

Distributed Randomized Algorithms for the PageRank Computation

Hideaki Ishii, *Member, IEEE*, and Roberto Tempo, *Fellow, IEEE*

Abstract—In the search engine of Google, the PageRank algorithm plays a crucial role in ranking the search results. The algorithm quantifies the importance of each web page based on the link structure of the web. We first provide an overview of the original problem setup. Then, we propose several distributed randomized schemes for the computation of the PageRank, where the pages can locally update their values by communicating to those connected by links. The main objective of the paper is to show that these schemes asymptotically converge in the mean-square sense to the true PageRank values. A detailed discussion on the close relations to the multi-agent consensus problems is also given.

Index Terms—Distributed computation, multi-agent consensus, PageRank algorithm, randomization, search engines, stochastic matrices.

I. INTRODUCTION

IN the last decade, search engines have become widely used indispensable tools for searching the web. For such engines, it is essential that the search results not only consist of web pages related to the query terms, but also rank the pages properly so that the users quickly have access to the desired information. The PageRank algorithm at Google is one of the successful algorithms that quantify and rank the importance of each web page. This algorithm was initially proposed in [8], and an overview can be found in, e.g., [9], [25].

One of the main features of the PageRank algorithm is that it is based solely on the link structure inherent in the web. The underlying key idea is that links from important pages make a page more important. More concretely, each page is considered to be voting the pages to which it is linked. Then, in the ranking of a page, the total number of votes as well as the importance of the voters are reflected. This problem is mathematically formulated as finding the eigenvector corresponding to the largest eigenvalue of a certain stochastic matrix associated with the web structure.

For the PageRank computation, a critical aspect is the size of the web. The web is said to be composed of over 8 billion pages, and its size is still growing. Currently, the computation is performed centrally at Google, where the data on the whole web structure is collected by crawlers automatically browsing

the web. In practice, the class of algorithms that can be applied is limited. In fact, the basic power method is employed, but it is reported that this computation takes about a week [25]. This clearly necessitates more efficient computational methods.

In this regard, several approaches have recently been proposed. In [22], an adaptive computation method is developed, which classifies web pages into groups based on the speed of convergence to the PageRank values and allocates computational resources accordingly. Another line of research is based on distributed approaches, where the computation is performed on multiple servers communicating to each other. For example, Monte Carlo methods are used in [4], while the work in [38] utilizes the block structure of the web to apply techniques from the Markov chain literature. In [12], [24], methods based on the so-called asynchronous iterations [5] in numerical analysis are discussed.

In this paper, we follow the distributed approach and, in particular, develop a randomized algorithm for the PageRank computation; for recent advances on probabilistic approaches in systems and control, see [34]. This algorithm is fully distributed and has three main features as follows: First, in principle, each page can compute its own PageRank value locally by communicating with the pages that are connected by direct links. That is, each page exchanges its value with the pages that it is linked to and those linked to it. Second, the pages make the decision to initiate this communication at random times which are independent from page to page. This means that, in its implementation, there is neither a fixed order among the pages nor a centralized agent in the web that determines the pages to update their values. Third, the computation required for each page is very mild.

The main result of the paper shows that the algorithm converges to the true PageRank values in the mean-square sense. This is achieved by computing the time average at each page. From a technical viewpoint, an important characteristic of the approach is that the stochasticity of the matrix in the original problem is preserved and exploited. We first propose a basic distributed update scheme for the pages and then extend this into two directions to enhance its performance and flexibility for implementation. It is further noted that in [19], [20], this approach has been generalized to incorporate failures in the communication as well as aggregation of the web structure. In [18], a related result on finding the variations in the PageRank values when the web data may contain errors is given.

We emphasize that the approach proposed here is particularly motivated by the recent research on distributed consensus, agreement, and flocking problems in the systems and control community; see, e.g., [6], [7], [10], [14], [21], [23], [26]–[28], [30], [31], [33], [35]–[37]. For additional details, we refer to [1], [3], [5]. Among such problems, our approach to PageRank computation is especially related to the consensus, where multiple

Manuscript received November 25, 2008; revised July 15, 2009 and November 30, 2009; accepted January 25, 2010. First published February 17, 2010; current version published September 09, 2010. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under KAKENHI Grant 21760323. Recommended by Associate Editor Z. Qu.

H. Ishii is with the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama 226-8502, Japan (e-mail: ishii@dis.titech.ac.jp).

R. Tempo is with IEIIT-CNR, Politecnico di Torino, Torino 10129, Italy (e-mail: roberto.tempo@polito.it).

Digital Object Identifier 10.1109/TAC.2010.2042984

agents exchange their values with neighboring agents so that they obtain consensus, i.e., all agents reach the same value. The objective is clearly different from that of the PageRank problem, which is to find a specific eigenvector of a stochastic matrix via the power method. However, the recursion appearing in the consensus algorithm is exactly in the same form as the one for our distributed PageRank computation except that the class of stochastic matrices is slightly different. These issues will be discussed further.

The organization of this paper is as follows: In Section II, we present an overview of the PageRank problem. The distributed approach is introduced in Section III, where we propose a basic scheme and prove its convergence. Its relation with multi-agent consensus problems is discussed in Section IV. We then develop two extensions of the basic distributed algorithm: One in Section V is to improve the rate of convergence by allowing multiple pages to simultaneously update and the other in Section VI to reduce the communication load among the pages. The proposed algorithm is compared with an approach known as asynchronous iteration from the field of numerical analysis in Section VII. Numerical examples are given in Section VIII to show the effectiveness of the proposed schemes. We conclude the paper in Section IX. Part of the material of this paper has appeared in a preliminary form in [17].

Notation: For vectors and matrices, inequalities are used to denote entry-wise inequalities: For $X, Y \in \mathbb{R}^{n \times m}$, $X \leq Y$ implies $x_{ij} \leq y_{ij}$ for $i = 1, \dots, n$ and $j = 1, \dots, m$; in particular, we say that the matrix X is nonnegative if $X \geq 0$ and positive if $X > 0$. A probability vector is a nonnegative vector $v \in \mathbb{R}^n$ such that $\sum_{i=1}^n v_i = 1$. Unless otherwise specified, by a stochastic matrix, we refer to a column-stochastic matrix, i.e., a nonnegative matrix $X \in \mathbb{R}^{n \times n}$ with the property that $\sum_{i=1}^n x_{ij} = 1$ for $j = 1, \dots, n$. Let $\mathbf{1} \in \mathbb{R}^n$ be the vector with all entries equal to 1 as $\mathbf{1} := [1 \ \dots \ 1]^T$. Similarly, $S \in \mathbb{R}^{n \times n}$ is the matrix with all entries being 1. For $x \in \mathbb{R}^n$, we denote by $|x|$ the vector containing the absolute values of the corresponding entries of x . The norm $\|\cdot\|$ for vectors is the Euclidean norm. The spectral radius of the matrix $X \in \mathbb{R}^{n \times n}$ is denoted by $\rho(X)$. We use I for the identity matrix.

II. THE PAGERANK PROBLEM

In this section, we provide a brief introductory description of the PageRank problem; this material can be found in, e.g., [8], [9], [25].

Consider a network of n web pages indexed by integers from 1 to n . This network is represented by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} := \{1, 2, \dots, n\}$ is the set of vertices corresponding to the web page indices while $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges representing the links among the pages. The vertex i is connected to the vertex j by an edge, i.e., $(i, j) \in \mathcal{E}$, if page i has an outgoing link to page j , or in other words, page j has an incoming link from page i . To avoid trivial situations, we assume $n \geq 2$.

The objective of the PageRank algorithm is to provide some measure of importance to each web page. The PageRank value, or simply the value, of page $i \in \mathcal{V}$ is a real number in $[0, 1]$; we denote this by x_i^* . The values are ordered such that $x_i^* > x_j^*$ implies that page i is more important than page j .

The basic idea in ranking the pages in terms of the values is that a page having links from important pages is also important. This is realized by determining the value of one page as a sum of the contributions from all pages that have links to it. In particular, the value x_i^* of page i is defined as

$$x_i^* = \sum_{j \in \mathcal{L}_i} \frac{x_j^*}{n_j}$$

where $\mathcal{L}_i := \{j : (j, i) \in \mathcal{E}\}$, i.e., this is the index set of pages linked to page i , and n_j is the number of outgoing links of page j . It is customary to normalize the total of all values so that $\sum_{i=1}^n x_i^* = 1$.

Let the values be in the vector form as $x^* \in [0, 1]^n$. Then, from what we described so far, the PageRank problem can be restated as

$$x^* = Ax^*, \quad x^* \in [0, 1]^n, \quad \sum_{i=1}^n x_i^* = 1 \quad (1)$$

where the matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, called the link matrix, is given by

$$a_{ij} := \begin{cases} \frac{1}{n_j} & \text{if } j \in \mathcal{L}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that the value vector x^* is a nonnegative unit eigenvector corresponding to the eigenvalue 1 of the nonnegative matrix A . In general, for this eigenvector to exist and to be unique, it is sufficient that the web as a graph is strongly connected [15].¹ However, the web is known not to be strongly connected. Thus, the problem is slightly modified as follows.²

First, note that in the real web, the so-called dangling nodes, which are pages having no links to others, are abundant. Such pages can be found, e.g., in the form of PDF document files having no outgoing links. These pages introduce zero columns into the link matrix. To simplify the discussion, we redefine the graph and thus the matrix A by bringing in artificial links for all dangling nodes (e.g., links back to the pages having links to a dangling node). As a result, the link matrix A becomes a stochastic matrix, that is, $\sum_{i=1}^n a_{ij} = 1$ for each j . This implies that there exists at least one eigenvalue equal to 1.

To emphasize the changes in the links that we have just made, we state the following as an assumption.

Assumption 2.1: The link matrix A given in (2) is a stochastic matrix.

Next, to guarantee the uniqueness of the eigenvalue 1, a modified version of the values has been introduced in [8] as follows: Let m be a parameter such that $m \in (0, 1)$, and let the modified link matrix $M \in \mathbb{R}^{n \times n}$ be defined by

$$M := (1 - m)A + \frac{m}{n}S. \quad (3)$$

¹A directed graph is said to be strongly connected if for any two vertices $i, j \in \mathcal{V}$, there is a sequence of edges which connects i to j . In terms of the link matrix A , strong connectivity of the graph is equivalent to A being irreducible.

²In fact, in the consensus literature, it is known that the eigenvalue 1 of a row-stochastic matrix is simple iff the underlying graph has at least one globally reachable node; this means that there is a node from which each node in the graph can be reached via a sequence of edges (see, e.g., [26], [28], [31]). For our purpose, it is indeed possible to provide the column-stochastic counterpart of globally reachable nodes, but the real web does not possess this property either.

