Asynchronous Implementation of Distributed Coordination Algorithms

Chen, Yao; Xia, Weiguo; Cao, Ming; Lu, Jinhu

Published in:
IEEE Transactions on Automatic Control

DOI:
10.1109/TAC.2017.2756340

Asynchronous Implementation of Distributed Coordination Algorithms: Conditions Using Partially Scrambling and Essentially Cyclic Matrices

Yao Chen, *Member, IEEE*, Weiguo Xia, *Member, IEEE*, Ming Cao, *Senior Member, IEEE* and Jinhu Lü *Fellow, IEEE*

***Abstract*—Given a distributed coordination algorithm (DCA) for agents coupled by a network, which can be characterized by a stochastic matrix, we say that the DCA can be *asynchronously* implemented if the consensus property is preserved when the agents are activated to update their states according to their own clocks. This paper focuses on two central problems in asynchronous implementation of DCA: which class of DCA can be asynchronously implemented, and which other cannot. We identify two types of stochastic matrices, called partially scrambling and essentially cyclic matrices, for which we prove that DCA associated with a partially scrambling matrix can be asynchronously implemented, and there exists at least one asynchronous implementation sequence which fails to realize consensus for DCA associated with an essentially cyclic matrix.**

***Index Terms*—Distributed coordination algorithm, asynchronous implementation, partially scrambling matrix, essentially cyclic matrix.**

I. Introduction

Distributed coordination algorithms (DCA) belong to a typical class of algorithms which gives rise to emerging collective behavior in complex systems through local interactions [13], [19]. Using such an algorithm, each agent updates its state through averaging those of its neighbors, making the states of all agents converge to some identical value, called consensus [4], [17], [15], [30]. Due to the special distributed converging property of DCA, it can be used not only in solving practical engineering problems, such as distributed gradient-descent for large-scale convex optimization problems [16], but also for explaining interesting social phenomena, such as opinion formation in social networks [9].

The convergence of DCA relates closely to the convergence of products of stochastic matrices [6], [7], [20]-[23], [25]- [28], [29], the analysis of which is difficult since the commonly used smooth Lyapunov function cannot be easily found [18]. An effective method for the analysis of DCA is evaluating the ergodic coefficient of the corresponding matrix products [10], [20], [21], [25], based on which many interesting results have been reported [1], [2], [12], [22], [23]. It should be noted that the constructed ergodic coefficients for DCA are generally non-smooth, the magnitude of which has strong connection with the structure of the corresponding graphs describing how the agents are coupled together. Based on this observation, the graphical approach, rather than the algebraic approach, usually plays a critical role in the analysis of DCA [4], [27]. Specifically, all the existing results only focus on some specific types of matrices, since the analysis on products of general stochastic matrices is much harder [24] and in fact is an open problem in the field of DCA. In this paper, we will use the graphical approach to study the asynchronous implementation of DCA with some special graphical structures.

The asynchronous implementation of DCA means that the state updating of each agent follows an independent clock, and it has been proved that asynchronous updating of states also guarantees consensus if self-loops are preserved in the graph [3]. However, for general DCA without self-loops in the graph, the dynamics of asynchronous implementation are rather complicated, and an important fact is that asynchronous updating may not lead to consensus even if the corresponding synchronous updating does [26]. Based on this observation, an interesting question for DCA is what type of DCA reaches consensus when implemented asynchronously. As a step towards answering this question, Xia and Cao proved that any asynchronous updating achieves consensus if the associated graph is neighbor-shared [26] (i.e., the associated stochastic matrix is scrambling), where by a neighbor-shared graph it is meant that any two nodes in the graph share a common neighbor [4]. For a further step, it is natural to ask: can we find a larger set of graphs in which any associated DCA guarantees consensus for any asynchronous implementation? Besides this problem, this paper also tries to address the corresponding inverse question: what kind of DCA cannot be asynchronously implemented in the sense that there always exists an asynchronous implementation for the given DCA which cannot lead to consensus? In this paper, we will report two sets of stochastic matrices that have been constructed for the first time, giving answers to the above two questions.

The rest of the paper is organized as follows: Section II formulates the asynchronous implementation problem; Section III proposes a set of stochastic matrices, called partially scrambling matrices, and proves that any partially scrambling matrix can be asynchronously implemented; Section IV gives

Yao Chen is with the Department of Computer Science, Southwestern University of Finance and Economics, Chengdu 611130, China (e-mail: chenyao07@gmail.com). Weiguo Xia is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: wgxia seu@dlut.edu.cn). Ming Cao is with the Faculty of Science and Engineering, ENTEG, University of Groningen, Groningen 9747 AG, the Netherlands (e-mail: m.cao@rug.nl). Jinhu Lü is with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: jhlu@iss.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grants 61304157 and 61603071, and the Fundamental Research Funds for the Central Universities under Grant DUT15RC(3)131.

a set of stochastic matrices, called essentially cyclic matrices, and proves that each essentially cyclic matrix cannot be asynchronously implemented; Section V presents some examples and corollaries; Section VI concludes this paper.

II. Problem Formulation

Any stochastic matrix[1] $A = (a_{ij})_{i,j=1}^N$ can be described by a graph $\mathcal{G}(A) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \cdots, N\}$ is the set of nodes and $\mathcal{E}$ is the set of edges: $(i, j) \in \mathcal{E}$ if and only if $a_{ji} > 0$. Given a set $\mathcal{S} \subseteq \mathcal{V}$, $\mathcal{G}_\mathcal{S}$ is defined as the induced subgraph of $\mathcal{G}$ over $\mathcal{S}$.

A directed path in $\mathcal{G}(A)$ is a sequence of distinct nodes $i_1, \cdots, i_k$ such that $(i_s, i_{s+1}) \in \mathcal{E}$ for $1 \le s \le k-1$. $\mathcal{G}(A)$ is rooted if it contains a node, called a root, that has a directed path to every other node. If $\mathcal{G}(A)$ is rooted, we define $\mathrm{root}(A)$ as the set of all the roots of $\mathcal{G}(A)$. Specifically, we define the following function $\mathcal{N}(\cdot, \cdot)$ for any stochastic matrix:

$$\mathcal{N}(A, \mathcal{S}) = \{j : a_{ij} > 0, i \in \mathcal{S}\},$$

where $A \in \mathbb{R}^{N \times N}$ is stochastic and $\mathcal{S}$ is a subset of $\mathcal{V}$.

Stochastic matrices can be used to describe the distributed coordination algorithm in the form

$$x_{k+1} = Ax_k, \quad k \ge 1,$$

where $x_k \in \mathbb{R}^N$ and $A \in \mathbb{R}^{N \times N}$ is a stochastic matrix. If A is SIA (i.e. stochastic, indecomposable, and aperiodic) [25], then for any $x_1 \in \mathbb{R}^N$, there exists $\xi \in \mathbb{R}$ such that $\lim_{k\to\infty} x_k = \mathbb{1}\xi$, where $\mathbb{1} \in \mathbb{R}^N$ is the all-one vector [3].

Given a stochastic matrix $A = (a_{ij})_{i,j=1}^N$, let $a_k \in \mathbb{R}^{1 \times N}$ ($k = 1, 2, \cdots, N$) denote its kth row. Define the following matrix

$$A_k = (e_1, \cdots, e_{k-1}, a_k^T, e_{k+1}, \cdots, e_N)^T, \tag{1}$$

where $e_k \in \mathbb{R}^N$ is the unit vector with the kth entry being 1. Since matrix A is stochastic, one can verify that A_k ($k = 1, 2, \cdots, N$) is also stochastic. The matrix A_k is called the asynchronous implementation of A on the kth node.

