and $\mathcal{M}_k$ containing the multiplicities of the eigenvalues in $\mathcal{C}_k$. The set $\mathcal{C}_k$ has the property that $\lambda \in \mathcal{C}_k$ implies $\lambda \in \mathcal{C}_{k+1}$. The remaining 'free' eigenvalues in $\mathcal{L}(W_k, G)$ are then optimized again in the next iteration. An additional index function $\xi(\cdot) : \mathbb{Z}^+ \to \mathbb{Z}^+$ is needed to keep track of the cardinality of $\mathcal{C}_k$ in the sense that $\xi(k) = |\mathcal{C}_k|$. The overall scheme proceeds with decreasing order of the minimal polynomial and stops when no further repeated eigenvalue can be found. At each iteration, properties (L1)–(L3) and assumption (A1) are preserved from $\mathcal{L}(W_{k-1}, G)$. The key steps at iteration k are now discussed. For notational simplicity, $\mathcal{L}(W_k, G)$ is denoted as $\mathcal{L}_k$.

Iteration k requires the following data as input from iteration $k - 1$: the matrix $\mathcal{L}_{k-1}$; index function $\xi(k-1)$; the set $\mathcal{C}_{k-1} = \{\lambda_1, \lambda_2, \ldots, \lambda_{\xi(k-1)}\}$ with $\lambda_1 = 0$ and $\lambda_2, \ldots, \lambda_{\xi(k-1)}$ being distinct eigenvalues of $\mathcal{L}_{k-1}$; and the set of multiplicities $\mathcal{M}_{k-1} := \{m_1, m_2, \ldots, m_{\xi(k-1)}\}$ where $m_1 = 1$ and $m_i \ge 2$ is the multiplicity of λ_i in $\mathcal{C}_{k-1}$. Let the respective numbers of fixed and free eigenvalues in $\mathcal{L}_{k-1}$ be

$$q_{k-1} := \sum_{i=1}^{\xi(k-1)} m_i, \quad \bar{q}_{k-1} = n - q_{k-1}. \tag{7}$$

Iteration k starts by computing the eigen-decomposition of $\mathcal{L}_{k-1}$ in the form of $\mathcal{L}_{k-1} = Q_{k-1}\Lambda_{k-1}Q_{k-1}'$ where

$$\Lambda_{k-1} = \begin{pmatrix} D_{k-1}^c & 0 \\ 0 & D_{k-1}^o \end{pmatrix} \tag{8}$$

with $D_{k-1}^c \in \mathcal{S}_{0+}^{q_{k-1}}$ being a diagonal matrix with elements, in the same order, as the eigenvalues in $\mathcal{C}_{k-1}$ including multiplicities, and $D_{k-1}^o \in \mathcal{S}_+^{\bar{q}_{k-1}}$ being the diagonal matrix containing the remaining eigenvalues of $\mathcal{L}_{k-1}$. Correspondingly, the Q_{k-1} can be expressed as

$[Q^c_{k-1}\ Q^o_{k-1}]$ of appropriate dimensions such that

$$\mathcal{L}_{k-1} = \begin{pmatrix} Q^c_{k-1} & Q^o_{k-1} \end{pmatrix} \begin{pmatrix} D^c_{k-1} & 0 \\ 0 & D^o_{k-1} \end{pmatrix} \begin{pmatrix} (Q^c_{k-1})' \\ (Q^o_{k-1})' \end{pmatrix}. \tag{9}$$

Consider the parameterization of $\mathcal{L}$ by a symmetric matrix $M \in \mathcal{S}^{\bar{q}_{k-1}}_+$ in the form of

$$\begin{aligned} H(M) &:= \begin{pmatrix} Q^c_{k-1} & Q^o_{k-1} \end{pmatrix} \begin{pmatrix} D^c_{k-1} & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} (Q^c_{k-1})' \\ (Q^o_{k-1})' \end{pmatrix} \\ &= Q^c_{k-1}D^c_{k-1}(Q^c_{k-1})' + Q^o_{k-1}M(Q^o_{k-1})'. \end{aligned} \tag{10}$$

The structural constraints of the graph G are imposed on M via $[H(M)]_{ij} = [Q^c_{k-1}D^c_{k-1}(Q^c_{k-1})']_{ij} + [Q^o_{k-1}M(Q^o_{k-1})']_{ij} = 0$ for $(i,j) \notin \mathcal{E}$. The collection of these structural constraints can be stated as $\Phi_{k-1}vec(M) = b_{k-1}$ where $vec(M)$ is the vectorial representation of M with Φ_{k-1} and b_{k-1} being the collection of appropriate terms from $Q^c_{k-1}D^c_{k-1}(Q^c_{k-1})'$ and $Q^o_{k-1}M(Q^o_{k-1})'$, respectively. Consider the following optimization problems over variables $\lambda \in \mathbb{R}$, $M \in \mathcal{S}^{\bar{q}_{k-1}}_+$:

$$\begin{aligned} (OP) \quad & \min \quad rank(\lambda I - M) && (11a) \\ & \text{s.t. } M \succ 0 && (11b) \\ & \Phi_{k-1}vec(M) = b_{k-1}. && (11c) \end{aligned}$$

Then, the next lemma summarizes its properties.