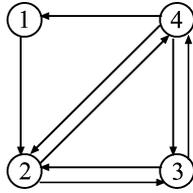


Fig. 1. Web with four pages.

In the original algorithm in [8], a typical value for m is chosen as $m = 0.15$; we use this value throughout this paper.³ Notice that M is a positive stochastic matrix. By the Perron theorem [15], this matrix is primitive;⁴ in particular, the eigenvalue 1 is of multiplicity 1 and is the unique eigenvalue of maximum modulus (i.e., with the maximum absolute value). Furthermore, the corresponding eigenvector is positive. Hence, we redefine the value vector x^* by using M in place of A in (1) as follows.

Definition 2.2: The PageRank value vector x^* is given by

$$x^* = Mx^*, \quad x^* \in [0, 1]^n, \quad \sum_{i=1}^n x_i^* = 1. \quad (4)$$

As mentioned in the Introduction, due to the large dimension of the link matrix M , the computation of the eigenvector corresponding to the eigenvalue 1 is difficult. The solution that has been employed in practice is based on the power method. That is, the value vector x^* is computed through the recursion

$$x(k + 1) = Mx(k) = (1 - m)Ax(k) + \frac{m}{n}\mathbf{1} \quad (5)$$

where $x(k) \in \mathbb{R}^n$ and the initial vector $x(0) \in \mathbb{R}^n$ is a probability vector. The second equality above follows from the fact $Sx(k) = \mathbf{1}$, $k \in \mathbb{Z}_+$. For implementation, the form on the far right-hand side is important, exhibiting that multiplication is required using only the sparse matrix A , and not the dense matrix M .

Based on this method, we can asymptotically find the value vector as shown below; see, e.g., [15].

Lemma 2.3: In the update scheme (5), for any initial state $x(0)$ that is a probability vector, it holds that $x(k) \rightarrow x^*$ as $k \rightarrow \infty$.

We now comment on the convergence rate of this scheme. Denote by $\lambda_1(M)$ and $\lambda_2(M)$ the largest and the second largest eigenvalues of M in magnitude. Then, for the power method applied to M , the asymptotic rate of convergence is exponential and depends on the ratio $|\lambda_2(M)/\lambda_1(M)|$. Since M is a positive stochastic matrix, we have $\lambda_1(M) = 1$ and $|\lambda_2(M)| < 1$. Furthermore, it is shown in [25] that the structure of the link matrix M leads us to the bound

$$|\lambda_2(M)| \leq 1 - m. \quad (6)$$

We next provide a simple example for illustration.

Example 2.4: Consider the web with four pages shown in Fig. 1. As a graph, this web is strongly connected, and there are no dangling nodes. The link matrix A and the modified link

³In [8], no specific reason is given for this choice of m . As shown later in (6), however, large m has the effect of faster convergence in the computation while it can also average out the PageRank values.

⁴A nonnegative matrix $X \in \mathbb{R}^{n \times n}$ is said to be primitive if it is irreducible and has only one eigenvalue of maximum modulus.

matrix M can easily be constructed by (2) and (3), respectively, as

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 1 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix},$$

$$M = \begin{bmatrix} 0.0375 & 0.0375 & 0.0375 & 0.3208 \\ 0.8875 & 0.0375 & 0.4625 & 0.3208 \\ 0.0375 & 0.4625 & 0.0375 & 0.3208 \\ 0.0375 & 0.4625 & 0.4625 & 0.0375 \end{bmatrix}$$

where we used the value $m = 0.15$ from [8]. Then, the value vector x^* can be computed as $x^* = [0.119 \ 0.331 \ 0.260 \ 0.289]^T$. Notice that page 2 has the largest value since it is linked from three pages while page 1, which has only one link to it, has the smallest value. On the other hand, pages 3 and 4 have the same number of incoming links, but page 4 has a larger value. This is because page 4 has more outgoing links, and thus it receives more contribution from page 3 than what it gives back. ∇

III. A DISTRIBUTED RANDOMIZED APPROACH

In this section, we propose a distributed approach to compute the value vector x^* .

Consider the web from the previous section. The basic protocol of the scheme is as follows: At time k , page i initiates its PageRank value update (i) by sending the value of page i to the pages that are linked and (ii) by requesting the values from the pages that are linked to page i . All pages involved in this process renew their values based on the latest available information.

To implement the scheme in a distributed manner, we assume that the pages taking the update action are determined in a random manner. This is specified by the random process $\theta(k) \in \mathcal{V}$, $k \in \mathbb{Z}_+$. If at time k , $\theta(k) = i$, then page i initiates an update action by communicating and exchanging the values with the pages connected by incoming and outgoing links. Specifically, $\theta(k)$ is assumed to be i.i.d., and its probability distribution is given by

$$\text{Prob}\{\theta(k) = i\} = \frac{1}{n}, \quad \forall k \in \mathbb{Z}_+. \quad (7)$$

This means that each page takes the update action with equal probability. In principle, this scheme may be implemented without requiring a centralized decision maker or any fixed order among the pages.

In particular, consider the distributed update scheme in the following form:

$$x(k + 1) = (1 - \hat{m})A_{\theta(k)}x(k) + \frac{\hat{m}}{n}\mathbf{1} \quad (8)$$

where $x(k) \in \mathbb{R}^n$ is the state whose initial state $x(0)$ is a probability vector, $\theta(k) \in \{1, \dots, n\}$ is the mode of the system, and A_i , $i = 1, \dots, n$, are called the distributed link matrices and are to be determined; $\hat{m} \in (0, 1)$ is a parameter replacing m in the centralized scheme (5).

The objective here is to design this distributed update scheme by finding the appropriate link matrices A_i and the parameter \hat{m} so that the PageRank values are computed through the time

average of the state x . Let $y(k)$ be the average of the sample path $x(0), \dots, x(k)$ as

$$y(k) := \frac{1}{k+1} \sum_{\ell=0}^k x(\ell). \quad (9)$$

We say that, for the distributed update scheme, the PageRank value x^* is obtained through the time average y if, for each initial state $x(0)$ that is a probability vector, $y(k)$ converges to x^* in the mean-square sense as follows:

$$E \left[\|y(k) - x^*\|^2 \right] \rightarrow 0, \quad k \rightarrow \infty. \quad (10)$$

This type of convergence is called ergodicity for stochastic processes [29].

In what follows, we develop the distributed update scheme of (8). The main result is presented as Theorem 3.4 showing the convergence of the scheme. The design consists of two steps, one for the link matrices A_i and then the parameter \hat{m} . In later sections, this approach will be extended to improve the convergence rate and the necessary communication load.

A. Distributed Link Matrices and Their Average

The first step in the development is to introduce the distributed link matrices. For each i , the matrix $A_i \in \mathbb{R}^{n \times n}$ is obtained as follows: (i) The i th row and column coincide with those of A ; (ii) the remaining diagonal entries are equal to $1 - a_{i\ell}$, $\ell = 1, \dots, n$, $\ell \neq i$; and (iii) all the remaining entries are zero. More formally, we have

$$(A_i)_{j\ell} := \begin{cases} a_{j\ell} & \text{if } j = i \text{ or } \ell = i, \\ 1 - a_{i\ell} & \text{if } j = \ell \neq i, \\ 0 & \text{otherwise,} \end{cases} \quad i = 1, \dots, n. \quad (11)$$

It follows that these matrices are stochastic because the original link matrix A possesses this property. As we shall see later, this property indeed is critical for the convergence of the scheme.

Example 3.1: We continue with the 4-page web in Example 2.4. The link matrices A_i are given by

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{2}{3} \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{2}{3} \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

▽

To clarify the properties of the link matrices A_i just introduced, we consider the simpler update scheme

$$x(k+1) = A_{\theta(k)}x(k) \quad (12)$$

where $x(k)$ is the state with $x(0)$ being a probability vector, and the mode $\theta(k)$ is specified in (7). In particular, we focus on its average dynamics. The mean $\bar{x}(k) := E[x(k)]$ of the state $x(k)$ follows the recursion $\bar{x}(k+1) = \bar{A}\bar{x}(k)$, where $\bar{A} := E[A_{\theta(k)}]$ is the average matrix and $E[\cdot]$ is the expectation with respect to

the random process $\theta(k)$. Hence, we now inspect this matrix \bar{A} . Due to the probability distribution of $\theta(k)$ in (7), we have

$$\bar{A} = \frac{1}{n} \sum_{i=1}^n A_i. \quad (13)$$

It is obvious that \bar{A} is a stochastic matrix since all A_i are stochastic.

The following lemma shows some properties of this matrix \bar{A} that will be useful later.

Lemma 3.2: For the average matrix \bar{A} given in (13), we have the following:

- i) $\bar{A} = (2/n)A + ((n-2)/n)I$.
- ii) There exists a vector $z_0 \in \mathbb{R}_+^n$ which is an eigenvector corresponding to the eigenvalue 1 for both matrices A and \bar{A} .

Proof:

- i) By definition of \bar{A} , we have

$$\begin{aligned} (\bar{A})_{j\ell} &= \frac{1}{n} \sum_{i=1}^n (A_i)_{j\ell} \\ &= \begin{cases} \frac{1}{n} \left[a_{jj} + \sum_{i=1, i \neq j}^n (1 - a_{ij}) \right] & \text{if } j = \ell, \\ \frac{2}{n} a_{j\ell} & \text{if } j \neq \ell. \end{cases} \end{aligned}$$

By definition of A , we have $a_{jj} = 0$ and $\sum_{i=1, i \neq j}^n a_{ij} = 1$. Thus, the expression for \bar{A} follows.

- ii) From (i), we have $\bar{A} - I = 2/n(A - I)$. This implies that every eigenvector z_0 of the link matrix A associated with the eigenvalue 1 is also an eigenvector of the average matrix \bar{A} for the same eigenvalue. □

The lemma above provides some justification for the proposed distributed approach. That is, even though the matrices A and \bar{A} have different structures, they share an eigenvector for the eigenvalue 1, which corresponds to the PageRank vector.

B. Mean-Square Convergence of the Distributed Update Scheme

As in the case with the original link matrix A , for the average matrix \bar{A} , the eigenvector corresponding to the eigenvalue 1 may not be unique. We follow an argument similar to that in Section II and introduce the modified versions of the distributed link matrices.

Since the link matrices A_i are stochastic, we can rewrite the distributed update scheme in (8) as

$$x(k+1) = M_{\theta(k)}x(k) \quad (14)$$

where the modified distributed link matrices are given by

$$M_i := (1 - \hat{m})A_i + \frac{\hat{m}}{n}S, \quad i = 1, \dots, n. \quad (15)$$

This expression is derived by the relation $Sx(k) = \mathbf{1}$ because $x(k)$ in (8) is a probability vector for each k . Note that M_i are positive stochastic matrices.