Given a stochastic matrix $A \in \mathbb{R}^{N \times N}$, a sequence of matrices $\{A_{\sigma(k)}\}_{k=1}^\infty$ ($\sigma(k) \in \mathcal{V}$) is called an asynchronous implementation sequence of matrix A if $\bigcup_{k=j}^{j+q-1} \{\sigma(k)\} = \mathcal{V}$ for all $j \ge 1$. An asynchronous implementation sequence $\{A_{\sigma(k)}\}_{k=1}^\infty$ of matrix A is said to realize consensus if for any initial condition $x_1 \in \mathbb{R}^N$, it holds

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} A_{\sigma(k)} \cdots A_{\sigma(2)} A_{\sigma(1)} x_1 = \mathbb{1}\xi. \tag{2}$$

where $\xi \in \mathbb{R}$ is a scalar depending on the initial value x_1 and the sequence $\{A_{\sigma(k)}\}_{k=1}^\infty$. If any asynchronous implementation sequence $\{A_{\sigma(k)}\}_{k=1}^\infty$ of matrix A realizes consensus, we say matrix A *can be asynchronously implemented*. If there exists at least one asynchronous implementation sequence $\{A_{\sigma(k)}\}_{k=1}^\infty$ of matrix A which cannot realize consensus, we say matrix A *cannot be asynchronously implemented*.

A stochastic matrix $A = (a_{ij})_{i,j=1}^N$ is called *scrambling* if: for any $i, j \in \mathcal{V}$ ($i \ne j$), there exists k such that $a_{ik} \cdot a_{jk} > 0$. According to [8], one knows that $\mathcal{G}(A)$ is rooted for any scrambling matrix A. In this paper, we use $\mathcal{Q}_\mathrm{s}$ to denote the set of scrambling matrices.

Define the ergodic coefficient of a stochastic matrix $A = (a_{ij})_{i,j=1}^N$ to be

$$\tau(A) = 1 - \min_{1 \le i < j \le N} \sum_{k=1}^N \min(a_{ik}, a_{jk}).$$

Based on the definition of scrambling matrices, it is easy to verify that a stochastic matrix A is scrambling if and only if $\tau(A) < 1$. This ergodic coefficient further satisfies

Proposition 1: [21] For any two stochastic matrices $A_1, A_2 \in \mathbb{R}^{N \times N}$, it holds that

$$\tau(A_1 A_2) \le \tau(A_1) \cdot \tau(A_2).$$

Specifically, the function $\mathcal{N}(\cdot, \cdot)$ and the ergodic coefficient $\tau(\cdot)$ have the following relationship:

Proposition 2: Given a stochastic matrix $A = (a_{ij})_{i,j=1}^N$, if for any two vertices $i, j \in \mathcal{V}$, it holds $\mathcal{N}(A, i) \bigcap \mathcal{N}(A, j) \ne \emptyset$, then $\tau(A) \le 1 - \min_{i,j=1}^N a_{ij}$.

In 2014, Xia and Cao proved the following important property for scrambling matrices

Proposition 3: [26] Given a matrix $A \in \mathcal{Q}_\mathrm{s}$, any asynchronous implementation sequence of A realizes consensus.

The above result motivates us to study the following two interesting problems:

P1) Find a set of stochastic matrices which is larger than $\mathcal{Q}_\mathrm{s}$ in which any asynchronous implementation of each matrix realizes consensus.

P2) Find a set of stochastic matrices in which there exists an asynchronous implementation sequence for each matrix which cannot realize consensus.

In the subsequent two sections, we find two sets of stochastic matrices, called partially scrambling and essentially cyclic matrices, for the solutions of the above two problems respectively.

III. Set of Matrices which Can be Asynchronously Implemented

In what follows, we will introduce the concepts of the absorbing set and partially scrambling matrix first.

For any stochastic matrix $A = (a_{ij})_{i,j=1}^N \in \mathbb{R}^{N \times N}$, a set $\mathcal{S} \subseteq \mathcal{V}$ is called *absorbing* with respect to A if

a) $\mathcal{G}(A)$ is rooted and $\mathcal{S} \bigcap \mathrm{root}(A) \ne \emptyset$;

b) For any $i \in \mathcal{S}$, $\mathcal{N}(A, i) \bigcap \mathcal{S} \ne \emptyset$.

Based on the above definition, one knows that if $\mathcal{G}(A)$ is rooted, then $\mathcal{V}$ is absorbing with respect to A. Specifically, if $a_{kk} > 0$ and $k \in \mathrm{root}(A)$, then the singleton $\{k\}$ is absorbing with respect to A.

A matrix $A = (a_{ij})_{i,j=1}^N$ is called *partially scrambling* if there exists $\nu \in \mathrm{root}(A)$ and an absorbing set $\mathcal{I} \subseteq \mathcal{V}$ which satisfies: for any $i \in \mathcal{I}$, there exists $k \in \mathcal{I}$ such that $a_{ik} a_{\nu k} > 0$.

A simple example of partially scrambling matrix is

$$A = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \end{pmatrix},$$

whose graph $\mathcal{G}(A)$ is given in Fig. 1. One can easily verify that A is partially scrambling by letting $\nu = 3$ and $\mathcal{I} = \{1, 2\}$.

[1]In this paper, when we say a matrix is stochastic, we mean this matrix is right stochastic in which the sum of each row equals 1.

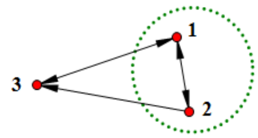

Fig. 1. An example of partially scrambling graph: the set $\mathcal{I} = \{1, 2\}$ is absorbing, $\nu = 3$ shares a common neighbor with each node of $\mathcal{I}$.

Let $\mathcal{Q}_{\rm ps}$ be the set of partially scrambling matrices and we will show that $\mathcal{Q}_{\rm ps}$ is larger than $\mathcal{Q}_{\rm s}$.

Proposition 4: $\mathcal{Q}_{\rm s} \subseteq \mathcal{Q}_{\rm ps}$.

Proof: For any $A \in \mathcal{Q}_{\rm s}$, since any scrambling matrix is rooted [8], one can choose $\nu \in \mathrm{root}(A)$. Furthermore, $\mathcal{V}$ is absorbing with respect to A since $\mathcal{G}(A)$ is rooted. For any $i \in \mathcal{V}$, since A is scrambling, there exists $k \in \mathcal{V}$ such that $a_{ik}a_{\nu k} > 0$. Hence, the two conditions of partially scrambling matrices are both satisfied and $A \in \mathcal{Q}_{\rm ps}$. □

The main result of this section is given as follows:

Theorem 1: Given any matrix $A \in \mathcal{Q}_{\rm ps}$, any asynchronous implementation sequence of matrix A realizes consensus.

The proof of *Theorem* 1 relies on the following *Proposition* 5 and *Lemma* 1-5. In *Proposition* 5 and *Lemma* 1-5, we assume that $A \in \mathcal{Q}_{\rm ps}$, $\{A_{\sigma(k)}\}_{k=1}^{\infty}$ is an asynchronous implementation sequence of A, q is the constant given in the definition of an asynchronous implementation sequence, $\mathcal{I}$ is an absorbing set of $\mathcal{V}$ with respect to A, and $\nu \in \mathrm{root}(A)$.

