

# Characterizing Convergence Speed for Consensus Seeking over Dynamically Switching Directed Random Networks

Jing Zhou and Qian Wang

**Abstract**—Characterizing convergence speed is one of the important research challenges in the design of distributed consensus algorithms for networked multi-agent systems. In this paper, we consider a group of agents that communicate via a dynamically switching directed random network. Each link in the network, which represents the directed information flow between any ordered pair of agents, could be subject to failure with certain probability. Hence we model the information flow using dynamic random digraphs. We characterize the convergence speed for the distributed discrete-time consensus algorithm over a variety of random networks with arbitrary weights. In particular, we propose the per-step (mean square) convergence factor as a measure of the convergence speed and derive the exact value for this factor. Numerical examples are also given to illustrate our theoretical results.

## I. INTRODUCTION

A series of work have been conducted on convergence analysis for distributed consensus algorithms in the literature. Convergence conditions have been derived for both stationary network topologies and switching network topologies in [12], [23], [16], [21], [22], [6], [18]; all focus on deterministic networks. Consensus over randomly switching network topologies has been studied in [9], [10], [19] for undirected information flow and in [26], [20] for directed information flow.

Previous work on analysis of convergence speed for consensus algorithms includes [18], [27] for stationary network topologies, [18], [2] for deterministically switching network topologies, and [9], [19], [13], [14], [25] for randomly switching network topologies. Note that both [9] and [19] have focused on undirected networks and there does not exist counterpart work on directed random networks in current literature. Theoretically, problems on directed graphs are more challenging than those on undirected graphs due to the fact that algebraic properties are mostly known for undirected graphs while little is known for directed graphs. Also, practically, many important real networks have directed edges.

As a complementary to [9] and [19], this paper mainly focuses on directed random networks with arbitrary edge weights and we give an analytical lower bound for the convergence speed of a discrete-time consensus algorithm. Particular contributions of this paper are listed as follows:

- We consider directed random networks rather than undirected random networks as in [9] and [19]. Further,

This work is supported in part by NSF under Grant 0409184.

Jing Zhou is a PhD candidate of Mechanical Engineering, Penn State University, University Park, PA 16802, USA [juz119@psu.edu](mailto:juz119@psu.edu)

Qian Wang is with Mechanical Engineering, Penn State University, University Park, PA 16802, USA [quw6@psu.edu](mailto:quw6@psu.edu)

we consider networks where different edges could have different edge probabilities and weights, rather than assume that each edge has 0-1 weight and all edges share a common edge probability as in [9], [19]. In addition, the weight matrix of a network may have negative elements, and thus classical results on stochastic matrices [3], [4], [24] are no longer applicable.

- In [19], a decay factor of the consensus algorithm is characterized in terms of the spectral radius of an augmented matrix derived from a Lyapunov-like matrix recursion via Kronecker product. In comparison, we characterize the convergence speed in terms of the spectral radius of a Laplacian matrix in an explicit form.

The paper is organized as follows. In Section 2, the problem formulation is presented. In Section 3, we derive the convergence speed for the discrete-time consensus algorithm over a variety of directed random networks. In Section 4, numerical examples are presented to illustrate our theoretical results. Conclusions are drawn in Section 5.

## II. PROBLEM STATEMENT

Let  $\mathbf{1} \in \mathbb{R}^n$  denote a column vector with all elements equal to one, i.e.,  $\mathbf{1} = [1 \cdots 1]^T$ . Let  $I_n$  denote the  $n \times n$  identity matrix. Let  $\|\cdot\|_2$  denote the standard Euclidean norm.

### A. Properties of random digraphs

We model a dynamically switching directed random network by a sequence of weighted random digraphs  $G = (V, E, W)$  where

- $V = \{v_i, i = 1, 2, \dots, n\}$  is a node (or vertex) set whose elements denote the agents in the network.
- $E \subseteq V \times V$  is an edge set whose elements denote the communication links between agents.  $E$  is a set of ordered pairs of nodes, i.e.,  $(v_i, v_j)$  and  $(v_j, v_i)$  are two different elements of  $E$ .  $(v_i, v_j)$  is an element of  $E$  if and only if agent  $v_i$  receives information from agent  $v_j$ , in which case  $v_i$  is referred to as the *child node* and  $v_j$  is referred to as the *parent node*.  $v_j$  is also referred to as a *neighbor* of  $v_i$  in this case. We assume that the existence of each edge is determined randomly and independently of other edges with the *edge probability*  $p_{ij}$  which satisfies  $0 \leq p_{ij} \leq 1$  for  $i, j = 1, \dots, n, j \neq i$ .  $p_{ij} = 1$  means that the edge  $(v_i, v_j)$  always exists while  $p_{ij} = 0$  means that the edge  $(v_i, v_j)$  never exists. An information link is referred to as a *potential link* when the associated edge probability  $p_{ij} > 0$ , which means that the link may fail with the probability  $1 - p_{ij}$ . We also define an *edge probability matrix*  $P = [p_{ij}]$ . In this

paper, we do not consider self-loops or multiple edges. Thus, we have  $p_{ii} = 0$  for  $i = 1, \dots, n$ . We then define  $n(n-1)$  independent Bernoulli random variables,  $\delta_{ij}$ 's,  $i, j = 1, \dots, n, j \neq i$ , as follows

$$\delta_{ij} = \begin{cases} 1, & \text{with probability } p_{ij} \\ 0, & \text{with probability } 1 - p_{ij} \end{cases} \quad (1)$$

where each random variable  $\delta_{ij}$  is associated with a directed edge  $(v_i, v_j)$ .