Lemma 3. *Suppose $\mathcal{L}_{k-1}$ satisfies (L1)–(L3) and (A1). Then (i) OP has a feasible solution. (ii) $spec(H(M^*)) = spec(M) \cup spec(D^c_{k-1})$. (iii) Suppose (λ^*, M^*) is the optimizer of OP. Then $H(M^*)$ satisfies (L1)–(L3) and (A1).*

Proof. (i) Choose $\hat{M} = D_o$ where D_o is that given in (9) and $\hat{\lambda}$ be any diagonal element of D_o. Then $(\hat{\lambda}, \hat{M})$ is a feasible solution to OP since $\mathcal{L}_{k-1}$ satisfies (L1)–(L3) and (A1). (ii) The property is obvious from the expression of $H(M^*)$ of (10). (iii) From (ii), the spectrum of $H(M^*)$ are the eigenvalues in $\mathcal{C}_{k-1}$ (with the corresponding multiplicities) and those of M^*. Since $M^* \succ 0$ and all values in $\mathcal{C}_{k-1}$ are non-negative with $0 \in \mathcal{C}_{k-1}$ being simple, (L1) and (L2) hold for $H(M^*)$. The condition of (L3) is ensured by constraint (11c). Since $\mathcal{L}_{k-1}$ satisfies (A1) and $M^* \succ 0$, the second largest eigenvalue of $H(M)$ is also strictly greater than 0 which implies $H(M^*)$ satisfies (A1).

Remark 2. The description above shows that the eigen-decomposition of $\mathcal{L}_{k-1}$ is done from scratch from presentation convenience. The expression of $[Q^c_{k-1}\ Q^o_{k-1}]$ in (8) can also be obtained incrementally. To see this, suppose (λ^*, M^*) is the solution of OP, then it follows from (9) that $Q^c_k = Q^c_{k-1}$, $Q^o_k = Q^o_{k-1}J_k$ where J_k is the transformation matrix such that $M^* = J_k D^o_k J'_k$.

Clearly, minimizing $rank(\lambda I - M)$ in (11a) is equivalent to maximizing the dimension of the nullspace of $\lambda I - M$, which in turn leads to λ having the largest multiplicity. However, rank minimization is a well-known difficult numerical problem. The numerical experiment undertaken (Section 5) suggests that the nuclear norm approximation is the most reliable since it results in a convex optimization problem and is known to be the tightest pointwise convex lower bound of the rank function (Recht et al., 2010). With such a relaxation, the optimization problem over $\lambda \in \mathbb{R}$ and $M \in \mathcal{S}^{\bar{q}_{k-1}}_+$ becomes

$$\begin{aligned} OPA(\Phi_{k-1}, b_{k-1}) : & \min_{\lambda,M} \|\lambda I - M\|_* && (12a) \\ & \text{s.t. } M \succeq \epsilon_M I_{\bar{q}_{k-1}}, (11c) && (12b) \end{aligned}$$

where ϵ_M is some small positive value to prevent eigenvalues of M being too close to 0. Suppose (λ^*, M^*) is the optimizer of OPA. There are many cases where the solution of OPA provides a low value of $rank(\lambda^* I - M^*)$. However, there are also many cases where their solutions differ. In one of these cases, further progress can be made. This special case is characterized by M^* having several eigenvalues that are relatively close (known hereafter as bunch eigenvalues) to one another but are not close enough for the nullspace of $(\lambda^* I - M^*)$ to have a dimension greater than one. When this situation is detected, a correction step is invoked. Specifically, suppose $spec(M^*) = \{\mu_1, \ldots, \mu_{\bar{q}_{k-1}}\}$ and there are ℓ bunch eigenvalues with $\bar{q}_{k-1} \geq \ell \geq 2$ in the sense that

$$|\lambda^* - \mu_i| < \epsilon_\mu \quad \forall i = 1, \ldots, \ell \tag{13}$$

where $\epsilon_\mu > 0$ is some appropriate tolerance, then the following correction step is invoked. For notational simplicity, q and $\bar{q}$ are used for q_{k-1} and $\bar{q}_{k-1}$, respectively. The inputs to the correction step are λ^*, M^*, ℓ from (12) and (13).