The basic idea of the proof of *Theorem* 1 can be summarized as follows: At first, we divide the asynchronous implementation sequence $A_{T:1} = A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(1)}$ into two parts for some large T, one is $A_{T:(r+1)} = A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}$, and the other is $A_{r:1} = A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}$; Second, we show that for any two vertices i and j, the function $\mathcal{N}(A_{T:(r+1)}, \cdot)$ makes i accessed by ν, and j accessed by one of the nodes in $\mathcal{I}$ (*Lemma* 3); Third, we show that for any $k' \in \mathcal{I}$, $\mathcal{N}(A_{r:1}, \nu)$ and $\mathcal{N}(A_{r:1}, k')$ share a common element (*Lemma* 4); At last, we combine the above two steps and demonstrate that $\mathcal{N}(A_{T:1}, i)$ and $\mathcal{N}(A_{T:1}, j)$ share a common neighbor (*Lemma* 5), which implies that $A_{T:1}$ is scrambling (*Proposition* 2) and the convergence to consensus can be obtained via *Proposition* 1.

Proposition 5: [11] For any $k \geq 1$ and $\mathcal{S} \subset \mathcal{V}$, it holds

$$\mathcal{N}(A_{\sigma(k+1)}A_{\sigma(k)}, \mathcal{S}) = \mathcal{N}(A_{\sigma(k)}, \mathcal{N}(A_{\sigma(k+1)}, \mathcal{S})).$$

Lemma 1: If $j \in \mathcal{I}$, then for any $T \geq 1$, we have

$$\mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(1)}, j) \bigcap \mathcal{I} \neq \emptyset.$$

Proof: According to the definition of $\mathcal{I}$, one knows that for any $\sigma(k) \in \mathcal{V}$ and $i \in \mathcal{I}$, $\mathcal{N}(A_{\sigma(k)}, i) \bigcap \mathcal{I} \neq \emptyset$. Applying *Proposition* 5 on $\mathcal{N}(\cdot,\cdot)$ we arrive at the conclusion. □

Lemma 2: For any $i, j \in \mathcal{V}$, if

$$\begin{aligned}\mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}, i) \bigcap \\ \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}, j) \neq \emptyset,\end{aligned}$$

for some $r \geq 0$ and $T \geq r+1$, then

$$\begin{aligned}\mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(1)}, i) \bigcap \\ \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(1)}, j) \neq \emptyset.\end{aligned}$$

Proof: It follows directly from *Proposition* 5. □

Lemma 3: Given $T \geq 2Nq+2$, for any $i, j \in \mathcal{V}$, there exists r which satisfies $T - r \leq 2Nq+1$ and

$$\begin{aligned}\nu \in \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}, i), \\ \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}, j) \bigcap \mathcal{I} \neq \emptyset.\end{aligned}$$

Proof: Since $\mathcal{I}$ is an absorbing set, $\mathcal{I} \bigcap \mathrm{root}(A) \neq \emptyset$. Letting $j_m \in \mathcal{I} \bigcap root(A)$, there exists a directed path from j_m to j in $\mathcal{G}(A)$ denoted by $j_m \to j_{m-1} \to \cdots \to j_1 \to j$, where $m \leq N$.

Denote

$$\begin{aligned} t^{(0)} &= \max\{k : \sigma(k) = j, 1 \leq k \leq T\}, \\ t^{(1)} &= \max\{k : \sigma(k) = j_1, 1 \leq k < t^{(0)}\}, \\ &\cdots \\ t^{(m-1)} &= \max\{k : \sigma(k) = j_{m-1}, 1 \leq k < t^{(m-2)}\}, \end{aligned}$$

from which one obtains that

$$\begin{aligned} j_1 &\in \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(t^{(0)})}, j), \\ j_2 &\in \mathcal{N}(A_{\sigma(t^{(0)}-1)}A_{\sigma(t^{(0)}-2)}\cdots A_{\sigma(t^{(1)})}, j_1), \\ &\cdots \\ j_m &\in \mathcal{N}(A_{\sigma(t^{(m-2)}-1)}A_{\sigma(t^{(m-2)}-2)}\cdots A_{\sigma(t^{(m-1)})}, j_{m-1}). \end{aligned}$$

According to *Proposition* 5, one derives that $j_m \in \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(t^{(m-1)})}, j)$ and $j_m \in \mathcal{I}$. According to the absorbing property of $\mathcal{I}$, for any $k \leq t^{(m-1)}$, we know that

$$\mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(k)}, j) \bigcap \mathcal{I} \neq \emptyset. \tag{3}$$

Specifically, from the property that $\bigcup_{j=k+1}^{k+q} A_{\sigma(j)} = \mathcal{V}$ ($k \geq 0$), one knows

$$\begin{aligned} T - t^{(0)} &\leq q, \\ t^{(i)} - t^{(i+1)} &\leq q, \text{ for } 0 \leq i \leq m-2, \end{aligned}$$

and hence $T - t^{(m-1)} \leq mq \leq Nq$.

Consider another path $i_p \to i_{p-1} \to \cdots \to i_1$ in $\mathcal{G}(A)$, where $i_p = \nu$, $p \leq N$, and

$$i_1 \in \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(t^{(m-1)})}, i), \tag{4}$$

for which the fact that $\nu \in \mathrm{root}(A)$ guarantees the existence of such a path. Similar to the above deductions, one can find some $d \leq Nq$ such that

$$\nu \in \mathcal{N}(A_{\sigma(t^{(m-1)}-1)}A_{\sigma(t^{(m-1)}-2)}\cdots A_{\sigma(t^{(m-1)}-d)}, i_1). \tag{5}$$

Combining (4) and (5) together leads to

$$\nu \in \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(t^{(m-1)}-d)}, i). \tag{6}$$

Let $r = t^{(m-1)} - d - 1$, and one knows

$$\begin{aligned} T - r = (T - t^{(m-1)}) + t^{(m-1)} - r &\leq Nq + d + 1 \\ &\leq 2Nq + 1. \end{aligned}$$

Combining (3) and (6), the proof is hence completed. □

Lemma 4: Given an absorbing set $\mathcal{I}$ and any $k' \in \mathcal{I}$, if $r \geq q(N+q)+1$, there exists $k'' \in \mathcal{V}$ such that

$$\begin{aligned} k'' \in \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, \nu) \bigcap \\ \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, k') \neq \emptyset. \end{aligned}$$

Proof: Since $r \geq q(N+q)+1 \geq q$, one can define

$$t_\nu = \max\{k : \sigma(k)=\nu, 1\leq k\leq r\},$$
$$t_{k'} = \max\{k : \sigma(k)=k', 1\leq k\leq r\},$$

and the property of asynchronous implementation sequence guarantees that $r-t_\nu \leq q-1$ and $r-t_{k'}\leq q-1$.

We make the following discussions:

CASE a): $t_{k'}=t_\nu$.

In this case, one knows $k'=\nu$, the result holds naturally.

CASE b): $t_{k'}>t_\nu$.

Denote

$$\begin{aligned} s_1 &= \{k'\},\\ s'_1 &= \{k : \sigma(k)\in s_1, t_\nu < k \leq r\},\\ t^{(1)} &= \max s'_1, \text{ if } s'_1\neq\emptyset,\\ k_1 &= \sigma(t^{(1)}), \end{aligned}$$

where k_1 also satisfies $k_1=k'$.

Furthermore, for any $p\geq 2$, we construct the following iterative formulas:

$$\begin{aligned} s_p &= \mathcal{N}(A_{\sigma(t^{(p-2)}-1)}\cdots A_{\sigma(t^{(p-1)})}, k_{p-1})\bigcap\mathcal{I},\\ s'_p &= \{k : \sigma(k)\in s_p, t_\nu < k < t^{(p-1)}\},\\ t^{(p)} &= \max s'_p, \quad \text{if } s'_p\neq\emptyset,\\ k_p &= \sigma(t^{(p)}), \end{aligned}$$

where $t^{(0)}=r+1$.