- $W = [w_{ij}] \in \mathbb{R}^{n \times n}$  is a weight matrix with all diagonal elements equal to 0, and each element  $w_{ij}$  denotes the weight associated with the edge  $(v_i, v_j)$ . The weight denotes how each agent evaluates the information collected from its neighboring agents to update the consensus algorithm. In general, we do not require  $W$  to have nonnegative elements or to be symmetric. A negative weight is often used to imply deteriorated communication channels, or natural disagreement of the child node over the information obtained from its parent node [20].

Algebraically, a weighted random digraph  $G$  is represented by an adjacency matrix  $A = [a_{ij}]$  and a Laplacian matrix  $L = [l_{ij}]$  defined as follows:

$$a_{ij} = \begin{cases} 0, & \text{if } i = j \\ w_{ij}\delta_{ij}, & \text{if } i \neq j \end{cases} \quad (2)$$

where  $w_{ij}$  is the corresponding entry of the weight matrix  $W$  and

$$l_{ij} = \begin{cases} \sum_{k=1, \dots, n} a_{ik}, & \text{if } i = j \\ -a_{ij}, & \text{if } i \neq j \end{cases} \quad (3)$$

Both the weighted adjacency matrix and the Laplacian matrix defined above are essentially random. Let  $\mathbf{G}$ ,  $\mathbf{A}$  and  $\mathbf{L}$  denote the sample spaces of all random digraphs, all adjacency matrices and all Laplacian matrices respectively. A directed graph with the Laplacian matrix  $L$  is said to be *balanced* if  $\mathbf{1}^T L = 0^T$ . Next we introduce  $n(n-1)$  constant matrices  $\{B_{ij}, i = 1, \dots, n-1, j = i+1, \dots, n\}$ :

$$B_{ij} = w_{ij} \mathbf{e}_i (\mathbf{e}_i - \mathbf{e}_j)^T \quad (4)$$

where  $\mathbf{e}_i \in \mathbb{R}^n$  denotes the vector with the  $i$ th entry equal to 1 and all other entries equal to 0. Note that  $B_{ij}$  is the Laplacian of a directed graph with only one edge  $(v_i, v_j)$ . Then the Laplacian  $L$  can be represented as follows,

$$L = \sum_{i=1}^n \sum_{j \neq i, j=1}^n \delta_{ij} B_{ij}. \quad (5)$$

Let  $L_c = [l_{ij}^c]$  denote the expected value of the Laplacian. Then,

$$\begin{aligned} L_c &\triangleq \mathbf{E}[L] = \mathbf{E}\left[\sum_{i=1}^n \sum_{j \neq i, j=1}^n \delta_{ij} B_{ij}\right] = \sum_{i=1}^n \sum_{j \neq i, j=1}^n \mathbf{E}[\delta_{ij}] B_{ij} \\ &= \sum_{i=1}^n \sum_{j \neq i, j=1}^n p_{ij} B_{ij} \end{aligned} \quad (6)$$

In another word,  $L_c = [l_{ij}^c]$  denotes the Laplacian of the directed graph  $G_c \in \mathbf{G}$  where all the edges with positive

edge probabilities exist. The elements of  $L_c$  are represented as follows,

$$l_{ij}^c = \begin{cases} \sum_{k=1}^n w_{ik} p_{ik}, & \text{if } i = j \\ -w_{ij} p_{ij}, & \text{if } i \neq j \end{cases} \quad (7)$$

We also define a constant matrix  $\tilde{L}_c$  that will be used in the derivation of convergence speed for a consensus algorithm,

$$\tilde{L}_c = \sum_{i=1}^n \sum_{j \neq i, j=1}^n (p_{ij} - p_{ij}^2) B_{ij}^T B_{ij} \quad (8)$$

Note that  $\tilde{L}_c = [\tilde{l}_{ij}^c]$  is a symmetric Laplacian matrix with entries defined as follows,

$$\tilde{l}_{ij}^c = \begin{cases} \sum_{k=1}^n (w_{ik}^2 (p_{ik} - p_{ik}^2) + w_{ki}^2 (p_{ki} - p_{ki}^2)), & \text{if } i = j \\ -(w_{ij}^2 (p_{ij} - p_{ij}^2) + w_{ji}^2 (p_{ji} - p_{ji}^2)), & \text{if } i \neq j \end{cases} \quad (9)$$

When all the edge probabilities are equal, i.e.,  $p_{ij} = p$  for  $i, j = 1, \dots, n, j \neq i$ , we have

$$L_c = p \sum_{i=1}^n \sum_{j \neq i, j=1}^n B_{ij}, \quad \tilde{L}_c = (p - p^2) \sum_{i=1}^n \sum_{j \neq i, j=1}^n B_{ij}^T B_{ij}$$

**Remark 1:** For any Laplacian matrix  $L$ , 0 is its eigenvalue with the corresponding right eigenvector  $\mathbf{1}$ , i.e.,  $L\mathbf{1} = \mathbf{0}$ . For a graph with arbitrary weights, its Laplacian matrix  $L$  is not guaranteed to be positive semi-definite.