Similarly to the argument on the link matrices A_i above, the problem at this second step is as follows: We shall find the modified link matrices M_i by choosing the parameter \hat{m} such that

their average and the link matrix M from (3) share an eigenvector for the eigenvalue 1. Since such an eigenvector is unique for M , it is necessarily equal to the value vector x^* (see (4)).

Let $\bar{x}(k) := E[x(k)]$ be the mean of the state $x(k)$ of the system (14). Its dynamics is expressed as

$$\bar{x}(k+1) = \bar{M}\bar{x}(k) \quad (16)$$

where $\bar{x}(0) = x(0)$ and the average matrix \bar{M} is given by $\bar{M} := E[M_{\theta(k)}]$.

A simple way of defining M_i would be to let $\hat{m} = m$ as in the case with M ; however, it can be shown that there is no clear relation between the original matrix M and the average matrix \bar{M} such as that between A and \bar{A} as we have seen in Lemma 3.2. Instead, we take the parameter \hat{m} as

$$\hat{m} = \frac{2m}{n - m(n-2)}. \quad (17)$$

For the value $m = 0.15$ used in this paper, we have $\hat{m} = 0.3/(0.85n + 0.3)$. For this choice of \hat{m} , the next result holds.

Lemma 3.3: For the parameter \hat{m} given in (17), we have the following:

- i) $\hat{m} \in (0, 1)$ and $\hat{m} < m$.
- ii) $\bar{M} = (\hat{m}/m)M + (1 - (\hat{m}/m))I$.
- iii) For the average matrix \bar{M} , the eigenvalue 1 is simple and is the unique eigenvalue of maximum modulus. The value vector x^* is the corresponding eigenvector.

Proof:

- i) By the assumptions $m \in (0, 1)$ and $n \geq 2$, \hat{m} in (17) is positive. Also, we have $1 - \hat{m} = n(1-m)/[n-m(n-2)]$. Hence, $1 - \hat{m} > 1 - m$, that is, $\hat{m} < m < 1$.
- ii) This can be shown by direct calculation as follows:

$$\begin{aligned} \bar{M} &= (1 - \hat{m})\bar{A} + \frac{\hat{m}}{n}S \\ &\quad (\text{by the definition of } \bar{M}, (15), \text{ and then } (13)) \\ &= (1 - \hat{m}) \left[\frac{2}{n}A + \frac{n-2}{n}I \right] + \frac{\hat{m}}{n}S \quad (\text{by Lemma 3.2(i)}) \\ &= \frac{\hat{m}}{m}M + \left(1 - \frac{\hat{m}}{m}\right)I \\ &\quad (\text{by } \hat{m} \text{ in } (17) \text{ and the definition of } M \text{ in } (3)) \end{aligned}$$

- iii) From (ii), we have $\bar{M} - I = \hat{m}/m(M - I)$. Hence, \bar{M} and M share an eigenvector for the eigenvalue 1. However, both \bar{M} and M are positive stochastic matrices. Therefore, by the Perron theorem [15], for these matrices, the eigenvalue 1 is of multiplicity 1 and is the unique one having the maximum magnitude. Moreover, by (4), the corresponding eigenvector coincides with x^* . \square

From (iii) in the lemma above, it follows that the value vector x^* can be obtained by the power method, i.e., by the average system in (16) as $\bar{x}(k) \rightarrow x^*$, $k \rightarrow \infty$. Hence, in an average sense, the distributed update scheme asymptotically provides the correct values. It is interesting to observe that this can be achieved though the original link matrix A does not explicitly appear in the scheme. In fact, an eigenvector of the matrix M is computed through randomly switching among the distributed link matrices M_i .

However, this property turns out not to be sufficient to guarantee convergence of $x(k)$ to the true value x^* . From Lemma A.4 (ii) in the Appendix, we can show that for any sequence $\{\theta(k)\}$, there exists a sequence of probability vectors $\{v(k)\}$ such that, for any $x(0)$, it holds that $x(k) - v(k)\mathbf{1}^T x(0) = x(k) - v(k) \rightarrow 0$ as $k \rightarrow \infty$. The vector v and hence the state x in general do not converge. This can be seen in Example 3.1 when, e.g., page 1 initiates an update ($\theta(k) = 1$); the update for page 4 is given by $x_4(k+1) = 2(1-m)/3x_4(k) + m/4$, showing that x_4 cannot stay at its equilibrium value. Therefore, in the distributed approach, we resort to computing the time average $y(k)$ of the states.

The following theorem is the main result of this section. It shows that the time average indeed converges to the value vector in the mean-square sense.

Theorem 3.4: In the distributed update scheme in (14), the PageRank value x^* is obtained through the time average y in (9) as $E[\|y(k) - x^*\|^2] \rightarrow 0$, $k \rightarrow \infty$.

The theorem highlights an ergodic property in the proposed update scheme. It can be shown by general Markov process results in, e.g., [11]. For completeness, however, a proof more specific to the current setup is provided in Appendix B; it employs tools for stochastic matrices and moreover is useful for an extension given in Section VI. Regarding the convergence of this algorithm, we see from (53) in the proof that it is of order $1/k$ and moreover depends on the size of n linearly through the parameter \hat{m} in (17).

Several remarks are in order. In practice, each page needs to communicate with the pages that are directly connected by incoming or outgoing links. We emphasize that the recursion to be used is (8) and not the equivalent expression of (14); in the latter case, the link matrices M_i are positive, which can imply that the values of all pages are required for an update of a page. Nevertheless, as can be seen in (8), the link matrices A_i are sparse. Thus, at time k , communication is required only among the pages corresponding to the nonzero entries in $A_{\theta(k)}$. Each page then performs weighted addition of its own value, the values just received, and the extra term \hat{m}/n . Consequently, the amount of computation required at each page is limited at any time.

Implementation issues such as how web pages can exactly make local computations are outside the scope of this paper. However, it is clear that certain regulations may be necessary so that page owners cooperate with the search engine and the PageRank values computed by them can be trusted.⁵ Another issue concerning reliability of the ranking is that of link spam, i.e., links added to enhance the PageRank of certain pages on purpose; a method to detect such spamming is studied in, e.g., [2], [25].

IV. RELATIONS TO CONSENSUS PROBLEMS

In this section, we discuss the relation between the two problems of PageRank and consensus. First, we describe a stochastic version of the consensus problem. Such problems have been studied in, e.g., [7], [14], [33], [36]; see also [35].

⁵In the consensus literature, problems involving cheating have been studied. An example is the Byzantine agreement problem, where among the agents there are malicious ones who send confusing information so that other agents cannot achieve consensus (see, e.g., [35]).

Consider a set $\mathcal{V} = \{1, 2, \dots, n\}$ of agents having scalar values. The network of agents is represented by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex i is connected to the vertex j by an edge $(i, j) \in \mathcal{E}$ if agent i can communicate its value to agent j . Assume that the graph is strongly connected.⁶

The objective is that all agents reach a common value by communicating to each other, where the pattern in the communication is randomly determined at each time. Let $x_i(k)$ be the value of agent i held at time k , and let $x(k) := [x_1(k) \cdots x_n(k)]^T \in \mathbb{R}^n$. The values are updated via the recursion

$$x(k+1) = A_{\theta(k)}x(k) \quad (18)$$

where $\theta(k) \in \{1, \dots, d\}$ is the mode specifying the communication pattern among the agents and d is the number of such patterns. The communication patterns are given as follows: Each $i \in \{1, \dots, d\}$ corresponds to the subset $\mathcal{E}_i \subset \mathcal{E}$ of the edge set. Then, the matrix A_i has $(A_i)_{j\ell} > 0$ if and only if $(\ell, j) \in \mathcal{E}_i$. We assume that (i) $(j, j) \in \mathcal{E}_i$ for all j , (ii) $\bigcup_{i=1}^d \mathcal{E}_i = \mathcal{E}$, and (iii) the matrix A_i is a row-stochastic matrix. The communication pattern is random, and in particular, $\theta(k)$ is an i.i.d. random process. Its probability distribution is given by $\text{Prob}\{\theta(k) = i\} = 1/d$ for $k \in \mathbb{Z}_+$.

We say that consensus is obtained if for any initial vector $x(0) \in \mathbb{R}^n$, it holds that

$$|x_i(k) - x_j(k)| \rightarrow 0, \quad k \rightarrow \infty \quad (19)$$

with probability one for all $i, j \in \mathcal{V}$.

A well-known approach is to update the value of each agent by taking the average of the values received at that time. In this case, the matrix A_i is constructed as

$$(A_i)_{j\ell} := \begin{cases} \frac{1}{n_{ij}} & \text{if } (\ell, j) \in \mathcal{E}_i, \\ 0 & \text{otherwise} \end{cases}$$

where n_{ij} is the number of agents ℓ with $(\ell, j) \in \mathcal{E}_i$, i.e., those that transmit their values to agent j .

Example 4.1: Consider the graph in Example 2.4 with four agents. We introduce four communication patterns arising from the protocol in the distributed PageRank algorithm: The edge subset \mathcal{E}_i contains all (i, j) and (j, i) in the original edge set \mathcal{E} including (i, i) that corresponds to a self-loop for $i, j = 1, 2, 3, 4$. The matrices A_i can be written as

$$A_1 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad A_4 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

▽

We now present the convergence result for consensus.

⁶As discussed in Section II, this assumption can be replaced with a weaker one that a globally reachable node exists.

Lemma 4.2: Assume that the graph is strongly connected. Then under the scheme of (18), where the communication pattern is chosen randomly, consensus is obtained in the sense of (19).

Outline of the Proof: Let $\bar{A} := E[A_{\theta(k)}]$ be the average matrix. This matrix is stochastic and irreducible. This is because the original graph is strongly connected, and hence under the probability distribution of $\theta(k)$, we have $(\bar{A})_{j\ell} > 0$ for each $(\ell, j) \in \mathcal{E}$. Furthermore, by definition, the diagonal entries are positive, and thus \bar{A} is a primitive matrix, implying that it has the unique eigenvalue 1 of maximum modulus [15]. Thus, by [33], it follows that consensus is obtained. ▽

In comparison with the distributed PageRank problem, the consensus problem has the features below:

- i) The graph is assumed to be strongly connected.
- ii) The goal is that all values $x_i(k)$ become equal, and moreover there is no restriction on its size.
- iii) Convergence with probability one can be attained for the values $x_i(k)$ directly; there is no need to consider their time average (as in $y_i(k)$ in (9)).
- iv) The matrices A_i are row stochastic and the diagonal entries are all positive. In contrast, in our distributed PageRank computation scheme, the link matrices are column stochastic. However, the coefficient of ergodicity, which is the tool employed for proving Theorem 3.4, is useful also for this problem; see, e.g., [33].