Correction step: COS_OPA(λ^*, M^*, ℓ)

Output: Y^*, Z^* and η.

Step 1: Let $\eta = 0$ and $(\lambda^* I - M^*)$ is approximated by rank $\bar{q} - \ell$ matrices $Y_\eta, Z_\eta \in \mathbb{R}^{\bar{q}\times(\bar{q}-\ell)}$ in the form of $\lambda^* I - M^* \approx Y_\eta Z'_\eta$.

Step 2: Solve the following optimization problem over variables $\lambda \in \mathbb{R}$, $M \in \mathcal{S}^{\bar{q}}_+$, Δ^Y, $\Delta^Z \in \mathbb{R}^{\bar{q}\times(\bar{q}-\ell)}$

$$\begin{aligned} & OPC_A(Y_\eta, Z_\eta, \Phi_{k-1}, b_{k-1}) : && (14a) \\ & \min \|\lambda I - M - Y_\eta Z'_\eta - Y_\eta(\Delta^Z)' - (\Delta^Y)Z'_\eta\|_F \\ & \text{s.t. } M \succeq \epsilon_M I_{\bar{q}}, (11c), \|\Delta^Z\|_F \leq \epsilon_Z, \|\Delta^Y\|_F \leq \epsilon_Y \end{aligned}$$

and let its optimizers be $\hat{\lambda}_\eta$, $\hat{M}_\eta$, $\hat{\Delta}^Y_\eta$, $\hat{\Delta}^Z_\eta$.

Step 3: If $\|\hat{\lambda}_\eta I - \hat{M}_\eta - (Y_\eta + \hat{\Delta}^Y_\eta)(Z_\eta + \hat{\Delta}^Z_\eta)'\|_F < \epsilon_C \cdot \bar{q}$, let $Y^* = Y_\eta + \hat{\Delta}^Y_\eta$ and $Z^* = Z_\eta + \hat{\Delta}^Z_\eta$ and stop.

Step 4: If $\eta \geq \eta_{max}$, then stop. Else let $Y_{\eta+1} = Y_\eta + 2\hat{\Delta}^Y_\eta$, $Z_{\eta+1} = Z_\eta + 2\hat{\Delta}^Z_\eta$, $\eta = \eta + 1$ and go to Step 2.

The motivation of the correction step is clear. It aims to search for the $Y_\eta Z'_\eta$ of rank $\bar{q} - \ell$ that best approximates $(\hat{\lambda}_\eta I - \hat{M}_\eta)$. However, $Y_\eta Z'_\eta + Y_\eta(\Delta^Z)' + (\Delta^Y)Z'_\eta$ may not be of rank $\bar{q} - \ell$. Hence, the stopping criterion of Step 3 uses $(Y_\eta + \hat{\Delta}^Y_\eta)(Z_\eta + \hat{\Delta}^Z)'$ to enforce that $\hat{\lambda}_\eta I - \hat{M}_\eta$ has a nullspace of dimension ℓ or more. This stopping criterion differs from the objective function in (14) by $\hat{\Delta}^Y_\eta(\hat{\Delta}^Z_\eta)'$, a term that is negligible when $Y_\eta Z'_\eta$ is sufficiently close to $(\hat{\lambda}_\eta I - \hat{M}_\eta)$. The updates of Y_η and Z_η in step 4 are based on (16a) used in ensuring properties (ii) of Lemma 4. The choice of (Y_0, Z_0) is important to ensure the convergence to the correct low rank Y^*, Z^*. While some options exists, the choice of (Y_0, Z_0) used is

$$Y_0 = Z_0 = U_{\bar{q}-\ell}\Sigma^{0.5}_{\bar{q}-\ell} \tag{15}$$

where $U\Sigma U'$ is the Singular Value Decomposition of $(\lambda^* I - M^*)$ and $U_{\bar{q}-\ell}$ is the first $\bar{q} - \ell$ columns of U, $\Sigma^{0.5}_{\bar{q}-\ell} = diag\{\sigma^{0.5}_1, \ldots, \sigma^{0.5}_{\bar{q}-\ell}\}$ with $\sigma_1, \ldots, \sigma_\ell$ being the largest ℓ singular values of $(\lambda^* I - M^*)$. From the Eckart–Young–Mirsky theorem (Markovsky, 2008), (Y_0, Z_0) of (15) is the best approximation to $(\lambda^* I - M^*)$ in the sense that $Y_0 Z'_0 = \Phi$ where Φ is the optimal solution to $\min_{\Phi\in\mathbb{R}^{\bar{q}\times\bar{q}}}\{\|\lambda^* I - M^* - \Phi\|_F | rank(\Phi) \leq \bar{q} - \ell\}$.