Since $t_\nu < t^{(p)} < t^{(p-1)}$, one knows the condition of $t_\nu < k < t^{(p-1)}$ will not be satisfied after several times of iteration, and hence there exists p such that $s'_p=\emptyset$.

Denote $m=\min\{p : s_p\neq\emptyset \text{ and } s'_p=\emptyset\}$, and then the above iterations imply that

$$\begin{aligned} k_2 &\in \mathcal{N}(A_{\sigma(t^{(0)}-1)}\cdots A_{\sigma(t^{(1)})}, k_1)\bigcap\mathcal{I},\\ k_3 &\in \mathcal{N}(A_{\sigma(t^{(1)}-1)}\cdots A_{\sigma(t^{(2)})}, k_2)\bigcap\mathcal{I},\\ &\cdots\\ k_{m-1} &\in \mathcal{N}(A_{\sigma(t^{(m-3)}-1)}\cdots A_{\sigma(t^{(m-2)})}, k_{m-2})\bigcap\mathcal{I},\\ \emptyset &\neq \mathcal{N}(A_{\sigma(t^{(m-2)}-1)}\cdots A_{\sigma(t^{(m-1)})}, k_{m-1})\bigcap\mathcal{I}. \end{aligned}$$

Consider the pair of indices k_{m-1} and ν, due to the fact that $k_{m-1}\in\mathcal{I}$, there exists $k^*\in\mathcal{I}$ such that $a_{k_{m-1},k^*}a_{\nu,k^*}>0$ and hence $k^*\in\mathcal{N}(A,k_{m-1})\bigcap\mathcal{N}(A,\nu)\bigcap\mathcal{I}$, which leads to

$$\begin{aligned} k^* &\in \mathcal{N}(A_{\sigma(t^{(m-2)}-1)}\cdots A_{\sigma(t^{(m-1)})}, k_{m-1})\bigcap\mathcal{I},\\ k^* &\in \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(t_\nu)}, \nu)\bigcap\mathcal{I}. \end{aligned}$$

If $k^*\neq\nu$, due to the fact that $s'_m=\emptyset$, one further knows

$$k^* \in \mathcal{N}(A_{\sigma(t^{(m-1)}-1)}\cdots A_{\sigma(t_\nu)}, k_{m-1})\bigcap\mathcal{I},$$

which indicates

$$k^* \in \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(t_\nu)}, k_1)\bigcap\mathcal{I}.$$

Therefore,

$$\begin{aligned} k^* \in\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(t_\nu)}, \nu)\bigcap\\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(t_\nu)}, k_1)\bigcap\mathcal{I}. \end{aligned}$$

According to *Lemma* 2 and using the absorbing property of $\mathcal{I}$, one derives that there exists k'' such that

$$\begin{aligned} k'' \in\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, \nu)\bigcap\\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, k_1)\bigcap\mathcal{I}, \end{aligned}$$

which in view of $k_1=k'$ completes the discussion.

If $k^*=\nu$, one derives $\nu\in\mathcal{N}(A_{\sigma(t^{(m-1)}-1)}\cdots A_{\sigma(t_\nu+1)}, k_{m-1})$, which indicates $\nu\in\mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(t_\nu+1)}, k_1)$. Since $\nu\in\mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(t_\nu+1)}, \nu)$, there also exists k'' such that

$$\begin{aligned} k'' \in\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, \nu)\bigcap\\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, k_1). \end{aligned}$$

CASE c): $t_{k'}<t_\nu$.

Since $\nu\in\mathrm{root}(A)$, one can find a cycle from ν to ν with length l ($l\leq N$), repeating this cycle for $[\frac{q}{l}]+1$ times generates a cycle with length $\hat{l}=l([\frac{q}{l}]+1)>q$. Specifically, the length of the merged cycle also satisfies $\hat{l}\leq l([\frac{q}{l}]+1)\leq N+q$. Let the merged cycle be $i_{\hat{l}}=\nu\rightarrow i_{\hat{l}-1}\rightarrow i_{\hat{l}-2}\cdots\rightarrow i_0=\nu$. Similar to the techniques in the proof of *Lemma* 3, one can find $1<r'\leq r$ such that[2]

$$\nu\in\mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(r')}, \nu),$$

where $q<r-r'\leq q(N+q)$.

Based on the definition of r', one further defines

$$t'_\nu = \max\{k : \sigma(k)=\nu, 1\leq k\leq r'\},$$

then $t'_\nu\leq r'<r-q<r-q+1\leq t_{k'}$. The remaining proof is similar to that of CASE b) and hence omitted.

Summarize the above three cases, the proof is hence completed. □

Lemma 5: For any $i,j\in\mathcal{V}$, if $T\geq(3N+q)q+2$, then

$$\begin{aligned} & \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(1)}, i)\bigcap\\ & \mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(1)}, j)\neq\emptyset. \end{aligned}$$

Proof: According to *Lemma* 3, one can find some $k'\in\mathcal{V}$ and r which satisfies $T-r\leq 2Nq+1$ such that

$$\nu\in\mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}, i),$$
$$k'\in\mathcal{N}(A_{\sigma(T)}A_{\sigma(T-1)}\cdots A_{\sigma(r+1)}, j)\bigcap\mathcal{I}\neq\emptyset.$$

Since $T\geq(3N+q)q+2$, one knows $r\geq(N+q)q+1$. According to *Lemma* 4, one further derives

$$\begin{aligned} & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, \nu)\bigcap\\ & \mathcal{N}(A_{\sigma(r)}A_{\sigma(r-1)}\cdots A_{\sigma(1)}, k')\neq\emptyset. \end{aligned}$$

Summarizing the above two facts leads to the completion of the proof. □

Based on the above lemmas, we are ready to present the proof of the main theorem.

Proof of Theorem 1: Given a sequence of implemenation matrices $\{A_{\sigma(k)}\}_{k=1}^{\infty}$, denote

$$Q_k = A_{\sigma(kT)}\cdots A_{\sigma((k-1)T+2)}\cdot A_{\sigma((k-1)T+1)},$$

[2]If $r\leq q(N+q)$, the existence of r' may not be guaranteed.

where $T = (3N + q)q + 2$.

According to *Lemma* 5 and *Proposition* 2, one knows that Q_k is scrambling and hence $\tau(Q_k) \leq 1 - \alpha^T$, where α is the minimal positive entry of A. Since $\prod_{k=1}^{\infty} A_{\sigma(k)} = \prod_{k=1}^{\infty} Q_k$ and $\tau(\prod_{k=1}^{\infty} Q_k) \leq \prod_{k=1}^{\infty} \tau(Q_k)$, we arrive at the conclusion. □

IV. Set of Matrices which Cannot be Asynchronously Implemented

To facilitate the description of the following problem, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set $\mathcal{S} \subseteq \mathcal{V}$, we define the two functions

$$\begin{aligned} \partial^-(\mathcal{S}) &= \{k : (i,k) \in \mathcal{E}, i \in \mathcal{S}, k \notin \mathcal{S}\}, \\ \partial^+(\mathcal{S}) &= \{k : (k,i) \in \mathcal{E}, i \in \mathcal{S}, k \notin \mathcal{S}\}. \end{aligned}$$

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\{V_i\}_{i=1}^r$ be a partition of $\mathcal{V}$: $\bigcup_{i=1}^r V_i = \mathcal{V}$ and $V_{i_1} \bigcap V_{i_2} = \emptyset$ for $i_1 \neq i_2$. The *reduced graph* of $\mathcal{G}$ with respect to $\{V_i\}_{i=1}^r$ is defined by $\widetilde{\mathcal{G}} = (\widetilde{\mathcal{V}}, \widetilde{\mathcal{E}})$, where $\widetilde{\mathcal{V}} = \{1, 2, \cdots, r\}$ and $(i,j) \in \widetilde{\mathcal{E}}$ if and only if there is a link from a node in V_i to a node in V_j.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called a *directed acyclic graph* (DAG) if $\mathcal{G}$ contains no cycle. Based on this definition, one knows that a DAG may not be rooted.