### B. Consensus Algorithm

Consider a dynamical system of  $n$  agents that communicate via a dynamically switching directed random network. Let  $x_i(k)$  denote the state of agent  $i$  at time instant  $k$ ,  $i = 1, \dots, n$ . For simplicity,  $x_i(k)$  is assumed to be a scalar. Extension to vector states can be made by using Kronecker product and thus is omitted here. Let  $\mathbf{x}(k) = [x_1(k), \dots, x_n(k)]^T$  denote the state vector of all agents. At each time instant, each agent updates its own state based on the local information collected from its neighboring agents according to a distributed control law, which is referred to as the distributed consensus algorithm.

In this paper, we consider the following discrete-time consensus algorithm [19],

$$x_i(k+1) = x_i(k) + \sum_{j \in \mathcal{N}(i,k)} w_{ij} (x_j(k) - x_i(k)) \quad (10)$$

where  $w_{ij}$  is the corresponding element of the weight matrix  $W$ ;  $\mathcal{N}(i,k)$  denotes the set of neighboring agents of agent  $i$  at time instant  $k$ . Then the overall system of all agents is given as,

$$\mathbf{x}(k+1) = (I_n - L(k))\mathbf{x}(k) \quad (11)$$

where  $L(k)$  is the Laplacian of the random graph at time instant  $k$ . Due to the randomness of the underlying network, the Laplacian  $L(k)$  is a random matrix and the system (11) is a stochastic system, which makes  $\mathbf{x}(k)$  a random vector and  $\{\mathbf{x}(k)\}$  a random sequence.

*Definition 1 (agreement set):* The agreement set  $\mathcal{A} \subseteq \mathbb{R}^n$  is defined as the subspace spanned by  $\{\mathbf{1}\}$ , i.e.,

$$\mathcal{A} = \{c\mathbf{1} : c \in \mathbb{R}\}.$$

Note that  $\mathcal{A}$  is the equilibrium set of the dynamical system (11), i.e.,  $\forall \mathbf{x}(K_0) = c\mathbf{1} \in \mathcal{A}$ , for all  $k \geq K_0$  we have  $\mathbf{x}(k) = c\mathbf{1}$  since  $\mathbf{x}(K_0 + 1) = (I_n - L(K_0))\mathbf{x}(K_0) = c(I_n - L(K_0))\mathbf{1} = c\mathbf{1}$  and recursively.

*Definition 2 (almost sure consensus):* The discrete-time consensus algorithm (11) asymptotically reaches *almost sure consensus* if the equilibrium set  $\mathcal{A}$  is almost surely stable and if the sequence  $\{\mathbf{x}(k)\}$  asymptotically converges to the agreement set  $\mathcal{A}$  almost surely, i.e., for each  $\varepsilon > 0$ ,

$$\lim_{k_0 \rightarrow \infty} \mathbf{P} \left\{ \sup_{k \geq k_0} \inf_{\mathbf{x}^* \in \mathcal{A}} \|\mathbf{x}(k) - \mathbf{x}^*\|_2 > \varepsilon \right\} = 0 \quad (12)$$

Almost sure consensus is also called *consensus with probability one (w.p.1)*.

Let  $Q = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_{n-1}] \in \mathbb{R}^{n \times (n-1)}$  denote a matrix such that  $\frac{1}{\sqrt{n}}\mathbf{1}$  and  $\mathbf{q}_i$ 's ( $i = 1, \dots, n-1$ ) form an orthonormal basis of  $\mathbb{R}^n$ .  $Q$  is not unique; however, it is true that  $Q^T \mathbf{1} = \mathbf{0}$ ,  $Q^T Q = I_{n-1}$  and  $QQ^T = I_n - \frac{1}{n}(\mathbf{1}\mathbf{1}^T)$ . Similar to [20], we define the *error vector*  $\xi(k)$  as follows,

$$\xi(k) = Q^T \mathbf{x}(k), \quad (13)$$

The state vector  $\mathbf{x}(k)$  can then be decomposed as,

$$\mathbf{x}(k) = Q\xi(k) + \frac{1}{n}(\mathbf{1}^T \mathbf{x}(k))\mathbf{1}. \quad (14)$$

*Lemma 1:*  $\inf_{\mathbf{x}^* \in \mathcal{A}} \|\mathbf{x}(k) - \mathbf{x}^*\|_2 = \|\xi(k)\|_2$ .

*Proof:* Given any  $c\mathbf{1} \in \mathcal{A}$  where  $c$  is a constant number,

$$\begin{aligned} & \|\mathbf{x}(k) - c\mathbf{1}\|_2^2 \\ &= \mathbf{x}^T(k)\mathbf{x}(k) - 2c\mathbf{1}^T \mathbf{x}(k) + c^2 n \\ &= n \left[ c - \frac{1}{n} \mathbf{1}^T \mathbf{x}(k) \right]^2 + \mathbf{x}^T(k)\mathbf{x}(k) - \frac{1}{n}(\mathbf{1}^T \mathbf{x}(k))^2 \end{aligned}$$

and  $\|\mathbf{x}(k) - c\mathbf{1}\|_2^2$  is minimized when  $c = \frac{1}{n} \mathbf{1}^T \mathbf{x}(k)$ , i.e.,  $\inf_{\mathbf{x}^* \in \mathcal{A}} \|\mathbf{x}(k) - \mathbf{x}^*\|_2 = \sqrt{\mathbf{x}^T(k)\mathbf{x}(k) - \frac{1}{n}(\mathbf{1}^T \mathbf{x}(k))^2}$ . On the other hand,  $\|\xi(k)\|_2^2 = \mathbf{x}^T(k)QQ^T \mathbf{x}(k) = \mathbf{x}^T(k)(I_n - \frac{1}{n}(\mathbf{1}\mathbf{1}^T))\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{x}(k) - \frac{1}{n}(\mathbf{1}^T \mathbf{x}(k))^2$ . ■