It is clear that many similarities exist between the algorithms for consensus and PageRank. We emphasize that the distributed PageRank approach in this paper has been particularly motivated by the recent advances in the consensus literature. We highlight two points that provide helpful insights into the PageRank problem as follows:

- 1) At the conceptual level, it is natural to view the web as a network of agents that can make its own computation as well as communication with their neighboring agents.
- 2) At the technical level, it is important to impose stochasticity on the distributed link matrices. For the distributed PageRank computation, very few works exploit this viewpoint.

V. EXTENSIONS TO SIMULTANEOUS UPDATES

So far, we have discussed the update scheme where only one page initiates an update at each time instant. In the web with billions of pages, however, this approach may not be practical. In this section, we extend the distributed algorithm by allowing multiple pages to simultaneously initiate updates.

Consider the web with n pages from Sections II and III. As before, at each time k , the page i initiates its PageRank value update (i) by sending its value to the pages that it is linked to and (ii) by requesting the pages that link to it for their values. All pages involved here update their values based on the new information. The difference from the simpler scheme before is that there may be pages that are requested for their values by multiple pages at the same time. The current scheme handles such situations.

These updates can take place in a fully distributed and randomized manner. The decision to make an update is a random variable. In particular, this is determined under a given probability $\alpha \in (0, 1]$ at each time k , and hence, the decision can be

made locally at each page. Note that the probability α is a global parameter in that all pages share the same α .

Formally, the proposed distributed update scheme is described as follows. Let $\eta_i(k) \in \{0, 1\}$, $i = 1, \dots, n$, $k \in \mathbb{Z}_+$, be Bernoulli processes given by

$$\eta_i(k) = \begin{cases} 1 & \text{if page } i \text{ initiates an update at time } k, \\ 0 & \text{otherwise} \end{cases}$$

where their probability distributions are specified as

$$\alpha := \text{Prob} \{ \eta_i(k) = 1 \}. \quad (20)$$

The process $\eta_i(k)$ is generated at the corresponding page i .

Similarly to the argument in Section III, we start with the update law as in (8):

$$x(k+1) = (1 - \hat{m})A_{\eta_1(k), \dots, \eta_n(k)}x(k) + \frac{\hat{m}}{n}\mathbf{1} \quad (21)$$

where $x(k) \in \mathbb{R}^n$, the initial state $x(0)$ is a probability vector, $\hat{m} \in (0, 1)$ is the parameter used instead of m in the centralized case, and $A_{\eta_1(k), \dots, \eta_n(k)}$ are the distributed link matrices.

The problem of distributed PageRank computation is formulated as follows: Find the distributed link matrices A_{p_1, \dots, p_n} and the parameter \hat{m} such that, for the corresponding distributed update scheme (21), the PageRank value x^* is obtained through the time average. This problem is a generalization of that in Section III, where only one page initiates an update at a time. The current approach is called the distributed scheme with simultaneous updates. Its analysis is more involved as we shall see.

A. Distributed Link Matrices and Their Average

We introduce the distributed link matrices. Let the matrices A_{p_1, \dots, p_n} be given by

$$(A_{p_1, \dots, p_n})_{ij} := \begin{cases} a_{ij} & \text{if } p_i = 1 \text{ or } p_j = 1, \\ 1 - \sum_{h: p_h=1} a_{hj} & \text{if } p_i = 0 \text{ and } i = j, \\ 0 & \text{if } p_i = p_j = 0 \text{ and } i \neq j \end{cases} \quad (22)$$

for $p_r \in \{0, 1\}$, $r \in \{1, \dots, n\}$, and $i, j \in \{1, \dots, n\}$. Clearly, there are 2^n matrices. They have the property that (i) if $p_i = 1$, then the i th column and the i th row are the same as those in the original link matrix A , (ii) if $p_i = 0$, then the i th diagonal entry is chosen so that the entries of the i th column add up to 1, and (iii) all other entries are 0. Hence, these are constructed as stochastic matrices. Notice that the link matrix A_{p_1, \dots, p_n} coincides with A_i in (11) when $p_i = 1$ and $p_j = 0$ for all $j \neq i$.

We next analyze the average dynamics of the distributed update scheme in (21). For simplicity, as in Section III, we use the same notation \bar{A} for the average link matrix given by

$$\bar{A} := E [A_{\eta_1(k), \dots, \eta_n(k)}] \quad (23)$$

where $E[\cdot]$ is the expectation with respect to $\eta_i(k)$, $i \in \mathcal{V}$. This matrix \bar{A} is stochastic.

The following result shows that the average link matrix \bar{A} has a clear relation to the original link matrix A . In particular, it implies that the two matrices share the eigenvector for eigenvalue 1.

Proposition 5.1: For the average link matrix \bar{A} given in (23), we have the following:

i) $\bar{A} = [1 - (1 - \alpha)^2]A + (1 - \alpha)^2I$.

ii) There exists a vector $z_0 \in \mathbb{R}_+^n$ which is an eigenvector corresponding to the eigenvalue 1 for both A and \bar{A} .

The proof of this proposition is preceded by a preliminary result. Observe that \bar{A} can be written as

$$\bar{A} = \sum_{\ell=0}^n \alpha^\ell (1 - \alpha)^{n-\ell} \hat{A}_\ell \quad (24)$$

where the matrices \hat{A}_ℓ , $\ell = 0, 1, \dots, n$, are given by

$$\hat{A}_\ell := \sum_{\substack{p_r \in \{0,1\}, r=1, \dots, n: \\ \sum_{r=1}^n p_r = \ell}} A_{p_1, \dots, p_n}. \quad (25)$$

The matrix \hat{A}_ℓ is the sum of matrices for the cases where ℓ pages simultaneously initiate updates.

These matrices can be explicitly written in terms of the original link matrix A . Here, we use the binomial coefficient given by ${}_r C_k := r! / [(r - k)!k!]$. Note that ${}_r C_0 = 1$ for any $r \in \mathbb{Z}_+$.

Lemma 5.2: The matrices \hat{A}_ℓ , $\ell = 0, 1, \dots, n$, in (25) can be expressed as follows:

$$\hat{A}_\ell = \begin{cases} A & \text{if } \ell = n, \\ nA & \text{if } \ell = n - 1, \\ ({}_n C_\ell - {}_{n-2} C_\ell)A + {}_{n-2} C_\ell I & \text{if } \ell = 0, \dots, n - 2. \end{cases} \quad (26)$$

Proof: We consider four cases as follows.

- 1) $\ell = n$: This is the case when all pages initiate updates, and thus by definition $\hat{A}_n = A_{1, \dots, 1} = A$.
- 2) $\ell = n - 1$: When all but one page initiate updates, it is obvious from the definition that $A_{0, 1, \dots, 1} = A_{1, 0, 1, \dots, 1} = \dots = A_{1, \dots, 1, 0} = A$. Since there are n such cases, their sum is $\hat{A}_{n-1} = nA$.
- 3) $\ell = 0$: In the case when none of the pages initiates an update, by definition, the matrix $A_{0, \dots, 0}$ reduces to the identity matrix as $\hat{A}_0 = A_{0, \dots, 0} = I$. Noting that ${}_n C_0 = {}_{n-2} C_0 = 1$, we have (26).
- 4) $\ell = 1, \dots, n - 2$: To prove the expression of \hat{A}_ℓ for these cases, we must show for each entry that

$$(\hat{A}_\ell)_{ij} = \begin{cases} ({}_n C_\ell - {}_{n-2} C_\ell)a_{ij} & \text{if } i \neq j, \\ {}_{n-2} C_\ell & \text{if } i = j, \end{cases} \quad i, j \in \{1, \dots, n\}. \quad (27)$$

In the following, the proof is divided into two steps for the cases of $i \neq j$ and $i = j$.

- i) $i \neq j$: By the definition of A_{p_1, \dots, p_n} in (22), its (i, j) entry reduces to

$$(A_{p_1, \dots, p_n})_{ij} = \begin{cases} a_{ij} & \text{if } p_i = 1 \text{ or } p_j = 1, \\ 0 & \text{otherwise} \end{cases}.$$

Thus, from (25)

$$\begin{aligned} (\hat{A}_\ell)_{ij} &= \sum_{\substack{p_r \in \{0,1\}, r=1, \dots, n: \\ \sum_{r=1}^n p_r = \ell}} (A_{p_1, \dots, p_n})_{ij} \\ &= \sum_{\substack{p_r \in \{0,1\}, r=1, \dots, n: \\ \sum_{r=1}^n p_r = \ell \text{ and} \\ (p_i=1 \text{ or } p_j=1)}} a_{ij} \\ &= ({}_{2n-1} C_{\ell-1} - {}_{n-2} C_{\ell-2})a_{ij} \end{aligned} \quad (28)$$

where the last equality is established by counting the number of cases where $p_i = 1$ or $p_j = 1$ holds among all possible combinations of $p_1, \dots, p_n \in \{0, 1\}$ such that

ℓ of them are equal to 1. Using the formula for binomial coefficients

$${}_r C_k = {}_{r-1} C_k + {}_{r-1} C_{k-1}, \quad r, k \in \mathbb{Z}_+ \quad (29)$$

we can show that ${}_{2n-1} C_{\ell-1} - {}_{n-2} C_{\ell-2} = {}_n C_{\ell} - {}_{n-2} C_{\ell}$. Hence, from (28), we arrive at (27) for $i \neq j$.

ii) $i = j$: Since $a_{ii} = 0$ in the link matrix A in (2) and by (22), the (i, i) entry of A_{p_1, \dots, p_n} is

$$(A_{p_1, \dots, p_n})_{ii} = \begin{cases} 1 - \sum_{h: p_h=1} a_{hi} & \text{if } p_i = 0, \\ 0 & \text{if } p_i = 1. \end{cases}$$