A stochastic matrix $A = (a_{ij})_{i,j=1}^N$ is called *essentially cyclic* if there exists a partition $\{V_i\}_{i=1}^r$ of $\mathcal{V}$ with $r \geq 3$ such that:

a) The subgraph $\mathcal{G}_{V_i}$ is a DAG;
b) The reduced graph with respect to $\{V_i\}_{i=1}^r$ is a directed cycle.

The above definition of essentially cyclic matrices is inspired by the definition of periodic matrices [21]: a stochastic matrix $A = (a_{ij})_{i,j=1}^N$ is called *periodic* if there exists an equivalent partition $\{V_i\}_{i=1}^r$ of $\mathcal{V}$ which makes the corresponding reduced graph a directed cycle, and makes each subgraph $\mathcal{G}_{V_i}$ a null graph.

Given any connected graph $\mathcal{G}$, one can decompose it into several strongly connected components with the corresponding reduced graph being a DAG [14]; such a property is exactly opposite to the decomposition of an essentially cyclic graph, in which each decomposed component contains no cycle but the reduced graph is cyclic.

Based on the definition of essentially cyclic matrices, one knows that

Proposition 6: Any SIP (stochastic, indecomposable, and periodic) matrix is essentially cyclic.

Furthermore, since the equivalent partition satisfies $r \geq 3$, any cycle in the corresponding graph of an essentially cyclic matrix has length greater than 3, which leads to the following proposition.

Proposition 7: Given a stochastic matrix A, if $\mathcal{G}(A)$ contains K_2 as a subgraph[3], then A is not essentially cyclic.

We use $\mathcal{Q}_{ec}$ to denote the set of essentially cyclic matrices. As shown in Fig. 2, the given graph is essential cyclic if we set $V_1 = \{1\}$, $V_2 = \{2\}$, and $V_3 = \{3,4\}$, and then all the items

[3] K_n is the fully connected graph with n nodes.

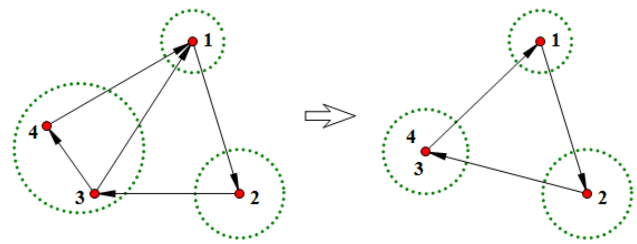

Fig. 2. An example of essentially cyclic graph with $r = 3$: the subgraph of $\{3,4\}$ contains no cycle, and the reduced graph is a cycle.

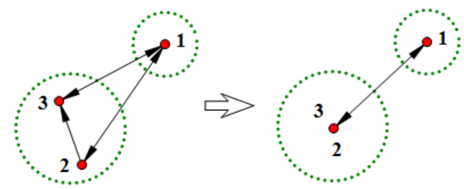

Fig. 3. An essentially cyclic graph with $r = 2$; however, this graph is also partially scrambling as shown in Fig. 1.

in the definition of essentially cyclic graph can be verified. Let the stochastic matrix A corresponding to Fig. 2 be

$$A = \begin{pmatrix} 0 & 0 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

One can further verify that the above A is SIA; however, one can find that the following asynchronous implementation sequence which cannot lead to consensus:

$$A_1(A_2)(A_4A_3) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The above implementation has the following properties:

a) The nodes inside each V_i are implemented as a group;
b) Each sub-implementation maps V_i onto V_j;
c) The order of the implementation follows the order of the reduced cyclic graph.

The main result of this section is based on the above three observations, which can be summarized as the following theorem.

Theorem 2: For any $A \in \mathcal{Q}_{ec}$, there exists an asynchronous implementation which cannot lead to consensus.

We would like to point out that the condition of $r \geq 3$ is very critical for the correctness of *Theorem* 2. Let us consider the graph given in Fig. 1, and this graph can be reorganized as that of Fig. 3, which can be viewed as essentially cyclic with $r = 2$. However, this graph is also partially scrambling as shown in Fig. 1, which makes *Theorem* 2 invalid in this case.

Before giving the detailed proof of *Theorem* 2, we would like to introduce the intuitive idea behind *Theorem* 2: At first, for each subgraph $\mathcal{G}(\mathcal{S})$ which is DAG, we show that there exists an asynchronous implementation sequence which maps $\mathcal{S}$ to $\partial^+(\mathcal{S})$ (*Lemma* 7 and 8) under the operation of $\mathcal{N}(\cdot,\cdot)$; Second, for a given essentially cyclic matrix A with the corresponding graph containing r DAGs, we use the corresponding asynchronous implementation sequences constructed in *Lemma* 7 for r times, and then we obtain an asynchronous implementation sequence of matrix A by combining these r asynchronous implementation sequences; Third, we show that by suitably reordering these r asynchronous implementation sequences the conditions of *Lemma* 6 will be satisfied, which leads to non-consensus.

Lemma 6: Given a stochastic matrix $A \subseteq \mathbb{R}^{N\times N}$, if there exists $V_1, V_2 \subseteq \mathcal{V}$, $V_1 \bigcap V_2 = \emptyset$ such that

$$\mathcal{N}(A, V_1) \subseteq V_2, \quad \mathcal{N}(A, V_2) \subseteq V_1,$$

then matirx A is not SIA.

Proof: By reordering the indices of $\mathcal{V}$, matrix A can be written as

$$A = \begin{pmatrix} 0 & A_{12} & 0 \\ A_{21} & 0 & 0 \\ \times & \times & \times \end{pmatrix},$$

where each '$\times$' means a block matrix with appropriate dimensions. The structure of A implies for any $k \geq 1$, there holds

$$A^{2k} = \begin{pmatrix} \times & 0 & 0 \\ 0 & \times & 0 \\ \times & \times & \times \end{pmatrix}, \quad A^{2k+1} = \begin{pmatrix} 0 & \times & 0 \\ \times & 0 & 0 \\ \times & \times & \times \end{pmatrix}.$$

Hence for a sufficiently large k, any column of A^k cannot be completely positive, which indicates A is not SIA. □

The following *Lemma* 7 defines an ordering function $f(\cdot)$ on a DAG, which is critical for the consequent *Lemma* 8.

Lemma 7: There exists a topological ordering $f(k)$ associated with each node k of a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, i.e., if (i, j) is an edge of $\mathcal{G}$, then $f(i) > f(j)$.

Proof: We define the following function $f(\cdot)$ associated with each node of $\mathcal{V}$:

1. Set $k := 1$, $\mathcal{G}_1 = \mathcal{G}$;
2. Set $V_k = \{j :$ the out degree of j in $\mathcal{G}_k$ is zero$\}$;
3. Set $f(j) = k$ for each $j \in V_k$;
4. Set $\mathcal{G}_{k+1}$ be the subgraph of $\mathcal{G}$ with node set $\mathcal{V}/\{V_i\}_{i=1}^k$;
5. If $\mathcal{G}_{k+1}$ is not null, set $k := k + 1$ and go to step 2.