The use of full rank matrices, Y and Z to find a low-rank approximation have appeared in the literature (Fazel et al., 2004). However, its use as a correction step after nuclear norm optimization is novel, to the best of the authors' knowledge. The convergence of COS_OPA is stated in the next lemma.

Lemma 4. *Suppose λ^*, M^*, ℓ are obtained from (12) and (13) and $\{\hat{\lambda}_\eta, \hat{M}_\eta, \hat{\Delta}^Y_\eta, \hat{\Delta}^Z_\eta\}$ is the sequence of solutions obtained from (14) following Step 1–4. (i) If $OPC_A(Y_{\eta-1}, Z_{\eta-1}, \Phi_{k-1}, b_{k-1})$ has a feasible solution, so does $OPC_A(Y_\eta, Z_\eta, \Phi_{k-1}, b_{k-1})$. (ii) Let $\alpha_\eta := \|\hat{\lambda}_\eta I - \hat{M}_\eta - Y_\eta Z'_\eta - Y_\eta(\hat{\Delta}^Z_\eta)' - (\hat{\Delta}^Y_\eta)Z'_\eta\|_F$. Then, $\{\alpha_\eta\}$ converges.*

Proof. (i) Since $OPC_A(Y_{\eta-1}, Z_{\eta-1}, \Phi_{k-1}, b_{k-1})$ has a feasible solution and is a convex optimization problem, its optimal solution $(\hat{\lambda}_{\eta-1}, \hat{M}_{\eta-1}, \hat{\Delta}^Y_{\eta-1}, \hat{\Delta}^Z_{\eta-1})$ can be computed. It is easy to verify that a feasible solution to $OPC_A(Y_\eta, Z_\eta, \Phi_{k-1}, b_{k-1})$ is $(\hat{\lambda}_{\eta-1}, \hat{M}_{\eta-1}, -\hat{\Delta}^Y_{\eta-1}, -\hat{\Delta}^Z_{\eta-1})$. (ii) It follows from (i) that

$$\alpha_\eta \le \|\hat{\lambda}_{\eta-1}I - \hat{M}_{\eta-1} - Y_\eta Z'_\eta + Y_\eta(\hat{\Delta}^Z)' + (\hat{\Delta}^Y)Z'_\eta\|_F$$

$$= \|\hat{\lambda}_{\eta-1}I - \hat{M}_{\eta-1} - \frac{1}{2}Y_\eta(Z_\eta - 2\hat{\Delta}^Z_{\eta-1})' - \frac{1}{2}(Y_\eta - 2\hat{\Delta}^Y_{\eta-1})Z'_\eta\|_F \quad (16a)$$

$$= \|\hat{\lambda}_{\eta-1}I - \hat{M}_{\eta-1} - \frac{1}{2}Y_\eta Z'_{\eta-1} - \frac{1}{2}(Y_{\eta-1})Z'_\eta\|_F \quad (16b)$$

$$= \|\hat{\lambda}_{\eta-1}I - \hat{M}_{\eta-1} - \frac{1}{2}(Y_{\eta-1})(Z_{\eta-1} + 2\hat{\Delta}^Z_{\eta+1})'$$
$$- \frac{1}{2}(Y_{\eta-1} + 2\hat{\Delta}^Y_{\eta-1})Z'_{\eta-1}\|_F = \alpha_{\eta-1}$$

where the first inequality follows from $\{\hat{\lambda}_\eta, \hat{M}_\eta, \hat{\Delta}^Y_\eta, \hat{\Delta}^Z_\eta\}$ being the optimal at time η and the equality of (16b) follows from update formulae given in Step 4 of COS_OPA. Since $\{\alpha_\eta\}$ is a monotonic non-increasing sequence bounded from below, it converges. □