Hence

$$\begin{aligned} (\hat{A}_{\ell})_{ii} &= \sum_{\substack{p_r \in \{0,1\}, r=1, \dots, n: \\ \sum_{r=1}^n p_r = \ell}} (A_{p_1, \dots, p_n})_{ii} \\ &= \sum_{\substack{p_r \in \{0,1\}, r=1, \dots, n: \\ \sum_{r=1}^n p_r = \ell, p_i = 0}} \left(1 - \sum_{h: p_h=1} a_{hi} \right) \\ &= {}_{n-1} C_{\ell} - \sum_{\substack{h=1 \\ h \neq i}}^n \sum_{\substack{p_r \in \{0,1\}, r=1, \dots, n: \\ \sum_{r=1}^n p_r = \ell, p_i = 0, p_h = 1}} a_{hi} \end{aligned}$$

where the first term is obtained by counting the number of possible combinations of p_1, \dots, p_n such that their sum equals ℓ and $p_i = 0$; the second term is a consequence of switching the order of two summations. By a combinatorial argument again, we have $(\hat{A}_{\ell})_{ii} = {}_{n-1} C_{\ell} - {}_{n-2} C_{\ell-1} \sum_{h=1, h \neq i}^n a_{hi}$. The original link matrix A in (2) is stochastic with diagonal entries being 0. This fact together with (29) yields $(\hat{A}_{\ell})_{ii} = {}_{n-1} C_{\ell} - {}_{n-2} C_{\ell-1} = {}_{n-2} C_{\ell}$. Therefore, (27) is attained for the case $i = j$. \square

Proof of Proposition 5.1:

i) By (24) and Lemma 5.2, the matrix \bar{A} can be computed directly as

$$\begin{aligned} \bar{A} &= \sum_{\ell=0}^n \alpha^{\ell} (1-\alpha)^{n-\ell} \hat{A}_{\ell} \\ &= \sum_{\ell=0}^{n-2} \alpha^{\ell} (1-\alpha)^{n-\ell} [(n C_{\ell} - {}_{n-2} C_{\ell}) A + {}_{n-2} C_{\ell} I] \\ &\quad + \alpha^{n-1} (1-\alpha) n A + \alpha^n A \\ &= \sum_{\ell=0}^n \alpha^{\ell} (1-\alpha)^{n-\ell} {}_n C_{\ell} A \\ &\quad + \sum_{\ell=0}^{n-2} \alpha^{\ell} (1-\alpha)^{n-\ell} {}_{n-2} C_{\ell} (I - A). \end{aligned} \quad (30)$$

In the first term above, we have by the binomial identity $\sum_{\ell=0}^n \alpha^{\ell} (1-\alpha)^{n-\ell} {}_n C_{\ell} = [\alpha + (1-\alpha)]^n = 1$. Similarly, for the second term, $\sum_{\ell=0}^{n-2} \alpha^{\ell} (1-\alpha)^{n-\ell} {}_{n-2} C_{\ell} = (1-\alpha)^2$. Substituting these relations into (30), we have $\bar{A} = A + (1-\alpha)^2 (I - A)$, which is the desired expression of \bar{A} .

ii) The equality in (i) implies that any eigenvector z_0 of the link matrix A associated with the eigenvalue 1 is also an eigenvector of the average matrix \bar{A} for this eigenvalue. \square

B. Mean-Square Convergence of the Distributed Update Scheme

To guarantee that the distributed scheme yields the PageRank value, we now examine the modified versions of the distributed link matrices. We first express the distributed update scheme of (21) in its equivalent form as

$$x(k+1) = M_{\eta_1(k), \dots, \eta_n(k)} x(k) \quad (31)$$

where the modified distribution link matrices are given by

$$M_{p_1, \dots, p_n} := (1 - \hat{m}) A_{p_1, \dots, p_n} + \frac{\hat{m}}{n} S, \quad p_1, \dots, p_n \in \{0, 1\}. \quad (32)$$

This form can be obtained by using the facts that the link matrices A_{p_1, \dots, p_n} are stochastic and that $Sx(k) = \mathbf{1}$. Clearly, these matrices M_{p_1, \dots, p_n} are positive and stochastic.

The objective here is to find the modified link matrices M_{p_1, \dots, p_n} , by selecting the parameter \hat{m} , so that their average and the link matrix M from (3) share an eigenvector corresponding to the eigenvalue 1. Since such an eigenvector is unique for M , it is necessarily equal to the value vector x^* .

As in the earlier case in Section III, we take the parameter \hat{m} to be different from the original m . In particular, let

$$\hat{m} = \frac{m[1 - (1-\alpha)^2]}{1 - m(1-\alpha)^2}. \quad (33)$$

For the value $m = 0.15$ used in this paper, we have $\hat{m} = 0.15[1 - (1-\alpha)^2]/[1 - 0.15(1-\alpha)^2]$. Then, let the average link matrix be $\bar{M} := E[M_{\eta_1(k), \dots, \eta_n(k)}]$. Here, the distributed link matrices are positive stochastic matrices, which implies that the average matrix \bar{M} enjoys the same property.

The next lemma is the key to establish the desired relation between the distributed link matrices and their average. It is stated without proof; it follows similarly to that for Lemma 3.3.

Lemma 5.3: The scalar \hat{m} in (33) and the link matrices M_{p_1, \dots, p_n} in (32) have the following properties:

- i) $\hat{m} \in (0, 1)$ and $\hat{m} \leq m$.
- ii) $\bar{M} = (\hat{m}/m)M + (1 - (\hat{m}/m))I$.
- iii) For the average matrix \bar{M} , the eigenvalue 1 is simple and is the unique eigenvalue of maximum modulus. The value vector x^* is the corresponding eigenvector.

We can show by (iii) in the lemma that, in an average sense, the distributed update scheme asymptotically obtains the correct values. More precisely, we have $E[x(k)] = \bar{M}^k x(0) \rightarrow x^*$ as $k \rightarrow \infty$. Further, as discussed in Section II, the asymptotic rate of convergence is dominated by the second largest eigenvalue $\lambda_2(\bar{M})$ in magnitude. By (6) and (ii) in the lemma, this eigenvalue can be bounded as

$$|\lambda_2(\bar{M})| = \frac{\hat{m}}{m} |\lambda_2(M)| + 1 - \frac{\hat{m}}{m} \leq \frac{1-m}{1-m(1-\alpha)^2}.$$

It is clear that this is a monotonically decreasing function of α and m . That is, higher probability in updates and/or larger m results in faster convergence in average.

We are now ready to state the main result of this section.

Theorem 5.4: Consider the distributed scheme with simultaneous updates in (31). For any update probability $\alpha \in (0, 1]$, the PageRank value x^* is obtained through the time average y in (9) as $E[\|y(k) - x^*\|^2] \rightarrow 0, k \rightarrow \infty$.

The proof follows along similar lines as that in Theorem 3.4. More specifically, we can prove either by the general Markov chain results of, e.g., [11] or by Appendix B where we replace the expression of \bar{M} there with the one in Lemma 5.3. Hence, the convergence is of order $1/k$, as in the algorithm of Section III; it also depends on the update probability α but is independent of n .

We remark that this scheme is fully decentralized when $\alpha < 1$. It is parameterized by α , which determines the frequency in the updates, communication load among the pages, and the rate of convergence in the mean as we have seen above. In practice, the recursion in (31) must be implemented in the equivalent form (21). It is clear that communication is required only over the links corresponding to the nonzero entries in the link matrices there. Each page then computes weighted additions of its own value, the values received from others, and the constant \hat{m}/n . On the other hand, when $\alpha = 1$, the scheme reduces to the original centralized one in Section II. In this case, the distributed link matrix is $M_{1,\dots,1}$ and coincides with the original M because $\hat{m} = m$ and $A_{1,\dots,1} = A$ from Lemma 5.2.

VI. UPDATE TERMINATION IN PAGERANK COMPUTATION

In this section, we further develop the distributed algorithm for calculating the PageRank. We relax the objective and aim at obtaining approximate values of the PageRank. The key feature here is to allow the pages to terminate their updates at the point when the values have converged to a certain level. The benefit is that such values need to be transmitted only once to the linked pages; hence, the computation and communication load can be reduced.

In a centralized setting, the idea of update termination for the PageRank computation has been introduced by [22]. We extend this idea to the distributed update scheme of Section V. First, we consider a simple case to attain a convergence result. Then, we provide the details of the proposed algorithm.

A. Convergence Properties for the Distributed Scheme

Consider the distributed update scheme with simultaneous updates in (21) for computing the values $x(k)$ together with their time average $y(k)$. Within this subsection, we fix the sample paths $\{\eta_j(k)\}_{k=0}^{k_0-1}, j = 1, \dots, n$, up to time $k_0 - 1$ of the processes specifying the updates in the pages. Suppose that some of the time averages $y_i(k_0)$ have, in an approximate sense, converged. This is measured by finding those that have varied only within sufficiently small ranges for a certain number of time steps. We introduce two parameters: Let $\delta \in (0, 1)$ be the relative error level, and let N_s be the number of steps. Using the history of its own time average y_i , each page i then determines at time k_0 whether the following condition holds:

$$|y_i(k_0) - y_i(k_0 - \ell)| \leq \delta y_i(k_0), \quad \ell = 1, 2, \dots, N_s. \quad (34)$$

If so, then (i) the page i will terminate its update and fix its estimate at $y_i(k_0)$, and then (ii) this value $y_i(k_0)$ is transmitted to the pages connected to i by direct links. After this point, these values will be used at the pages performing further updates.

The question of interest is whether the pages that continue with their updates after time k_0 will reach a good estimate of their true values. In what follows, we show that the answer is positive and the approximation level achievable in the estimate will be as good as that for the pages that have terminated their updates. Note that the analysis is based on the given sample paths $\{\eta_j(k)\}_{k=0}^{k_0-1}$, and hence the state x and its average y up to time k_0 are fixed; we study their stochastic behaviors after this time.

Let $\mathcal{C}(k_0)$ be the set of page indices that have reached good estimates at time k_0 as

$$\mathcal{C}(k_0) := \{i \in \mathcal{V} : |y_i(k_0) - y_i(k_0 - \ell)| \leq \delta y_i(k_0), \ell = 1, 2, \dots, N_s\}.$$

The cardinality of this set is denoted by $n_1(k_0)$. We assume $n_1(k_0) \geq 1$. Also, let $\mathcal{N}(k_0) := \mathcal{V} \setminus \mathcal{C}(k_0)$.