One can verify that the above function $f(\cdot)$ is a topological ordering of $\mathcal{G}$. □

Lemma 8: Given a stochastic matrix $A \in \mathbb{R}^{N\times N}$ and a set $\mathcal{S} \subsetneq \mathcal{V}$. If $\partial^+(\mathcal{S}) \neq \emptyset$ and the subgraph $\mathcal{G}_\mathcal{S}$ contains no cycle[4], then there exists an asynchronous realization sequence $A_{\sigma(k)}$ $(k = 1, 2, \cdots, s)$ such that

$$\mathcal{N}(A_{\sigma(s)}A_{\sigma(s-1)} \cdots A_{\sigma(1)}, \mathcal{S}) = \partial^+(\mathcal{S}),$$

where $s = |\mathcal{S}|$ and $\bigcup_{k=1}^s \sigma(k) = \mathcal{S}$.

Proof: For the set of nodes $\mathcal{S}$, since $\mathcal{G}_\mathcal{S}$ is a DAG, there exists a topological ordering $f(k)$ for each node k of $\mathcal{S}$. Based on the ordering function $f(\cdot)$ in *Lemma* 7, we define a sequence $\{i_k\}_{k=1}^s$ which satisfies $\bigcup_{k=1}^s \{i_k\} = \mathcal{S}$ and $f(i_1) \leq f(i_2) \leq \cdots \leq f(i_s)$. Then, we will show that $\mathcal{N}(A_{i_1}A_{i_2} \cdots A_{i_s}, \mathcal{S}) \subseteq \partial^+(\mathcal{I})$.

[4]Self-loop is a special case of a cycle and hence is not allowed in $\mathcal{G}_\mathcal{S}$.

For these nodes $i_1, i_2, \cdots, i_s$, without loss of generality, suppose that

$$\begin{aligned} f(i_1) = f(i_2) = \cdots = f(i_{k_1}) &\neq f(i_{k_1+1}), \\ f(i_{k_1+1}) = f(i_{k_1+2}) = \cdots = f(i_{k_2}) &\neq f(i_{k_2+1}), \\ f(i_{k_2+1}) = f(i_{k_2+2}) = \cdots = f(i_{k_3}) &\neq f(i_{k_3+1}), \\ &\cdots \\ f(i_{k_p+1}) = f(i_{k_p+2}) = \cdots = f(i_{k_{p+1}}), \end{aligned}$$

where $k_{p+1} = s$ and set $k_0 = 1$.

For nodes $i_1, i_2, \cdots, i_{k_1}$, according to the definition of $f(\cdot)$ in *Lemma* 7, there is no direct connections among them, and hence the implementations A_{i_1}, A_{i_2}, $\cdots$, $A_{i_{k_1}}$ are independent and these implementations map the nodes from i_1, i_2, $\cdots$, k_1 to i_{k_1+1}, i_{k_1+2}, $\cdots$, k_2, which leads to $\mathcal{N}(A_{i_{k_0}}A_{i_{k_0+1}} \cdots A_{i_{k_1}}, \mathcal{S}) = \{i_{k_1+1}, i_{k_1+2}, \cdots, i_{k_2}\}$. Similarly, it holds

$$\begin{aligned} \mathcal{N}(A_{i_{k_1+1}}A_{i_{k_1+2}} \cdots A_{i_{k_2}}, \mathcal{S}) &= \{i_{k_2+1}, i_{k_2+2}, \cdots, i_{k_3}\}, \\ \mathcal{N}(A_{i_{k_2+1}}A_{i_{k_2+2}} \cdots A_{i_{k_3}}, \mathcal{S}) &= \{i_{k_3+1}, i_{k_3+2}, \cdots, i_{k_4}\}. \\ &\cdots \\ \mathcal{N}(A_{i_{k_p+1}}A_{i_{k_p+2}} \cdots A_{i_{k_{p+1}}}, \mathcal{S}) &= \partial^+(\mathcal{I}). \end{aligned}$$

Using the conductivity of the function $\mathcal{N}(\cdot, \cdot)$, one derives that $\mathcal{N}(A_{i_1}A_{i_2} \cdots A_{i_s}, \mathcal{S}) \subseteq \partial^+(\mathcal{I})$. Set $\sigma(k) = i_{s-k+1}$ and this completes the proof. □

Based on the above three lemmas, one obtains the proof of *Theorem* 2.

Proof of Theorem 2: The critical step of the proof is to find an algorithm which generates the asynchronous implementation sequence of A which cannot lead to consensus.

For the reason of simplicity, suppose that $\mathcal{V}$ can be partitioned equivalently into $r = 3$ components. The case of $r > 3$ can be proved similarly. Hence the reduced graph $\widetilde{\mathcal{G}} = (\widetilde{\mathcal{V}}, \widetilde{\mathcal{E}})$ satisfies $\widetilde{\mathcal{E}} = \{(1, 2), (2, 3), (3, 1)\}$. Without loss of generality, suppose $V_1 = \{1, 2, \cdots, s_1\}$, $V_2 = \{s_1 + 1, s_1 + 2, \cdots, s_2\}$, and $V_3 = \{s_2 + 1, s_2 + 2, \cdots, s_3\}$, where $s_3 = N$.

For the component V_1, based on the definition of essentially cyclic graph, one knows the subgraph $\mathcal{G}_{V_1}$ contains no cycle. According to *Lemma* 8, we can find a matrix $B_1 = A_{i_1}A_{i_2} \cdots A_{i_{s_1}}$ such that $\mathcal{N}(B_1, V_1) = \partial^+(V_1) \subseteq V_3$, where $\bigcup_{k=1}^{s_1} \{i_k\} = V_1$.

Similarly, one can construct two matrices $B_2 = A_{i_{s_1+1}}A_{i_{s_1+2}} \cdots A_{i_{s_2}}$ and $B_3 = A_{i_{s_2+1}}A_{i_{s_2+2}} \cdots A_{i_{s_3}}$ such that

$$\mathcal{N}(B_3, V_3) = \partial^+(V_3) \subseteq V_2, \ \mathcal{N}(B_2, V_2) = \partial^+(V_2) \subseteq V_1, \quad (7)$$

where $\bigcup_{k=s_1+1}^{s_2} \{i_k\} = V_2$ and $\bigcup_{k=s_2+1}^{s_3} \{i_k\} = V_3$.

According to the above equalities, one derives

$$\begin{aligned} &\mathcal{N}(B_1B_2B_3, V_1) \subseteq \mathcal{N}(B_2B_3, V_3) = \mathcal{N}(B_3, V_3) \subseteq V_2, \\ &\mathcal{N}(B_1B_2B_3, V_2) = \mathcal{N}(B_2B_3, V_2) \subseteq \mathcal{N}(B_3, V_1) \subseteq V_1, \\ &\mathcal{N}(B_1B_2B_3, V_3) = \mathcal{N}(B_2B_3, V_3) = \mathcal{N}(B_3, V_3) \subseteq V_2. \end{aligned}$$

According to *Lemma* 6, the above three equations imply that the matrix $B_1B_2B_3$ is not SIA, and hence repetitive products of $B_1B_2B_3$ cannot reach consensus. □

V. Discussions and Examples

According to *Theorem* 1 and 2, one knows the two sets of matrices $\mathcal{Q}_{\rm ps}$ and $\mathcal{Q}_{\rm ec}$ do not intersect with each other. Denote $\mathcal{Q}_{\rm SIA}$ as the set of SIA matrices, and an interesting question is whether $\mathcal{Q}_{\rm ps}$ and $\mathcal{Q}_{\rm ec}$ are complementary in $\mathcal{Q}_{\rm SIA}$, which is answered in the following proposition.

Proposition 8: It holds $\mathcal{Q}_{\rm ps}\bigcup(\mathcal{Q}_{\rm ec}\bigcap\mathcal{Q}_{\rm SIA})\subsetneq\mathcal{Q}_{\rm SIA}$.