As mentioned earlier, the other subalgorithm at iteration k is OPB. OPB is similar to OPA except that λ is not a variable. Instead, λ is a prescribed value taken successively from $\mathcal{C}_{k-1}\setminus\{0\}$. The need for such a step arises from the nonlinear nature of the rank function. Numerical experiment suggests that the same eigenvalue can be obtained from M even though this eigenvalue has been obtained from OPA or COS_OPA in an earlier iteration. Hence, the intention of OPB is to check if additional multiplicities can be added to those eigenvalues in $\mathcal{C}_{k-1}\setminus\{0\}$. Specifically, the optimization problem is

$$OPB(\lambda, \Phi_{k-1}, b_{k-1}) : min_M \ \|\lambda I - M\|_* \quad (17a)$$

$$\text{s.t.} \quad M \succeq \epsilon_M I_n, (11c). \quad (17b)$$

Similar to OPA, a correction step COS_OPB is invoked to search for additional multiplicity of λ from $\mathcal{C}_{k-1}\setminus\{0\}$ when condition (13) is detected. The COS_OPB routine is essentially the same as COS_OPA except that λ is no longer a variable. Hence, steps 1–4 have to be modified accordingly so that λ is a fixed parameter. Details are omitted due to space limitations. The convergence of COS_OPB can be easily shown following the arguments in the proof of Lemma 4.

Corollary 1. *Suppose $\mathcal{L}_{k-1}$ satisfies (L1)–(L3) and (A1). Then (i) OPA(Φ_{k-1}, b_{k-1}) and OPB($\lambda, \Phi_{k-1}, b_{k-1}$) have a feasible solution, (ii) $H(M^*)$ where M^* is the optimizer of OPA(Φ_{k-1}, b_{k-1}) or OPB(λ, Φ_{k-1}, b_{k-1}) satisfies (L1)–(L3) and (A1). (iii) If COS_OPA or COS_OPB terminates at Step 3 with $\hat{M}_\eta$ being the solution, then $H(\hat{M}_\eta)$ satisfies (L1)–(L3) and (A1).*

Proof. The proof follows same reasonings as those given in the proof of Lemma 3. □

4. The overall algorithm

The main algorithm can now be stated in 1. In the description of the algorithm, quantities r_A, r^i_B, r_B are the estimates of the rank of M_A, M^i_B and M_B, respectively. Note that the ranks of M_A, M^i_B and M_B are never computed exactly. Their values are only guaranteed via the successful termination of the COS routine, as given by steps 2 and 4(ii), respectively.

The overall convergence of Algorithm 1 is guaranteed since OPA and OPB are convex optimization problems and the convergence of OPC_A and OPC_B are ensured by result in Lemma 4. Numerical solution of OPA and OPB can be obtained using SeDuMi (Sturm, 1999) under the CVX interface (Grant & Boyd, 2013) and the conversion to standard LMI formulation is discussed in Recht et al. (2010).

Algorithm 1 The Minimal Polynomial Algorithm

```
Input: $\mathcal{A}_s(G)$ (the standard adjacency matrix of graph G), $\epsilon_M, \epsilon_Z, \epsilon_Y, \epsilon_C, \epsilon_\mu$.
Output: $\mathcal{L}(W_k, G), \mathcal{C}_k = \{\lambda_1, \dots, \lambda_k\}$ and $\mathcal{M}_k = \{m_1, \dots, m_k\}$.
Initialization: Extract $V(G), \mathcal{E}(G)$ from $\mathcal{A}_s(G)$ and let $\mathcal{L}_0 = \mathcal{L}_s$. Set
    $\mathcal{C}_0 = \{0\}, \mathcal{M}_0 = \{1\}, \xi(0) = 1$ and $k = 1$.
Main
1 Compute the eigen-decomposition of $\mathcal{L}_{k-1} = Q_{k-1}\Lambda_{k-1}Q'_{k-1}$
      according to (9) where $Q_{k-1}$ is obtained either from scratch
      or incrementally per Remark 2. Set up $\Phi_{k-1}, b_{k-1}$ according to (7).
      Compute $q_{k-1}, \bar{q}_{k-1}$ according to (10). Call
      OPA($\Phi_{k-1}, b_{k-1}$) and denote its optimizer as $(\lambda^\dagger_A, M^\dagger_A)$ with
      $spec(M^\dagger_A) = \{\mu_1, \dots, \mu_{\bar{q}_{k-1}}\}$. Let $r_A = \bar{q}_{k-1}$.
2 If (13) is satisfied with $\ell \ge 2$, then Call COS_OPA($\lambda^\dagger_A, M^\dagger_A, \ell$).
      If COS_OPA terminates successfully, let the optimizer be
      $(\lambda^*_A, M^*_A)$ and $r_A = \bar{q}_{k-1} - \ell + 1$.
3 If $\xi(k-1) = 1$, let $r_B = \bar{q}_{k-1}$ and goto step 5. Else, let $n_C =$
      $\xi(k-1)$.
4 For each $i = 2, \dots, n_C$,
      (i) call OPB($\lambda_i, \Phi_{k-1}, b_{k-1}$) and denote its optimizer as $M^\dagger_i$
          with $spec(M^\dagger_i) = \{\mu_1, \dots, \mu_{\bar{q}_{k-1}}\}$. Let $r^i_B = \bar{q}_{k-1}$.
      (ii) If (13) is satisfied with $\ell \ge 1$, then call COS_OPB
          ($\lambda_i, M^\dagger_i, \ell$). If COS_OPB terminates successfully, let its
          optimizer be $M^*_i$ and $r^i_B = \bar{q}_{k-1} - \ell$.
      Next $i$
      Let $i^*_B = \arg\min_{i=2,\dots,n_C} r^i_B$ and $r_B = r^{i^*_B}_B$. If $r_B \le \bar{q}_{k-1} - 1$, let
      $(\lambda_B, M_B) = (\lambda^{i^*_B}, M^*_{i^*_B})$.
5 If $r_A = \bar{q}_{k-1}$ and $r_B = \bar{q}_{k-1}$, then terminate.
6 If $r_A < r_B$, then let $(\lambda^*, M^*) = (\lambda^*_A, M^*_A), \mathcal{C}_k = \mathcal{C}_{k-1} \cup \{\lambda^*\}, \xi(k) =$
      $\xi(k-1) + 1, m_{\xi(k)} = \bar{q}_{k-1} - r_A + 1, \mathcal{M}_k = \mathcal{M}_{k-1} \cup \{m_{\xi(k)}\}$.
      Else, let $(\lambda^*, M^*) = (\lambda^*_B, M^*_B), \mathcal{C}_k = \mathcal{C}_{k-1}, \xi(k) = \xi(k-1)$,
      $m_{i^*_B} = m_{i^*_B} + (\bar{q}_{k-1} - r_B), \mathcal{M}_k = \mathcal{M}_{k-1}$.
7 Let $\mathcal{L}_k = H(M^*)$ where $H(\cdot)$ is that given by (10) and $k = k + 1$.
      Go to 1.
```