Therefore, (12) is equivalent to

$$\lim_{k_0 \rightarrow \infty} \mathbf{P} \left\{ \sup_{k \geq k_0} \|\xi(k)\|_2 > \varepsilon \right\} = 0. \quad (15)$$

The discrete-time consensus algorithm (11) asymptotically reaches almost sure consensus if and only if the equilibrium point  $\mathbf{0}$  is almost surely asymptotically stable for the error vector  $\xi(k)$ . Hence, we characterize the convergence speed of almost sure consensus by computing the convergence speed of the error vector  $\xi(k)$ .

### C. Measures of convergence speed

Per-step convergence factor has been defined in [27] as a measure of convergence speed for consensus algorithms over fixed, undirected network topologies. In this paper, since we consider directed random networks, we modify the definition of this measure in a probabilistic setting as follows:

*Definition 3 (per-step (mean square) convergence factor/time):* The per-step convergence factor, in the mean square sense, for (11) is defined as

$$r_s = \sup_{\xi(k) \neq \mathbf{0}, k \in \mathbf{Z}} \frac{\mathbf{E}[\|\xi(k+1)\|_2^2 | \xi(k)]}{\|\xi(k)\|_2^2}, \quad (16)$$

and the associated per-step convergence time is defined as

$$\tau_s = \frac{1}{\log(1/r_s)}. \quad (17)$$

*Lemma 2:*  $r_s < 1$  implies almost sure consensus.

*Proof:* From (16), we have

$$\mathbf{E}[\|\xi(k+1)\|_2^2 - \|\xi(k)\|_2^2 | \xi(k)] \leq -(1 - r_s)\|\xi(k)\|_2^2.$$

When  $r_s < 1$ ,  $1 - r_s > 0$ . According to [15] (Theorem 2 on pp. 197), we conclude that the equilibrium point  $\mathbf{0}$  is exponentially stable in the mean square for the error vector  $\xi(k)$ . According to [17], exponential stability in the mean square implies almost sure asymptotic stability. ■

## III. CONVERGENCE AND CONVERGENCE SPEED

In this section, we derive the the per-step (mean square) convergence factor for the discrete-time consensus algorithm (11) over a dynamically switching directed random network. Before proceeding with the final theorem for convergence speed, we first give several facts that will be used in the proof.

*Proposition 1:* The evolution of the error vector for the discrete-time consensus algorithm (11) is given by the following recursive equation,

$$\xi(k+1) = Q^T(I_n - L(k))Q\xi(k) \quad (18)$$

*Proof:* According to (11) and (13),

$$\xi(k+1) = Q^T \mathbf{x}(k+1) = Q^T(I_n - L(k))\mathbf{x}(k).$$

Then by (14), we have

$$\xi(k+1) = Q^T(I_n - L(k))(Q\xi(k) + \frac{1}{n}(\mathbf{1}^T \mathbf{x}(k))\mathbf{1}).$$

Since  $Q^T \mathbf{1} = \mathbf{0}$  and  $L(k)\mathbf{1} = \mathbf{0}$ , we further have

$$\xi(k+1) = Q^T(I_n - L(k))Q\xi(k). \quad \blacksquare$$

*Lemma 3:*  $B_{ij}^T Q Q^T B_{ij} = (1 - \frac{1}{n})B_{ij}^T B_{ij}$ .

*Proof:* Since  $Q Q^T = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ , we have  $B_{ij}^T Q Q^T B_{ij} = B_{ij}^T B_{ij} - \frac{1}{n}B_{ij}^T \mathbf{1}\mathbf{1}^T B_{ij}$  where

$$\begin{aligned} B_{ij}^T \mathbf{1}\mathbf{1}^T B_{ij} &= (w_{ij}(\mathbf{e}_i - \mathbf{e}_j)\mathbf{e}_i^T)\mathbf{1}\mathbf{1}^T(w_{ij}\mathbf{e}_i(\mathbf{e}_i - \mathbf{e}_j)^T) \\ &= w_{ij}^2(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \\ &= w_{ij}^2(\mathbf{e}_i - \mathbf{e}_j)\mathbf{e}_i^T \mathbf{e}_i(\mathbf{e}_i - \mathbf{e}_j)^T \\ &= B_{ij}^T B_{ij} \end{aligned}$$

Hence,

$$B_{ij}^T Q Q^T B_{ij} = (1 - \frac{1}{n})B_{ij}^T B_{ij}. \quad \blacksquare$$

Next we derive the explicit form for per-step (mean square) convergence factor over directed random networks.

*Proposition 2:* The conditional variance of the error vector for the discrete-time consensus algorithm (11) over a dynamically switching directed random network is

$$\begin{aligned} & \mathbf{E}[\|\xi(k+1)\|_2^2 | \xi(k)] \\ &= (\mathcal{Q}\xi(k))^T [(I_n - L_c)^T R(I_n - L_c) + (1 - \frac{1}{n})\tilde{L}_c] \\ & \quad \times \mathcal{Q}\xi(k) \end{aligned} \quad (19)$$

where  $R \triangleq I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ .

*Proof:* By Proposition 1,

$$\begin{aligned} & \mathbf{E}[\|\xi(k+1)\|_2^2 | \xi(k)] \quad (20) \\ &= \mathbf{E}[\xi^T(k+1)\xi(k+1) | \xi(k)] \\ &= \xi^T(k) \mathbf{E}[\mathcal{Q}^T(I_n - L^T(k))\mathcal{Q}\mathcal{Q}^T(I_n - L(k))\mathcal{Q}] \xi(k) \\ &= \xi^T(k) (I_{n-1} - \mathcal{Q}^T(\mathbf{E}[L^T(k)] + \mathbf{E}[L(k)])\mathcal{Q}) \\ & \quad + \mathbf{E}[\mathcal{Q}^T L^T(k)\mathcal{Q}\mathcal{Q}^T L(k)\mathcal{Q}] \xi(k) \end{aligned} \quad (21)$$

Define the index sets  $I_1 = \{i, j = 1, \dots, n, j \neq i\}$ ,  $I_2 = \{i, j, s, l = 1, \dots, n, j \neq i, l \neq s\}$  and  $I_3 = \{i, j, s, l = 1, \dots, n, j \neq i, s = i, l = j\}$ . Then by (5),

$$\begin{aligned} & \mathbf{E}[\mathcal{Q}^T L^T(k)\mathcal{Q}\mathcal{Q}^T L(k)\mathcal{Q}] \\ &= \sum_{I_2} \mathbf{E}[\delta_{ij}\delta_{sl}] \mathcal{Q}^T B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ &= (\sum_{I_2 \setminus I_3} + \sum_{I_3}) \mathbf{E}[\delta_{ij}\delta_{sl}] \mathcal{Q}^T B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ &= \sum_{I_2 \setminus I_3} \mathbf{E}[\delta_{ij}\delta_{sl}] \mathcal{Q}^T B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ & \quad + \sum_{I_1} \mathbf{E}[\delta_{ij}^2] \mathcal{Q}^T B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{ij} \mathcal{Q} \end{aligned}$$

Since  $\delta_{ij}$ 's are independent Bernoulli random variables with success probability  $p_{ij}$ , we have  $\mathbf{E}[\delta_{ij}^2] = p_{ij}$  for  $i, j = 1, \dots, n, j \neq i$  and  $\mathbf{E}[\delta_{ij}\delta_{sl}] = p_{ij}p_{sl}$  for  $i, j, s, l = 1, \dots, n, j \neq i, l \neq s, s \neq i, l \neq j$ . Hence,

$$\begin{aligned} & \mathbf{E}[\mathcal{Q}^T L^T(k)\mathcal{Q}\mathcal{Q}^T L(k)\mathcal{Q}] \\ &= \mathcal{Q}^T \left( \sum_{I_2 \setminus I_3} p_{ij}p_{sl} B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{sl} + \sum_{I_1} p_{ij} B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{ij} \right) \mathcal{Q} \\ &= \sum_{I_2 \setminus I_3} p_{ij}p_{sl} \mathcal{Q}^T (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ & \quad + \mathcal{Q}^T \left( \sum_{I_1} (p_{ij} - p_{ij}^2 + p_{ij}^2) B_{ij}^T \mathcal{Q}\mathcal{Q}^T B_{ij} \right) \mathcal{Q} \end{aligned}$$

By the definitions of the index sets,  $\sum_{I_1} p_{ij}^2 (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{ij} = \sum_{I_3} p_{ij}p_{sl} (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{sl}$ . Thus,

$$\begin{aligned} & \mathbf{E}[\mathcal{Q}^T L^T(k)\mathcal{Q}\mathcal{Q}^T L(k)\mathcal{Q}] \\ &= \sum_{I_2 \setminus I_3} p_{ij}p_{sl} \mathcal{Q}^T (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ & \quad + \mathcal{Q}^T \left( \sum_{I_1} (p_{ij} - p_{ij}^2) (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{ij} \right) \mathcal{Q} \end{aligned}$$

$$\begin{aligned} & + \sum_{I_3} p_{ij}p_{sl} \mathcal{Q}^T (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ &= \sum_{I_2} p_{ij}p_{sl} \mathcal{Q}^T (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{sl} \mathcal{Q} \\ & \quad + \mathcal{Q}^T \left( \sum_{I_1} (p_{ij} - p_{ij}^2) (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{ij} \right) \mathcal{Q} \\ &= \mathcal{Q}^T \left( \sum_{i=1}^n \sum_{j \neq i, j=1}^n p_{ij} (B_{ij})^T \right) \mathcal{Q}\mathcal{Q}^T \left( \sum_{s=1}^n \sum_{l \neq s, l=1}^n p_{sl} B_{sl} \right) \mathcal{Q} \\ & \quad + \mathcal{Q}^T \left( \sum_{i=1}^n \sum_{j \neq i, j=1}^n (p_{ij} - p_{ij}^2) (B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{ij} \right) \mathcal{Q}. \end{aligned}$$