Proof: Consider the matrix

$$A=\begin{pmatrix} 0 & 1/2 & 0 & 0 & 1/2\\ 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

and the corresponding graph $\mathcal{G}(A)$. One can easily check that $A\in\mathcal{Q}_{\rm SIA}$.

Since $\mathcal{G}(A)$ contains K_2 as a subgraph, then $A\notin\mathcal{Q}_{\rm ec}$ from *Proposition* 7.

In graph $\mathcal{G}(A)$, one finds that only node 1 and 3 share a common neighbor 2. If $A\in\mathcal{Q}_{\rm ps}$, then the absorbing set $\mathcal{I}$ can be set as $\mathcal{I}=\{1\}$ or $\mathcal{I}=\{3\}$. However, since neither node 1 nor node 3 contains a self-loop, $\{1\}$ and $\{3\}$ cannot be the absorbing sets, which is a contradiction and hence $A\notin\mathcal{Q}_{\rm ps}$.

Summarizing the above completes the proof. $\square$

In what follows, we will give some corollaries and examples of *Theorem* 1 and 2.

Corollary 1: Given a stochastic matrix $A\in\mathbb{R}^{N\times N}$, if $\mathcal{G}(A)$ is rooted with the diagonal entry corresponding to a root is positive, then $A\in\mathcal{Q}_{\rm ps}$.

Proof: Set $\mathcal{I}=\{\nu\}$, where $\nu\in{\rm root}(A)$ with the corresponding diagonal entry of ν is positive in A. One can check that all the conditions of partially scrambling matrices are satisfied. $\square$

Corollary 2: Given a stochastic matrix $A=(a_{ij})_{i,j=1}^{N}\in\mathbb{R}^{N\times N}$, if there exists $\mathcal{I}\subseteq\mathcal{V}$ such that

a) For each $i\in\mathcal{I}$, it holds $a_{ii}>0$;
b) For each $j\in\mathcal{V}/\mathcal{I}$ and $i\in\mathcal{V}$, there exists $k\in\mathcal{V}$ such that $a_{ik}a_{jk}>0$;
c) $\mathcal{G}(A)$ is rooted,

then $A\in\mathcal{Q}_{\rm ps}$.

Proof: If the root ν of $\mathcal{G}(A)$ belongs to $\mathcal{I}$, then $A\in\mathcal{Q}_{\rm ps}$ from *Corollary* 1. If the root of ν of $\mathcal{G}(A)$ belongs to $\mathcal{V}/\mathcal{I}$, then considering that set $\mathcal{V}$ is absorbing, A still belongs to $\mathcal{Q}_{\rm ps}$ from *Theorem* 1. $\square$

In the definition of asynchronous implementation in section II, each $\sigma(k)$ is only an element of set $\mathcal{V}$, and in fact, $\sigma(k)$ can be generalized to a subset of $\mathcal{V}$, which is called multiple asynchronous implementation defined below.

Multiple asynchronous implementation of DCA associated with a stochastic matrix A is defined as: for any sequence of matrices $\{A_{\sigma(k)}\}_{k=1}^{\infty}$ which satisfies $\sigma(k)\subseteq\mathcal{V}$, $\bigcup_{k=j}^{j+q-1}\sigma(k)=\mathcal{V}$ for all $j\geq 1$, it holds that

$$\lim_{k\to\infty}A_{\sigma(k)}\cdots A_{\sigma(2)}A_{\sigma(1)}x_1=\mathbb{1}\xi,$$

where $x_1\in\mathbb{R}^N$ and $\xi\in\mathbb{R}$ is decided by x_1 and the sequence $\{A_{\sigma(k)}\}_{k=1}^{\infty}$. The matrix $A_{\sigma(k)}$ $(\sigma(k)\subset\mathcal{V})$ is a direct generalization of $A_{\sigma(k)}$ $(\sigma(k)\in\mathcal{V})$ by preserving multiple rows $\sigma(k)$ of A in $A_{\sigma(k)}$.

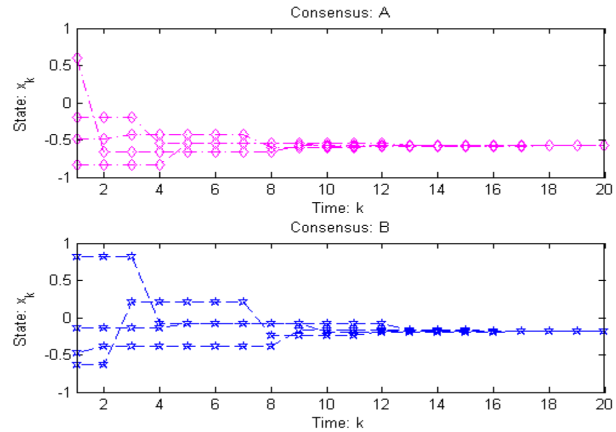

Fig. 4. Asynchronous implementation of matrices A and B given in (8).

Corollary 3: If $A\in\mathcal{Q}_{\rm ps}$, then any multiple asynchronous implementation of A guarantees consensus.

Proof: The proof of *Corollary* 3 requires a slight modification of *Lemma* 4, and we omit the details since the basic ideas of them are quite similar. $\square$

Since synchronous implementation is a special case of multiple asynchronous implementation (let $\sigma(k)=\mathcal{V}$ for each $k\geq 1$), one derives:

Corollary 4: If $A\in\mathcal{Q}_{\rm ps}$, then A is SIA.

Example 1: Given the two matrices

$$A=\begin{pmatrix} 0 & 0 & \frac{1}{2} & \frac{1}{2}\\ 0 & 0 & \frac{1}{2} & \frac{1}{2}\\ 0 & \frac{1}{2} & 0 & \frac{1}{2}\\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix},\ B=\begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2}\\ \frac{1}{2} & 0 & 0 & \frac{1}{2}\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0 \end{pmatrix}, \qquad (8)$$

the connectivity of $\mathcal{G}(A)$ and $\mathcal{G}(B)$ can be easily verified; however A is not scrambling since the third and fourth rows do not have positive entries in the same column. B is neither scrambling due to the same reason. For matrix A, set $\mathcal{I}=\{3,4\}$ and $\nu=1$, one can verify all the conditions of partially scrambling matrix are satisfied; for matrix B, set $\mathcal{I}=\{1,4\}$ and $\nu=2$, the conditions of partially scrambling matrix are also satisfied. According to *Theorem* 1, one knows that both A and B can be asynchronously implemented.

In order to verify *Theorem* 1, we choose $q=8$ and generate the indices $\sigma(4k+4),\sigma(4k+3),\sigma(4k+2),\sigma(4k+1)$, for each $k\geq 0$ via the following procedure:

a) Set $k:=0$;
b) Set $\sigma(4k+j)=j$ for each $j=1,2,3,4$;
c) Randomly choose two elements among $\sigma(4k+4)$, $\sigma(4k+3)$, $\sigma(4k+2)$, $\sigma(4k+1)$, and swap their positions;
d) Repeat c) for 5 times;
e) Set $k:=k+1$ and go to b).

The above procedure guarantees $\bigcup_{k=j}^{j+q-1}\{\sigma(k)\}=\mathcal{V}$ for each $k\geq 1$. Given two sets of random initial values, the corresponding asynchronous dynamics of x_k defined in (2) with respect to A and B are given in Fig. 4, and one can see both of them realize consensus.