5. Numerical examples

This section begins with the examples of graph with known minimal polynomials. In particular, a complete graph, a star-shaped graph and a special regular graph (van Dam & Haemers, 1998) are used. For each, the minimal polynomials of standard Laplacian are used as a reference. The parameters in Algorithm 1 are as follows: $\epsilon_M = 0.01$, $\epsilon_Z = 0.01$, $\epsilon_Y = 0.01$, $\epsilon_C = 10^{-7}$, $\epsilon_\mu = 0.01$. Most of the computations in this section can be done in tens of seconds except when $n = 50$ where the computational times are in the range of 5-10 min on a Windows 7 PC with a Intel Core i5-3570 processor and 8 GB memory. The matlab implementation of this code is available in Wang and Ong (2017).

The spectra of the standard Laplacian matrices of these graphs are as follows: $\{0, 8, 8, 8, 8, 8, 8, 8\}$, $\{0, 1, 1, 1, 1, 1, 1, 8\}$, and $\{0, 4, 4, 4, 4, 4, 4, 8\}$, respectively. The corresponding spectra of $\mathcal{L}(W, G)$ after Algorithm 1 are $\{0, 0.8876, 0.8876, 0.8876, 0.8876, 0.8876, 0.8876, 0.8876\}$, $\{0, 7.9988, 0.9998, 0.9998, 0.9998, 0.9998, 0.9998, 0.9998\}$ and $\{0, 2.0006, 1.0003, 1.0003, 1.0003,$

Table 1
The steps in Algorithm 1 for the graph in Fig. 1(d): $s_{\mathcal{L}_k}$ denotes the order of $m_{\mathcal{L}_k}$.

k	Defining step	Spectra of $H(M^*)$	$s_{\mathcal{L}_k}$
0	Standard Laplacian	{0, 2.5721, 3.7509, 4.5858, 6.6243, 7.1464, 7.4142, 8, 8.8035, 9.1028}	10
1	COS_OPA	{0, 2.7248, 1.0923, 0.9993, 0.9995, 2.2656, 2.1184, 2.0841, 2.0841, 2.0841}	8
2	COS_OPB	{0, 2.9857, 0.7331, 1.3773, 1.5193, 2.3287, 2.0841, 2.0841, 2.0841, 2.0841}	7
3	COS_OPA	{0, 2.9382, 0.7463, 2.2875, 1.3773, 1.3773, 2.0841, 2.0841, 2.0841, 2.0841}	6
4	Terminated	{0, 2.9382, 0.7463, 2.2875, 1.3773, 1.3773, 2.0841, 2.0841, 2.0841, 2.0841}	6