By Lemma 3, we have  $(B_{ij})^T \mathcal{Q}\mathcal{Q}^T B_{ij} = (1 - \frac{1}{n}) B_{ij}^T B_{ij}$ . Therefore,

$$\begin{aligned} & \mathbf{E}[\mathcal{Q}^T L^T(k)\mathcal{Q}\mathcal{Q}^T L(k)\mathcal{Q}] \\ &= \mathcal{Q}^T \left( \sum_{i=1}^n \sum_{j \neq i, j=1}^n p_{ij} (B_{ij})^T \right) \mathcal{Q}\mathcal{Q}^T \left( \sum_{s=1}^n \sum_{l \neq s, l=1}^n p_{sl} B_{sl} \right) \mathcal{Q} \\ & \quad + (1 - \frac{1}{n}) \mathcal{Q}^T \left( \sum_{i=1}^n \sum_{j \neq i, j=1}^n (p_{ij} - p_{ij}^2) B_{ij}^T B_{ij} \right) \mathcal{Q} \\ &= (\mathcal{Q}^T (L_c)^T \mathcal{Q})(\mathcal{Q}^T L_c \mathcal{Q}) + (1 - \frac{1}{n}) \mathcal{Q}^T \tilde{L}_c \mathcal{Q} \end{aligned} \quad (22)$$

By (20), (6), and (22), we have

$$\begin{aligned} & \mathbf{E}[\|\xi(k+1)\|_2^2 | \xi(k)] \\ &= \xi^T(k) [I_{n-1} - \mathcal{Q}^T (L_c^T + L_c) \mathcal{Q}] \xi(k) + (\mathcal{Q}^T L_c^T \mathcal{Q})(\mathcal{Q}^T L_c \mathcal{Q}) \\ & \quad + (1 - \frac{1}{n}) \mathcal{Q}^T \tilde{L}_c \mathcal{Q} \xi(k) \\ &= \xi^T(k) [\mathcal{Q}^T (I_n - L_c^T) \mathcal{Q}\mathcal{Q}^T (I_n - L_c) \mathcal{Q}] \\ & \quad + (1 - \frac{1}{n}) \mathcal{Q}^T \tilde{L}_c \mathcal{Q} \xi(k) \\ &= (\mathcal{Q}\xi(k))^T [(I_n - L_c)^T R(I_n - L_c) + (1 - \frac{1}{n})\tilde{L}_c] \mathcal{Q}\xi(k) \end{aligned}$$

*Theorem 1:* The per-step (mean square) convergence factor for the discrete-time consensus algorithm (11) over a dynamically switching directed random network is

$$r_s = \rho_d \triangleq \rho \left( (I_n - L_c)^T R(I_n - L_c) + (1 - \frac{1}{n})\tilde{L}_c \right) \quad (23)$$

where  $\rho(\cdot)$  denotes the spectral radius of a matrix.

*Proof:* From Proposition 2,

$$\begin{aligned} & \mathbf{E}[\|\xi(k+1)\|_2^2 | \xi(k)] \\ &= (\mathcal{Q}\xi(k))^T [(I_n - L_c)^T R(I_n - L_c) + (1 - \frac{1}{n})\tilde{L}_c] \mathcal{Q}\xi(k) \\ &\leq \rho_d \|\mathcal{Q}\xi(k)\|_2^2 \\ &= \rho_d \xi^T(k) \mathcal{Q}^T \mathcal{Q}\xi(k) = \rho_d \xi^T(k) \xi(k) = \rho_d \|\xi(k)\|_2^2 \end{aligned}$$

Hence, we obtain  $r_s \leq \rho_d$ . To prove  $\rho_d \leq r_s$ , note that the matrix

$$(I_n - L_c)^T R(I_n - L_c) + (1 - \frac{1}{n})\tilde{L}_c$$

is positive semi-definite and all its eigenvalues are nonnegative; thus,  $\rho_d$  is the largest eigenvalue. Let  $\mathbf{v} \in \mathbb{R}^{n-1}$  denote

the eigenvector corresponding to  $\rho_d$ . Also, note that 0 is its eigenvalue with corresponding eigenvector  $\mathbf{1}$ . Thus,  $\mathbf{v} \perp \mathbf{1}$  and  $\mathbf{v}^T \mathbf{1} = 0$ . Choose  $\mathbf{x}(k) = \mathbf{v}$ . Then,

$$Q\xi(k) = QQ^T \mathbf{x}(k) = (I_n - \frac{1}{n} \mathbf{1}\mathbf{1}^T) \mathbf{v} = \mathbf{v}$$

and

$$\begin{aligned} & \mathbf{E}[\|\xi^T(k+1)\|_2^2 | \xi(k)] \\ &= (Q\xi(k))^T [(I_n - L_c)^T R (I_n - L_c) + (1 - \frac{1}{n}) \tilde{L}_c] Q\xi(k) \\ &= \mathbf{v}^T [(I_n - L_c)^T R (I_n - L_c) + (1 - \frac{1}{n}) \tilde{L}_c] \mathbf{v} \\ &= \rho_d \mathbf{v}^T \mathbf{v} \\ &= \rho_d \xi^T(k) Q^T Q \xi(k) \\ &= \rho_d \|\xi^T(k)\|_2^2 \end{aligned}$$

Hence,  $r_s \geq \rho_d$ . Therefore, we conclude that  $r_s = \rho_d$ . ■

From Proposition 2 and Theorem 1, we have the following result. The proof is straight-forward and thus omitted due to space limit.