Example 2: As shown in Fig. 5, the graph on the left is a cyclic graph with period 3. Then adding two edges within two clusters generates an aperiodic graph. One can check that all the conditions of *Theorem* 2 are satisfied, and then any

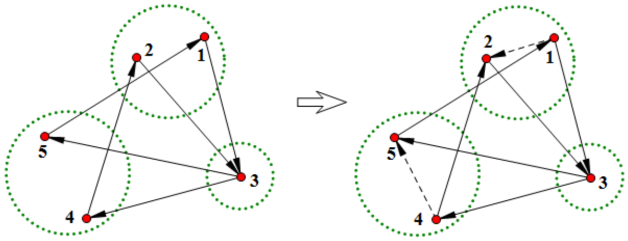

Fig. 5. Example 2: adding edges within clusters.

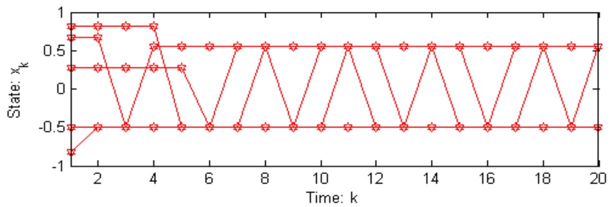

Fig. 6. Asynchronous implementation of matrix A given in (9).

stochastic matrix associated with the graph on the right cannot be asynchronously implemented.

Given the following stochastic matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}, \tag{9}$$

whose graph $\mathcal{G}(A)$ is on the right of Fig. 5. According to the proof of *Theorem* 2, one can construct the following periodic indices

$$\sigma(k) = \begin{cases} 4, & \text{when } k \equiv 1(\text{mod}5), \\ 5, & \text{when } k \equiv 2(\text{mod}5), \\ 3, & \text{when } k \equiv 3(\text{mod}5), \\ 1, & \text{when } k \equiv 4(\text{mod}5), \\ 2, & \text{when } k \equiv 0(\text{mod}5). \end{cases}$$

Given a set of random initial values, the dynamics of x_k driven by the above $\sigma(k)$ are given in Fig. 6, and one can see that such an implementation does not realize consensus.

VI. Conclusion

This paper has discussed two problems on asynchronous implementation of DCA: what type of stochastic matrices can be asynchronously implemented, and what type cannot. We have found two types of stochastic matrices, called partially scrambling and essentially cyclic matrices, based on which we have proved that any partially scrambling matrix can be asynchronously implemented, while any essentially cyclic matrix cannot. Since the identified two types of stochastic matrices are not complementary, our future research will focus on identifying the maximal subclass of SIA matrices in which any asynchronous implementation sequence of each realizes consensus.

References

[1] S. Bolouki and R. P. Malhame, "Consensus algorithms and the decomposition-separation theorem," *Proc. 52nd IEEE Conference on Decision and Control*, Florence, Italy, pp. 1490-1495, 2013.

[2] S. Bolouki and R. P. Malhame, "Ergodicity and class-ergodicity of balanced asymmetric stochastic chains," *Proc. 2013 European Control Conference*, Zürich, Switzerland, pp. 221-226, 2013.

[3] M. Cao, A. S. Morse, and B. D. O. Anderson, "Agreeing asynchronously," *IEEE Trans. Automat. Contr.*, vol. 53, no. 8, pp. 1826-1838, Aug. 2008.

[4] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: a graphical approach," *SIAM J. Contr. & Optim.*, vol. 47, no. 2, pp. 575-600, 2008.

[5] Y. Chen, J. Lü, F. Han, and X. Yu, "On the cluster consensus of discrete-time multi-agent systems," *Systems & Control Letters*, vol. 60, no. 7, pp. 517-523, 2011.

[6] Y. Chen, D. Ho, J. Lü, and Z. Lin, "Convergence rate for discrete-time multiagent systems with time-varying delays and general coupling coefficients," *IEEE Trans. Neural. Net. & Learn. Sys.*, doi: 10.1109/TNNLS.2015.2473690

[7] Y. Chen, W. Xiong, and F. Li, "Convergence of infinite products of stochastic matrices: a graphical decomposition criterion," *IEEE Trans. Auto. Contr.*, doi: 10.1109/TAC.2016.2521782

[8] Y. Chen, J. Lü, X. Yu, and Z. Lin, "Consensus of discrete-time second-order multiagent systems based on infinite products of general stochastic matrices," *SIAM J. Contr. Optim.*, vol. 51, no. 4, pp. 3274-3301, 2013.

[9] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118-121, 1974.

[10] J. Hajnal, "Weak ergodicity in non-homogeneous Markov chains," *Proc. Camb. Phil. Soc.*, vol. 54, pp. 233-246, 1958.

[11] D. J. Hartfiel, "Nonhomogeneous matrix products," Singapore: World Scientific, 2002.

[12] J. M. Hendrickx and J. Tsitsiklis, "Convergence of type-symmetric and cut-balanced consensus seeking system," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 214-218, 2013.

[13] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Contr.*, vol. 48, no. 6, pp. 988-1001, Jun. 2003.

[14] G. Li, Z. Zhu, Z. Cong, and F. Yang, "Efficient decomposition of strongly connected components on GPUs," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 1-10, 2014.

[15] J. Liu, A. S. Morse, B. D. O. Anderson, and C. Yu, "Contractions for consensus processes," *Proc. 50th Conference on Decision & Control & European Control Conference (CDC-ECC)*, Orlando, FL, USA, pp. 1974-1979, 2011.

[16] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Automat. Contr.*, vol. 60, no. 3, pp. 601-615, 2015.

[17] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215-233, Jan. 2007.

[18] A. Olshevsky and J. N. Tsitsiklis, "On the nonexistence of quadratic Lyapunov function for consensus algorithms," *IEEE Trans. Automat. Contr.*, vol. 53, no. 11, pp. 2642-2645, 2008.

[19] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automat. Contr.*, vol. 50, no. 5, pp. 655-661, May 2005.

[20] T. A. Sarymsakov, "Inhomogeneous Markov chians (in russian)," *Teor. Verojatnost. i Primen.*, vol. 6, pp. 194-201, 1961.

[21] E. Seneta, *Nonnegative Matrices and Markov Chains*. Berlin: Springer, 2006.

[22] B. Touri and A. Nedić, "On backward product of stochastic matrices," *Automatica*, vol. 48, no. 8, pp. 1477-1488, 2012.

[23] B. Touri and A. Nedić, "Product of random stochastic matrices," *IEEE Trans. Automat. Contr.*, vol. 59, no. 2, pp. 437-448, 2014.

[24] J. N. Tsitsiklis and V. D. Blondel, "The Lyapunov exponent and joint spectral radius of pairs of matrices are hard - when not impossible - to compute and to approximate," *Math. of Contr. Sign. & Sys.*, vol. 10, pp. 31-40, 1997.

[25] J. Wolfowitz, "Products of indepensable, aperiodic, stochastic matrices," *Proc. Amer. Math. Soc.*, vol. 14, pp. 733-737, 1963.

[26] W. Xia and M. Cao, "Sarymsakov matrices and asynchronous implementation of distributed coordination algorithms," *IEEE Trans. Automat. Contr.*, vol. 59, no. 8, pp. 2228-2233, 2014.

[27] W. Xia, M. Cao, and K. H. Johansson, "Structural balance and opinion separation in trust-mistrust social networks," *IEEE Trans. Contr. Net. Sys.* vol. 3, no. 1, pp. 46-56, 2016.

[28] W. Xia, J. Liu, M. Cao, K. H. Johansson, and T. Başar, "Products of generalized stochastic sarymsakov matrices", In *Proc. 50th Conference on Decision & Control*, Osaka, Japan, pp. 3621-3626, 2015.

[29] F. Xiao and L. Wang, "Consensus protocols for discrete-time multi-agent systems with time-varying delays," *Automatica*, vol. 44, no. 10, pp. 2577-2582, 2008.

[30] W. Yu, G. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 95, no. 6, pp. 1089-1095, 2010.