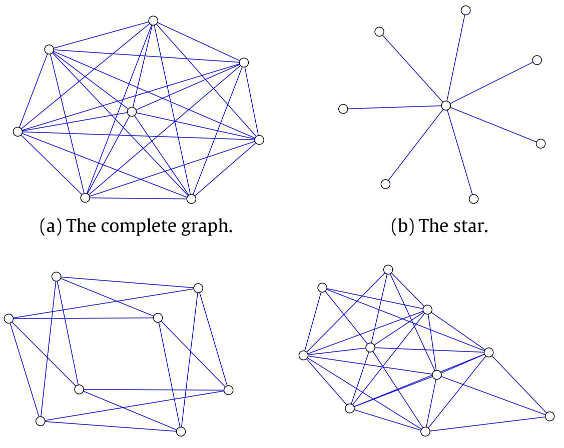
(a) The complete graph. (b) The star. (c) The regular graph. (d) A random graph.

Fig. 1. Special examples and a 10-agent random graph.

Table 2
The mean and standard deviation of the order of the minimal polynomial over 20 random graphs: $\bar{s}_{\mathcal{L}(W)}$ denotes the mean of $s_{\mathcal{L}(W)}$ and $s^{std}_{\mathcal{L}(W)}$ denotes the standard deviation of $s_{\mathcal{L}(W)}$.

Threshold	n	$\bar{s}_{\mathcal{L}(W)}$	$s^{std}_{\mathcal{L}(W)}$	$\bar{s}_{L_s}$	$s^{std}_{L_s}$
0.3	10	5.45	0.865	7.55	1.43
0.6	10	8.5	0.5	9.95	0.218
0.3	20	7.85	1.96	19.8	0.536
0.6	20	16.9	0.624	20	0
0.3	50	19.3	1.58	50	0
0.6	50	40.1	1.26	50	0

1.0003, 1.0003, 1.0003}, respectively. Hence, the algorithm preserves the order of the minimal polynomials for these cases.

The next example is a 10-agent graph with a randomly generated topology as given in Fig. 1(d). It is used to illustrate the progress of a $\mathcal{L}(W, G)$ of a typical graph as it goes through Algorithm 1 as well as to check the relevance of the subroutines involved. The next table shows such a case. The second column (labeled Defining Step) of Table 1 refers to the procedure that determines the M matrix of $H(M)$. As the table shows, all routines (OPA, COS_OPA, OPB and COS_OPB) are needed to achieve the minimal polynomial. It also validates the necessity of OPB and COS_OPB in the algorithm.

Table 2 shows results from Algorithm 1 on graphs for which the minimal polynomials are unknown. These graphs are generated based on the following procedure. For every pair of nodes in $\mathcal{V}$, the existence of a link connecting them follows a uniform density function with a threshold. The link exists if and only if the density function returns a value above the threshold. Graphs of various sizes and topologies are generated in this way. For each graph G generated, validity of assumption (A1) is ensured by checking that the second smallest eigenvalue of $\mathcal{L}_s$ satisfies $\lambda_2(\mathcal{L}_s(G)) > 0$. For each choice of threshold and size, 20 examples are generated randomly. Let $s_{\mathcal{L}(W)}$ denote the order of $m_{\mathcal{L}(W)}$ from Algorithm 1 and recall that $s_{\mathcal{L}(W)} - 1$ is the number of steps needed to achieve consensus from (6). The mean and standard deviations of $s_{\mathcal{L}(W)} - 1$ over the 20 examples are given in Table 2.

Two trends are clear from Table 2 besides the obvious that the order of the minimal polynomial increases for increasingly sparse networks. Let $\bar{s}_{\mathcal{L}(W)}$ denote the mean of $s_{\mathcal{L}(W)}$ (third column of Table 2). The first is that the relative decrease in $\frac{\bar{s}_{\mathcal{L}(W)}}{n}$ from dense to sparse networks increases for increasing values of n. It went from 0.85 to 0.54 when $n = 10$ and 0.8 to 0.386 when $n = 50$. This suggests that the proposed approach is more effective for larger networks. The second trend is that percentage of decrease in the order of the minimal polynomial is more pronounced for better-connected system — the ratio of $\frac{\bar{s}_{\mathcal{L}(W)}}{n}$ increases from 0.545 to about 0.386 when n increases from 10 to 50. The percentage decrease is less in the case for sparsely-connected networks, $\frac{\bar{s}_{\mathcal{L}(W)}}{n}$ goes from 0.85 to 0.8 for the corresponding increase in n. This suggests that the difficulties in reducing the order of minimal polynomial for sparsely-connected networks.