Based on these sets, we introduce a coordinate transformation for the state $x(k)$ and partition it as

$$x(k) = \begin{bmatrix} x_{\mathcal{C}}(k) \\ x_{\mathcal{N}}(k) \end{bmatrix}, \quad k \geq k_0$$

where $x_{\mathcal{C}}(k) \in \mathbb{R}^{n_1(k_0)}$ contains the values of the pages in $\mathcal{C}(k_0)$ and $x_{\mathcal{N}}(k) \in \mathbb{R}^{n-n_1(k_0)}$ contains those of the pages in $\mathcal{N}(k_0)$. With slight abuse of notation, we write the transformed state by $x(k)$. Also, we use the shorthand notation A_p for A_{p_1, \dots, p_n} , $p_i \in \{0, 1\}, i \in \mathcal{V}$. Then, the distributed link matrices A_p in (22) and the average link matrix \bar{A} in (23) are partitioned accordingly as

$$A_p = \begin{bmatrix} A_{p,\mathcal{C}\mathcal{C}} & A_{p,\mathcal{C}\mathcal{N}} \\ A_{p,\mathcal{N}\mathcal{C}} & A_{p,\mathcal{N}\mathcal{N}} \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} \bar{A}_{\mathcal{C}\mathcal{C}} & \bar{A}_{\mathcal{C}\mathcal{N}} \\ \bar{A}_{\mathcal{N}\mathcal{C}} & \bar{A}_{\mathcal{N}\mathcal{N}} \end{bmatrix}. \quad (35)$$

Since the time average $y_{\mathcal{C}}$ has converged sufficiently by time k_0 , the proposed approach employs the value $y_{\mathcal{C}}(k_0)$ as $x_{\mathcal{C}}(k)$ for all $k \geq k_0$. Hence, the value at time k_0 is reset as

$$x(k_0) = \begin{bmatrix} y_{\mathcal{C}}(k_0) \\ x_{\mathcal{N}}(k_0) \end{bmatrix}.$$

The updates are carried out through the distributed algorithm given by

$$x(k+1) = \tilde{A}_{\eta(k)} x(k) + \frac{\hat{m}}{n} \tilde{s}, \quad k \geq k_0 \quad (36)$$

where

$$\begin{aligned} \tilde{A}_{\eta(k)} &= \begin{bmatrix} I & 0 \\ \tilde{A}_{\eta(k),\mathcal{N}\mathcal{C}} & \tilde{A}_{\eta(k),\mathcal{N}\mathcal{N}} \end{bmatrix} \\ &:= \begin{bmatrix} I & 0 \\ (1 - \hat{m})A_{\eta(k),\mathcal{N}\mathcal{C}} & (1 - \hat{m})A_{\eta(k),\mathcal{N}\mathcal{N}} \end{bmatrix}, \\ \tilde{s} &:= \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \end{aligned} \quad (37)$$

with $\mathbf{1} \in \mathbb{R}^{n-n_1(k_0)}$. We note that the matrices \tilde{A}_p are nonnegative, but are no longer stochastic; the sums of the entries of the first $n_1(k_0)$ columns are larger than 1 while those of the other columns are smaller than 1. Hence, though $x(k) \geq 0$ still holds, the state $x(k)$ may not be a probability vector. In addition, this scheme is in the distributed form of (21), and not the one in (31) based on the modified link matrices.

The time average $y(k)$ is also modified by fixing the entries for $i \in \mathcal{C}(k_0)$ as

$$y(k) = \begin{bmatrix} y_{\mathcal{C}}(k_0) \\ y_{\mathcal{N}}(k) \end{bmatrix}, \quad k \geq k_0$$

where $y_{\mathcal{N}}(k)$ is determined through the original formula (9).

For the approximate update scheme (36), its average state $\bar{x}(k) := E[x(k)]$ follows the recursion

$$\bar{x}(k+1) = \hat{A}\bar{x}(k) + \frac{\hat{m}}{n}\tilde{s}, \quad k \geq k_0 \quad (38)$$

where the average link matrix $\hat{A} := E[\tilde{A}_{\eta(k)}]$ is given by

$$\hat{A} = \begin{bmatrix} I & 0 \\ \hat{A}_{\mathcal{N}\mathcal{C}} & \hat{A}_{\mathcal{N}\mathcal{N}} \end{bmatrix} := \begin{bmatrix} I & 0 \\ (1-\hat{m})\bar{A}_{\mathcal{N}\mathcal{C}} & (1-\hat{m})\bar{A}_{\mathcal{N}\mathcal{N}} \end{bmatrix}. \quad (39)$$

Regarding this average link matrix, the following result will be useful in the subsequent development.

Lemma 6.1: The submatrix $\hat{A}_{\mathcal{N}\mathcal{N}}$ of the average link matrix \hat{A} as given in (39) satisfies the following:

- i) $\rho(\hat{A}_{\mathcal{N}\mathcal{N}}) \in [0, 1 - \hat{m}]$ and, in particular, $\hat{A}_{\mathcal{N}\mathcal{N}}$ is a stable matrix.
- ii) $(I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \geq 0$.

Proof:

- i) Since the original average link matrix \bar{A} in (35) is a stochastic matrix, the block diagonal matrix $\text{diag}(0, \bar{A}_{\mathcal{N}\mathcal{N}})$ containing the submatrix $\bar{A}_{\mathcal{N}\mathcal{N}}$ satisfies $\bar{A} \geq \text{diag}(0, \bar{A}_{\mathcal{N}\mathcal{N}}) \geq 0$. By the property of nonnegative matrices [15], it follows that $1 = \rho(\bar{A}) \geq \rho(\bar{A}_{\mathcal{N}\mathcal{N}}) \geq 0$. Therefore, $\rho(\hat{A}_{\mathcal{N}\mathcal{N}}) = \rho((1-\hat{m})\bar{A}_{\mathcal{N}\mathcal{N}}) \in [0, 1 - \hat{m}]$. By Lemma 5.3 (i), $\hat{m} \in (0, 1)$ and hence $\rho(\hat{A}_{\mathcal{N}\mathcal{N}}) < 1$.
- ii) Let $\lambda := 1 - \rho(\hat{A}_{\mathcal{N}\mathcal{N}})$. This is the eigenvalue of $I - \hat{A}_{\mathcal{N}\mathcal{N}}$ with the smallest real part. By (i), we have $\lambda > 0$. Thus, $I - \hat{A}_{\mathcal{N}\mathcal{N}}$ is an M -matrix,⁷ so it has an inverse that is nonnegative [16]. \square

We remark that in (i) in the lemma, the level of stability is affected by the parameter α as it determines the size of \hat{m} .

It is clear that the value vector x^* (with the coordinate change) is an equilibrium of the system (38). We partition it as

$$x^* = \begin{bmatrix} x_{\mathcal{C}}^* \\ x_{\mathcal{N}}^* \end{bmatrix}. \quad (40)$$

It is also simple to show that the vector $\tilde{x}(k_0)$ given by

$$\tilde{x}(k_0) = \begin{bmatrix} \tilde{x}_{\mathcal{C}}(k_0) \\ \tilde{x}_{\mathcal{N}}(k_0) \end{bmatrix} := \begin{bmatrix} y_{\mathcal{C}}(k_0) \\ (I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \left(\hat{A}_{\mathcal{N}\mathcal{C}} y_{\mathcal{C}}(k_0) + \frac{\hat{m}}{n} s \right) \end{bmatrix} \quad (41)$$

is an equilibrium of the system (38). This vector always exists by (ii) in the lemma above.

⁷A matrix $X \in \mathbb{R}^{n \times n}$ is said to be an M -matrix if its off-diagonal entries are nonpositive and all eigenvalues have positive real parts.

After the pages in $\mathcal{C}(k_0)$ have terminated their updates, the dynamics of the scheme can be characterized as follows.

Lemma 6.2: For the distributed update scheme (36) and its average system (38), the following statements hold.

- i) The average state $\bar{x}(k)$ converges to $\tilde{x}(k_0)$ and, in particular, $\bar{x}_{\mathcal{N}}(k) \rightarrow \tilde{x}_{\mathcal{N}}(k_0)$ as $k \rightarrow \infty$.
- ii) If $|\tilde{x}_{\mathcal{C}}(k_0) - x_{\mathcal{C}}^*| \leq \delta x_{\mathcal{C}}^*$, then $|\tilde{x}_{\mathcal{N}}(k_0) - x_{\mathcal{N}}^*| \leq \delta x_{\mathcal{N}}^*$.

Proof:

- i) Since $\tilde{x}(k_0)$ is an equilibrium of the average system (38), it follows that:

$$\bar{x}(k+1) - \tilde{x}(k_0) = \begin{bmatrix} I & 0 \\ \hat{A}_{\mathcal{N}\mathcal{C}} & \hat{A}_{\mathcal{N}\mathcal{N}} \end{bmatrix} (\bar{x}(k) - \tilde{x}(k_0)).$$

Here, we have $\bar{x}_{\mathcal{C}}(k) - \tilde{x}_{\mathcal{C}}(k_0) = 0$, $k \geq k_0$. Thus

$$\bar{x}_{\mathcal{N}}(k+1) - \tilde{x}_{\mathcal{N}}(k_0) = \hat{A}_{\mathcal{N}\mathcal{N}} (\bar{x}_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}(k_0)).$$

By Lemma 6.1 (i), $\hat{A}_{\mathcal{N}\mathcal{N}}$ is a stable matrix. Hence, we have $\bar{x}_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}(k_0) \rightarrow 0$ as $k \rightarrow \infty$.

- ii) For the average system (38), $\tilde{x}(k_0)$ and x^* are both equilibria, and thus

$$\tilde{x}_{\mathcal{N}}(k_0) - x_{\mathcal{N}}^* = (I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \hat{A}_{\mathcal{N}\mathcal{C}} (\tilde{x}_{\mathcal{C}}(k_0) - x_{\mathcal{C}}^*).$$

By Lemma 6.1 (ii), $(I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \geq 0$. Moreover, by construction, $\hat{A}_{\mathcal{N}\mathcal{C}} \geq 0$. Thus, using the assumption, we have

$$\begin{aligned} |\tilde{x}_{\mathcal{N}}(k_0) - x_{\mathcal{N}}^*| &\leq (I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \hat{A}_{\mathcal{N}\mathcal{C}} |\tilde{x}_{\mathcal{C}}(k_0) - x_{\mathcal{C}}^*| \\ &\leq (I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \hat{A}_{\mathcal{N}\mathcal{C}} \delta x_{\mathcal{C}}^* \\ &\leq \delta (I - \hat{A}_{\mathcal{N}\mathcal{N}})^{-1} \left(\hat{A}_{\mathcal{N}\mathcal{C}} x_{\mathcal{C}}^* + \frac{\hat{m}}{n} \mathbf{1} \right) = \delta x_{\mathcal{N}}^* \end{aligned}$$

where the last equality follows because x^* is an equilibrium of (38). \square

The lemma shows that if the values in $y_{\mathcal{C}}(k_0) = \tilde{x}_{\mathcal{C}}(k_0)$ are actually close to the true values $x_{\mathcal{C}}^*$, then via the recursion in (36), we can still obtain an approximate value $\tilde{x}_{\mathcal{N}}(k_0)$ in the average sense for all other states; the approximation level is the same as that for $\tilde{x}_{\mathcal{C}}(k_0)$, represented by the parameter δ .

The following is the main convergence result for the scheme described in this section.