**Corollary 1:** When  $L_c$  is balanced, the per-step (mean square) convergence factor for the discrete-time consensus algorithm (11) over a dynamically switching directed random network is

$$r_s = \rho_d = \rho \left( R - L_c - L_c^T + L_c^T L_c + (1 - \frac{1}{n}) \tilde{L}_c \right)$$

#### IV. NUMERICAL EXAMPLES

##### A. Weight matrix with negative elements

This example considers the case in which some communication links have negative weights. Consider a group of 5 agents over a dynamically switching directed random network. Both the probability matrix  $P$  and the weight matrix  $W$  are generated randomly. In particular, each element  $p_{ij}$  ( $i, j = 1, \dots, 5$ ) of the edge probability matrix  $P$  is uniformly distributed between 0 and 1, and each element  $w_{ij}$  ( $i, j = 1, \dots, 5$ ) of the weight matrix  $W$  is randomly generated as the summation of two random variables (one uniformly distributed between 0 and 0.2 and the other uniformly distributed between -0.1 and 0)

$$P = \begin{bmatrix} 0 & 0.5837 & 0.2896 & 0.8208 & 0.4468 \\ 0.3081 & 0 & 0.8155 & 0.8644 & 0.7873 \\ 0.2856 & 0.2717 & 0 & 0.0858 & 0.2818 \\ 0.0781 & 0.7836 & 0.6091 & 0 & 0.4700 \\ 0.9532 & 0.2431 & 0.1231 & 0.3875 & 0 \end{bmatrix},$$

$$W = \begin{bmatrix} 0 & 0.1478 & 0.0511 & 0.1093 & 0.0629 \\ -0.0008 & 0 & 0.0130 & 0.0689 & -0.0032 \\ -0.0237 & 0.0094 & 0 & -0.0321 & 0.0554 \\ 0.0417 & -0.0011 & 0.0903 & 0 & -0.0026 \\ 0.1024 & 0.1091 & 0.0099 & -0.0200 & 0 \end{bmatrix}.$$

Note that  $P$  and  $W$  are not symmetric and some elements of  $W$  are negative. Hence, the corresponding matrix  $I_n - L(k)$  could have negative elements and thus it is not a stochastic matrix; therefore, the classical results in [3] [4] [24] can

not be directly applied to prove the almost sure asymptotic stability. By simple calculation using (23), we obtain the per-step convergence factor  $r_s = \rho_d = 0.9538 < 1$ . We conclude that the discrete-time consensus algorithm (11) almost surely converges to the agreement set asymptotically. Fig (1) shows the state trajectory of the 5 agents by applying the consensus algorithm (11). The initial condition of the simulation is randomly generated from a uniform distribution between 0 and 1 as  $X(0) = [0.0697, 0.2332, 0.7374, 0.7585, 0.6368]^T$ . For  $k > 111$ ,  $\|\xi(k)\|_2^2 < 0.001$  and the group decision value is 0.6414 at the 111th step. Note that the average of  $X(0)$  is 0.4871 rather than 0.6414. Hence, the observation for undirected random network that agents converge to the average of initial values [9] may not be applicable to directed random networks.

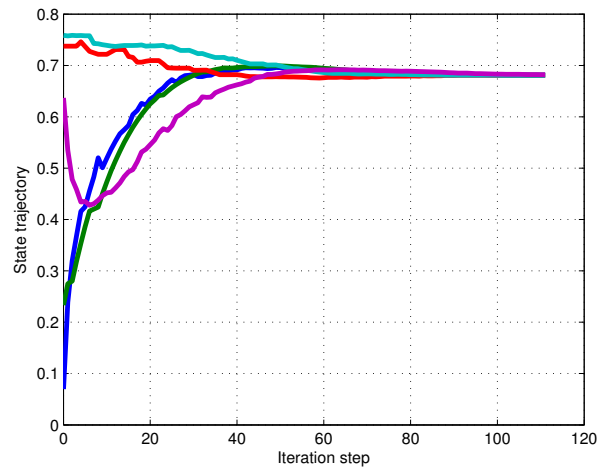


Fig. 1. State trajectory over a randomly switching directed network.

##### B. Common edge probability $p$

Consider a group of 6 agents over a dynamically switching directed random network where all links share a common edge probability  $p$ . Fig (2) plots the per-step convergence factor  $r_s$  as a function of  $p$  for a given set of weight matrices:

$$W_1 = \begin{bmatrix} 0 & 0.0822 & 0.0433 & 0.0913 & 0.0738 & 0.0898 \\ 0.0225 & 0 & 0.0611 & 0.0591 & 0.0511 & 0.0183 \\ 0.0184 & 0.0030 & 0 & 0.0575 & 0.0195 & 0.0022 \\ 0.0216 & 0.0192 & 0.0808 & 0 & 0.0868 & 0.0091 \\ 0.0427 & 0.0247 & 0.0509 & 0.0159 & 0 & 0.0522 \\ 0.0971 & 0.0567 & 0.0315 & 0.0275 & 0.0395 & 0 \end{bmatrix}$$

is randomly generated and uniformly distributed between 0 and 0.1, and  $W_2 = 4W_1$ ,  $W_3 = 5W_1$ .