6. Conclusions

This work presents an approach to speed up finite-time consensus of a multi-agent system of a known graph by searching over the weights of a weighted Laplacian matrix. This work uses a novel iterative process wherein two optimization problems are solved at each iteration. In each optimization problem, a nuclear norm convex optimization is first solved followed by a correction step via a low-rank approximation. The numerical experiment shows that the minimal polynomials obtained from the iterative process are of a lower or equal order to that obtained from standard Laplacian. Using the minimal polynomial so obtained, finite-time consensus can be achieved in $(s_{\mathcal{L}} - 1)$ time step where $s_{\mathcal{L}}$ is the order of the minimal polynomial of the weighted Laplacian of the known graph.

References

Björck, A., & Pereyra, V. (1970). Solution of Vandermonde systems of equations. *Mathematics of Computation*, *24*(112), 893–903.
Fallat, S., & Hogben, L. (2007). The minimum rank of symmetric matrices described by a graph: A survey. *Linear Algebra and its Applications*, *426*(2–3), 558–582.
Fazel, M., Hindi, H., & Boyd, S. (2004). Rank minimization and applications in system theory. In *Proceedings of the American control conference, Vol. 4* (pp. 3273–3278). IEEE.
Friedberg, S. H., Insel, A., & Spence, L. E. (2003). *Linear algebra* (4th ed.). Prentice Hall.
Grant, M., & Boyd, S. (2013). CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://Cvxr.Com/Cvx.
Hendrickx, J. M., Jungers, R. M., Olshevsky, A., & Vankeerberghen, G. (2014). Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, *50*(2), 635–640.
Hendrickx, J. M., Shi, G., & Johansson, K. H. (2015). Finite-time consensus using stochastic matrices with positive diagonals. *IEEE Transactions on Automatic Control*, *60*(4), 1070–1073.
Ko, C., & Shi, L. (2009). Scheduling for finite time consensus. In *Proceedings of the American control conference* (pp. 1982–1986). IEEE.
Markovsky, I. (2008). Structured low-rank approximation and its applications. *Automatica*, *44*(4), 891–909.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, *49*(9), 1520–1533.

Recht, B., Fazel, M., & Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, *52*(3), 471–501.

Ren, W., & Beard, R. W. (2007). *Distributed consensus in multi-behicle cooperative control: theory and applications*. Springer.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, *11*(1–4), 625–653.

Sundaram, S., & Hadjicostis, C. N. (2007). Finite-time distributed consensus in graphs with time-invariant topologies. In *Proceedings of American control conference* (pp. 711–716).

van Dam, E. R., & Haemers, W. H. (1998). Graphs with constant μ and μ. *Discrete Mathematics*, *182*(1–3), 293–307.

van Dam, E. R., Koolen, J. H., & Tanaka, H. (2014). Distance-regular graphs. arXiv preprint arXiv:1410.6294.

Wang, Z., & Ong, C. (2017). A Matlab code for computing a low-order minimal polynomial of an undirected graph. https://sites.google.com/view/ongcjwebsite/download.

Wang, L., & Xiao, F. (2010). Finite-time consensus problems for networks of dynamic agents. *IEEE Transactions on Automatic Control*, *55*(4), 950–955.

Xiao, L., & Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, *53*(1), 65–78.

Yuan, Y., Stan, G., Shi, L., Barahona, M., & Goncalves, J. (2013). Decentralised minimum-time consensus. *Automatica*, *49*(5), 1227–1235.

Yuan, Y., Stan, G., Shi, L., & Goncalves, J. (2009). Decentralized final value theorem for discrete-time LTI systems with applications to minimal-time distributed consensus. In *Proceedings of the 48th IEEE conference on decision and control*.

Zheming Wang received his B.S. degree from the Department of Mechanical Engineering & Automation of Shanghai Jiao Tong University in 2012 and Ph.D. degree from the Department of Mechanical Engineering of National University of Singapore in 2016. His main research interests lie in model predictive control and distributed optimization algorithms.

Chong Jin Ong received his B.Eng. (Hons) and M.Eng. degrees in mechanical engineering from the National University of Singapore in 1986 and 1988, respectively, and the M.S.E. and Ph.D. degrees in mechanical and applied mechanics from University of Michigan, Ann Arbor, in 1992 and 1993, respectively. He joined the National University of Singapore in 1993 and is now an Associate Professor with the Department of Mechanical Engineering. His research interests are in model predictive control, robust control and machine learning.