Theorem 6.3: Consider the distributed scheme in (36), where under the given sample paths $\{\eta_i(k)\}_{k=0}^{k_0-1}$, $i = 1, \dots, n$, at time k_0 , the updates at $n_1(k_0)$ pages have terminated. The time average $y_{\mathcal{N}}(k)$, $k \geq k_0$, converges to the equilibrium $\tilde{x}_{\mathcal{N}}(k_0)$ in the mean-square sense as $E[\|y_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}(k_0)\|^2] \rightarrow 0$, $k \rightarrow \infty$.

The proof is presented in Appendix C. It is based on that for Theorem 3.4. However, unlike the setup there, the distributed link matrices in the current scheme are not stochastic. This means that we cannot employ general Markov process results of, e.g., [11]. In contrast, the proof relies on the stability of the submatrix $\hat{A}_{\mathcal{N}\mathcal{N}}$ as shown in Lemma 6.1 (i). Also, for this reason, the analysis does not involve the modified link matrices such as $M_{\eta_1(k), \dots, \eta_n(k)}$ that appeared in the previous section.

B. Distributed Algorithm With Update Termination

We present the distributed algorithm with update termination based on the results from this section.

Algorithm 6.4: For $i \in \mathcal{V}$, page i executes the following.

- 0) Initialize the parameters $n, \alpha, x_i(0), N_s$, and δ . Set $k = 0$, $\mathcal{C}(0) = \emptyset$, and $n_1(0) = 0$.
- 1) At time k , generate $\eta_i(k) \in \{0, 1\}$ under the probability α . If $\eta_i(k) = 1$, then send the value $x_i(k)$ to pages $j \notin \mathcal{C}(k)$ that it is linked to, and request pages $j \notin \mathcal{C}(k)$ having links to it for their values.
- 2) Update the value $x_i(k)$ and its time average by

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^n \left(\tilde{A}_{\eta(k)} \right)_{ij} x_j(k) + \frac{\hat{m}}{n}, \\ y_i(k) &= \frac{1}{k+1} \sum_{\ell=0}^k x_i(\ell) \end{aligned} \quad (42)$$

where $\tilde{A}_{\eta(k)}$ is constructed by (37) using $\mathcal{C}(k)$.

- 3) Check if $y_i(k)$ has sufficiently converged based on (34). If so, then (i) add i to the set $\mathcal{C}(k)$, (ii) send $y_i(k)$ to the pages having direct links to page i , and (iii) fix $x_i(\ell) = y_i(\ell) = y_i(k)$ for $\ell \geq k$.
- 4) If $\mathcal{C}(k) = \mathcal{V}$, then terminate the algorithm. Otherwise, set $\mathcal{C}(k+1) = \mathcal{C}(k)$ and $k = k+1$, and then go to Step 1. \square

We remark that, in this scheme, the choice of the parameters δ and N_s affects the accuracy in the values when the pages terminate their updates as well as the time when the pages decide to do so. Taking δ smaller and/or N_s larger will improve the value estimates, but will require longer time before the updates terminate; this in turn will keep the computation and communication load higher.

VII. DISCUSSION ON ASYNCHRONOUS ITERATION METHODS

In this section, we discuss the application of a numerical analysis method known as asynchronous iteration [5] to the distributed computation of PageRank values. Deterministic algorithms for the PageRank problem under this approach have been discussed in, e.g., [12], [24]. We present a randomization-based algorithm and clarify its relation to the schemes of this paper.

Consider the original update scheme in (21) based on the power method. Let $\eta_i(k) \in \{0, 1\}$ for $i \in \mathcal{V}$, $k \in \mathbb{Z}_+$, be the i.i.d. random processes whose distributions are as in (20). Similarly to the scheme with simultaneous updates in Section V, when $\eta_i(k) = 1$ at time k , then page i initiates an update; such an event occurs with probability α . However, the difference is that this update is performed as in the power method, and moreover pages whose corresponding processes $\eta_i(k)$ are zero do not make any updates.

The distributed update recursion is given as follows:

$$\tilde{x}(k+1) = \tilde{M}_{\eta_1(k), \dots, \eta_n(k)} \tilde{x}(k) \quad (43)$$

where the initial state $\tilde{x}(0)$ is a probability vector and the link matrices are given by

$$\left(\tilde{M}_{p_1, \dots, p_n} \right)_{ij} := \begin{cases} (1-m)a_{ij} + \frac{m}{n} & \text{if } p_i = 1, \\ 1 & \text{if } p_i = 0 \text{ and } i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (44)$$

It is clear that these matrices keep the rows of the original link matrix M in (3) for the pages that initiate updates. Other pages

just keep their previous values. Thus, these matrices are not stochastic.

The following result shows that through this algorithm, we can compute the PageRank values.

Lemma 7.1: Under the distributed update scheme of (43), for every initial state $x(0)$ that is a probability vector, the PageRank value x^* is obtained as $\tilde{x}(k) \rightarrow x^*$, $k \rightarrow \infty$, with probability one.

The distributed update scheme in (43) is a randomized version of the one in [5, Section 6.2], and the proof can be extended in a straightforward way. Specifically, it relies on the property $\rho(M) = 1$ that the link matrix M has. The algorithm is based on general asynchronous iteration algorithms for distributed computation of fixed points in the field of numerical analysis. It is interesting to note that the proof of the result above employs an argument similar to that of Lyapunov functions. We also point out that the convergence rate is exponential and the general scheme can handle delays in the communication.

In comparison, the algorithms proposed in this paper have the following characteristic features. First, the link matrices in the update schemes (8) and (21) are stochastic, and this property is exploited in the convergence analysis. It moreover provides the relation to consensus type problems as discussed in Section IV. Second, there is a difference with regard to the type of links over which communication takes place. In particular, it is shown in a subsequent paper [20] that the present approach can be extended in such a way that each page communicates only with those connected by outgoing links; the information of such links are by default available locally. By contrast, in the asynchronous iteration algorithm, pages must utilize the incoming links. This means that popular pages linked from many pages need extra storage to keep the data of such links.

VIII. NUMERICAL EXAMPLE

We present an example with 1,000 web pages ($n = 1,000$). The links among the pages were randomly generated. The first ten pages are designed to have high rankings and are given links from over 90% of the pages. Others have between 2 and 333 links per page.

We ran Algorithm 6.4 in Section VI where each page initiates an update at a fixed probability $\alpha = 0.01$ and terminates the updates when an approximate value is obtained. In the distributed scheme (42), we generated sample paths of the processes $\eta_i(k)$, $i \in \mathcal{V}$, which determine the pages initiating updates, and computed the state $x(k)$. The initial state $x(0)$ was taken as a random probability vector. The parameters for the update termination were chosen as follows: The number N_s of steps before stopping the update was $N_s = 800$ and the parameter δ determining the level of approximation was $\delta = 0.01$. We chose these values so that the characteristics of this scheme are visible in the plots.

The responses of the time average y_i for $i = 21, \dots, 30$ are shown in Fig. 2. The times when the corresponding pages terminated their updates are marked by \circ . We observe that the convergence is fairly fast, and the updates stop by time $k = 4,500$ for these pages.

Let the errors in the estimates be $e(k) := y(k) - x^*$. In Fig. 3, these are shown for two cases: $\|e(k)\|_1$ under the ℓ_1 norm and $\|e(k)\|_\infty$ under the ℓ_∞ norm. The plot shows that the error in the

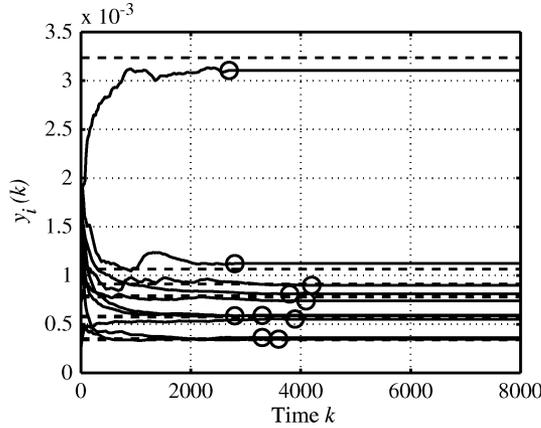


Fig. 2. Sample paths of y_i (solid lines) with the times at which updates stopped (marked by \circ for each page) and the true PageRank values x_i^* (dashed lines) for $i = 21, \dots, 30$.

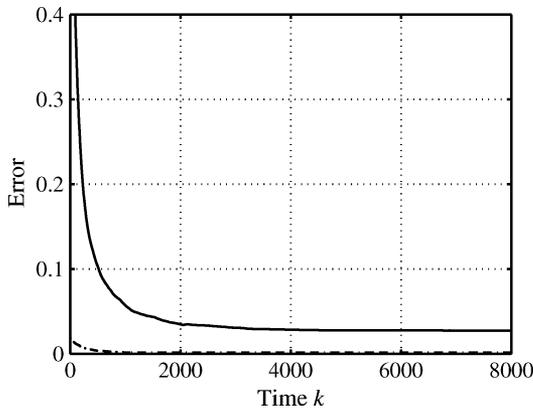


Fig. 3. Estimation error: $\|e(k)\|_1$ (solid line) and $\|e(k)\|_\infty$ (dash-dot line).

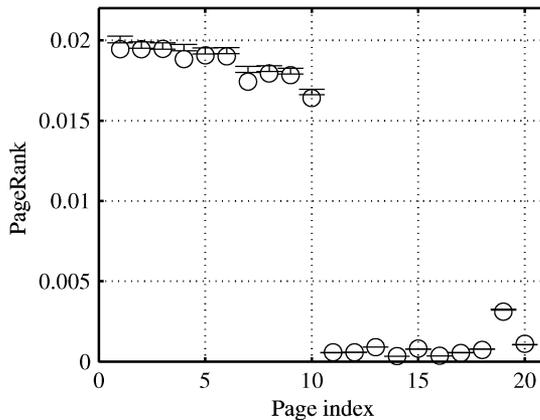


Fig. 4. Ranges of approximate PageRank values (marked by two lines) and y_i for $i = 1, \dots, 20$, at $k = 8,000$ (marked as \circ).

individual values of y_i (measured by the ℓ_∞ norm) rapidly decreases and remains small while the total error (in the ℓ_1 norm) also decreases but at a slower rate.

In Fig. 4, the final values of y_i for the first twenty pages are plotted as \circ together with the acceptable ranges of error, that is, $[(1 - \delta)x_i^*, (1 + \delta)x_i^*]$ by two lines connected in the middle. As we mentioned in Section VI, the time average y is no longer normalized in this case. However, the sum of all y_i at $k = 8,000$ turned out to be about 0.989, which is very close to the desired value 1.