From Fig (2), we see that for all  $0 < p \leq 1$ , the per-step convergence factors for  $W_1$  and  $W_2$  are less than 1, which guarantees almost sure consensus over the directed random networks with these two weight matrices. For  $W_3$ , when  $p = 1$ , the spectral radius of the matrix  $I - L_c$  is 1.1813, which implies that the discrete-time consensus algorithm (11) diverges over the fixed directed network with weight

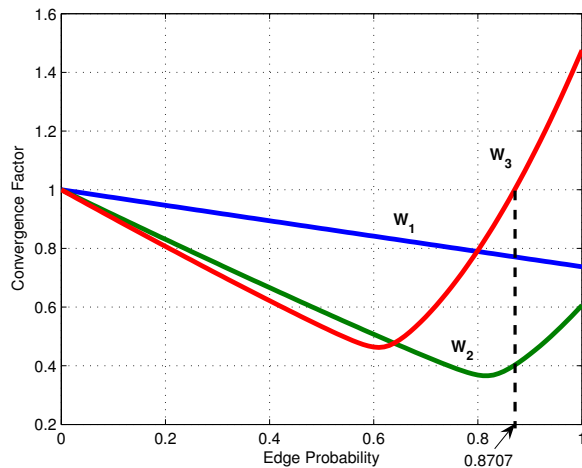


Fig. 2. Per-step convergence factor as a function of the edge probability.

matrix  $W_3$  in the deterministic setting. By contrast, when  $0 < p < 0.8708$ , the per-step convergence factor for  $W_3$  is less than 1, which guarantees almost sure convergence of the consensus algorithm (11) over the directed random network.

## V. CONCLUSIONS

In this paper, we have derived the convergence speed of a discrete-time consensus algorithm for a dynamically switching directed random network. The per-step convergence factor in the mean square sense is defined and calculated to provide a lower bound for the convergence speed of the consensus algorithm. Simulations are also given to illustrate the theoretic results.

## REFERENCES

[1] R. Albert and A.L. Barabási, “Statistical mechanics of complex networks”, *Review of Modern Physics*, vol. 74, pp. 47-97, 2002.

[2] D. Angeli and P. A. Bliman, “Convergence speed of distributed consensus and topology of the associated information spread”, *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 300-305.

[3] V. D. Blondel, J. M. Hendrickx, A. Olshevsky and J. N. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking”, *Proceedings of the IEEE Conference on Decision and Control*, 2005, pp. 2996-3000.

[4] M. Cao, A. S. Morse, and B. D. O. Anderson, “Reaching a Consensus in a Dynamically Changing Environment: Convergence Rates, Measurement Delays and Asynchronous Event”, *SIAM Journal on Control and Optimization*, 2008, pp. 601-623.

[5] R. Diestel, *Graph Theory*, Springer-Verlag, New York, NJ; 2005.

[6] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations”, *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465-1476, 2004.

[7] C. Godsil and G. Royle, *Algebraic graph theory*, Springer-Verlag, New York, NJ; 2001.

[8] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press; 1996.

[9] Y. Hatano and M. Mesbahi, “Agreement over random networks”, *IEEE Transactions on Automatic Control*, vol. 50, pp. 1867-1872, 2005.

[10] Y. Hatano, A. K. Das and M. Mesbahi, “Agreement in presence of noise: pseudogradients on random geometric networks”, *Proceedings of IEEE Conference on Decision and Control*, 2005, pp. 6382-6387.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press; 1985.

[12] A. Jadbabaie, J. Lin and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules”, *IEEE Transactions on Automatic Control*, vol. 48, pp. 988-1001, 2003.

[13] Z. Jin and R. M. Murray, “Random consensus protocol in large-scale networks”, *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 2007.

[14] S. Ka and J. Moura, “Distributed average consensus in sensor networks with random link failures”, *International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[15] H. J. Kushner, *Introduction to stochastic control*, Holt, Reinhart, and Winston, New York, NJ; 1971.

[16] L. Moreau, “Stability of multiagent systems with time-dependent communication links”, *IEEE Transactions on Automatic Control*, vol. 50, pp. 169-182, 2005.

[17] M. Mahmoud, *Active fault tolerant control systems: stochastic analysis and synthesis*, Springer, Berlin, New York; 2003.

[18] A. Olshevsky and J. N. Tsitsiklis, “Convergence rates in distributed consensus and averaging”, *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 3387-3392.

[19] S. Patterson, B. Bamieh and A. E. Abbadi, “Distributed average consensus with stochastic communication failures”, *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 4215-4220.

[20] M. Porfiri and D. J. Stilwell, “Consensus seeking over random weighted directed graphs”, *IEEE Transactions on Automatic Control*, vol. 52, pp. 1767-1773, 2007.

[21] W. Ren and R. W. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies”, *IEEE Transactions on Automatic Control*, vol. 50, pp. 655-661, 2005.

[22] R. O. Saber and R. M. Murray, “Consensus Protocols for Networks of Dynamic Agents”, *Proceedings of the American Control Conference*, 2003, pp. 951-956.

[23] R. O. Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays”, *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520-1533, 2004.

[24] A. Tahbaz-Salehi and A. Jadbabaie, “Necessary and Sufficient Condition for Consensus Over Random Networks”, *IEEE Transactions on Automatic Control*, 2008, pp. 791-795.

[25] H. Wang and Y. Guo, “Consensus on scale-free network”, *Proceedings of the IEEE Conference on Decision and Control*, 2008, pp. 748-752.

[26] C. W. Wu, “Synchronization and convergence of linear dynamics in random directed networks”, *IEEE Transactions on Automatic Control*, vol. 51, pp. 1207-1210, 2006.

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