IX. CONCLUSION

In this paper, we first gave an overview of the PageRank computation problem, which is critical in making accurate search results with Google. We introduced a randomization-based distributed approach for the computation of PageRank values and showed the mean-square convergence of the proposed schemes. It was demonstrated that the approach has a clear relation to consensus type problems. The algorithms were generalized in the recent papers [19], [20], where random link failures and computations based on aggregating groups of pages are addressed and more discussions on the advantages of this approach can be found. Future research will deal with the effects of communication delays and the improvement of convergence rate, and also study issues related to implementation of the proposed distributed algorithms.

APPENDIX

In this appendix, following some preliminary material, the proofs of Theorems 3.4 and 6.3 are given.

A. Preliminaries

We present some results related to infinite products of stochastic matrices from [13], [32]. These are required for the proof of Theorem 3.4 given in the next subsection.

First, the notion of weak ergodicity is introduced.

Definition A.1: Given a sequence of stochastic matrices $\{P(k)\}_{k=0}^\infty \subset \mathbb{R}^{n \times n}$, let their (backward) product be $T(k) := P(k) \cdots P(0)$. The sequence $\{P(k)\}_{k=0}^\infty$ is said to be weakly ergodic if

$$t_{ri}(k) - t_{rj}(k) \rightarrow 0, \quad k \rightarrow \infty, \quad \forall i, j, r \in \{1, \dots, n\}. \quad (45)$$

In (45), $t_{ri}(k)$ for all i tend to be equal as $k \rightarrow \infty$, that is, all columns of the product matrix $T(k)$ coincide in the limit. However, in general, the columns do not converge to a single vector.

To characterize matrix sequences that are weakly ergodic, we employ the tool known as the coefficient of ergodicity. Let $\tau(\cdot)$ be the scalar function for stochastic matrices in $\mathbb{R}^{n \times n}$ given by

$$\tau(P) := \frac{1}{2} \max_{i,j} \sum_{r=1}^n |p_{ri} - p_{rj}|. \quad (46)$$

This function $\tau(\cdot)$ has the following properties.

Lemma A.2:

- i) $\tau(P) \in [0, 1]$ and, moreover, $\tau(P) = 0$ if and only if there exists a probability vector $v \in \mathbb{R}^n$ such that $P = v\mathbf{1}^T$, where $\mathbf{1} = [1 \dots 1]^T \in \mathbb{R}^n$.
- ii) $\tau(P) = \max\{\|Px\|_1 : x \in \mathbb{R}^n, \|x\|_1 = 1, \sum_i x_i = 0\}$.
- iii) $\tau(PQ) \leq \tau(P)\tau(Q)$ for stochastic matrices P, Q .

Weak ergodicity can be characterized by the following lemma.

Lemma A.3: For a sequence of stochastic matrices $\{P(k)\}_{k=0}^\infty$, their product $P(k) \cdots P(0)$ is weakly ergodic if $\tau(P(k)) \leq \tau_0$ for all k , where $\tau_0 \in (0, 1)$ is a scalar.

B. Distributed Update Scheme

We now analyze the proposed algorithm in Section III.

Lemma A.4: For the distributed update scheme (14) and its average system (16), the following hold.

- i) The matrices $M_i, i \in \{1, \dots, n\}$, and \bar{M} satisfy $\tau(M_i) < 1 - \hat{m}$ and $\tau(\bar{M}) < 1 - \hat{m}$.
- ii) For any mode sequence $\{\theta(k)\}$, the matrix sequence $\{M_{\theta(k)}\}$ is weakly ergodic.

Proof:

- i) We show only for \bar{M} since the case of M_i is similar. Recall that by (13) and (15), the average matrix \bar{M} can be expressed as $\bar{M} = (1 - \hat{m})\bar{A} + (\hat{m}/n)S$. Thus, we have

$$\begin{aligned} \tau(\bar{M}) &= \frac{1}{2} \max_{i,j} \sum_{r=1}^n |(\bar{M})_{ri} - (\bar{M})_{rj}| \\ &= (1 - \hat{m}) \frac{1}{2} \max_{i,j} \sum_{r=1}^n |(\bar{A})_{ri} - (\bar{A})_{rj}| \\ &= (1 - \hat{m})\tau(\bar{A}). \end{aligned}$$

The matrix \bar{A} is stochastic, and hence, by Lemma A.2 (i), it holds that $\tau(\bar{A}) \leq 1$. Consequently, we arrive at the inequality $\tau(\bar{M}) \leq 1 - \hat{m}$.

- ii) This follows from (i) and Lemma A.3 since $\hat{m} \in (0, 1)$ by Lemma 3.3 (i). □

Proof of Theorem 3.4: Let the error from the average be $e(k) := x(k) - x^*$. Note that $e(k)$ satisfies $\sum_{i=1}^n e_i(k) = 0$. This is because in the systems (14) and (16), by assumption, the initial states are probability vectors, and furthermore, $M_i, i = 1, \dots, n$, and \bar{M} are stochastic matrices; hence, both $x(k)$ and x^* are nonnegative vectors whose entries add up to 1.

Observe that

$$y(k) - x^* = \frac{1}{k+1} \sum_{\ell=0}^k (x(\ell) - x^*) = \frac{1}{k+1} \sum_{\ell=0}^k e(\ell).$$

Thus

$$\begin{aligned} &E \left[\|y(k) - x^*\|^2 \right] \\ &= E \left[\left\| \frac{1}{k+1} \sum_{\ell=0}^k e(\ell) \right\|^2 \right] \\ &= \frac{1}{(k+1)^2} E \left[\sum_{\ell=0}^k e(\ell)^T e(\ell) + 2 \sum_{\ell=0}^{k-1} \sum_{r=1}^{k-\ell} e(\ell)^T e(\ell+r) \right] \\ &= \frac{1}{(k+1)^2} \left\{ \sum_{\ell=0}^k E [e(\ell)^T e(\ell)] \right. \\ &\quad \left. + 2 \sum_{\ell=0}^{k-1} \sum_{r=1}^{k-\ell} E [e(\ell)^T e(\ell+r)] \right\}. \end{aligned} \quad (47)$$

We use the norm relation $\|z\| \leq \|z\|_1$ for $z \in \mathbb{R}^n$ [15] and the property $\|x(\ell)\|_1 = \|x^*\|_1 = 1$ to obtain the bound $\|e(\ell)\| \leq 2$. Then, in the first summation term in (47), we have

$$\sum_{\ell=0}^k E [e(\ell)^T e(\ell)] \leq 4(k+1). \quad (48)$$

In the second summation term, we see that the summands can be written as

$$\begin{aligned} &E [e(\ell)^T e(\ell+r)] \\ &= E [e(\ell)^T (x(\ell+r) - x^*)] \\ &= E [e(\ell)^T (M_{\theta(\ell+r-1)} \cdots M_{\theta(\ell)} x(\ell) - x^*)]. \end{aligned} \quad (49)$$

Here, by taking the expectation of the matrix product $M_{\theta(\ell+r-1)} \cdots M_{\theta(\ell)}$ with respect to the random variables $\theta(\ell+r-1), \dots, \theta(\ell)$

$$\begin{aligned} &E [e(\ell)^T e(\ell+r)] \\ &= E [e(\ell)^T (E [M_{\theta(\ell+r-1)} \cdots M_{\theta(\ell)}] x(\ell) - x^*)] \\ &= E [e(\ell)^T (E [M_{\theta(\ell+r-1)}] \cdots E [M_{\theta(\ell)}] x(\ell) - x^*)] \\ &= E [e(\ell)^T (\bar{M}^r x(\ell) - x^*)] \end{aligned} \quad (50)$$

where the second and third equalities follow from the independence of $\theta(\ell+r-1), \dots, \theta(\ell)$ and the definition of the average matrix \bar{M} , respectively. Since, by Lemma 3.3 (iii), x^* is the eigenvector of \bar{M} for the eigenvalue 1, it follows that $\bar{M}^r x(\ell) - x^* = \bar{M}^r (x(\ell) - x^*)$. Further, we have $x(\ell) - x^* = e(\ell)$ and again apply the fact $\|z\| \leq \|z\|_1, z \in \mathbb{R}^n$, to derive from (50) that

$$\begin{aligned} E [e(\ell)^T e(\ell+r)] &= E [e(\ell)^T \bar{M}^r e(\ell)] \\ &\leq E [\|e(\ell)\| \|\bar{M}^r e(\ell)\|] \\ &\leq 2E [\|\bar{M}^r e(\ell)\|_1] \end{aligned} \quad (51)$$

where in the last inequality, we also used $\|e(\ell)\| \leq 2$. As we have mentioned above, it holds that $\sum_{i=1}^n e_i(\ell) = 0$. Thus, apply Lemma A.2 (ii) and (iii) to (51) and obtain

$$\begin{aligned} E [e(\ell)^T e(\ell+r)] &\leq 2\tau(\bar{M}^r) E [\|e(\ell)\|_1] \\ &\leq 2\tau(\bar{M})^r E [\|e(\ell)\|_1] \leq 4\tau(\bar{M})^r \end{aligned} \quad (52)$$

where the last inequality is due to $\|e(\ell)\|_1 \leq 2$. Note that by Lemma A.4, $\tau(\bar{M}) < 1$.

Finally, by substituting (48) and (52) into (47), we have

$$\begin{aligned} E [\|y(k) - x^*\|^2] &\leq \frac{1}{(k+1)^2} \\ &\quad \left\{ 4(k+1) + 2 \sum_{\ell=0}^{k-1} \sum_{r=0}^{k-\ell} 4\tau(\bar{M})^r \right\} \\ &\leq \frac{4}{k+1} \left(1 + \frac{2}{1 - \tau(\bar{M})} \right) \end{aligned}$$

and hence using the bound on $\tau(\bar{M})$ in Lemma A.4 (i), we obtain

$$E [\|y(k) - x^*\|^2] \leq \frac{4(2 + \hat{m})}{\hat{m}(k+1)} \rightarrow 0, \quad k \rightarrow \infty. \quad (53)$$

Thus, the PageRank value x^* is obtained through the time average y . □

Proof of Theorem 6.3: For simplicity, let the initial time of the update scheme (36) to be $k_0 + 1 = 0$. Further, we write $\tilde{x}_{\mathcal{N}}$ for $\tilde{x}_{\mathcal{N}}(k_0)$. Denote the error between the state and the equilibrium by $e(k) := x_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}$. Then

$$y_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}} = \frac{1}{k+1} \sum_{\ell=0}^k (x_{\mathcal{N}}(\ell) - \tilde{x}_{\mathcal{N}}) = \frac{1}{k+1} \sum_{\ell=0}^k e(\ell).$$

Thus

$$E \left[\|y_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}\|^2 \right] = \frac{1}{(k+1)^2} \left\{ \sum_{\ell=0}^k E [e(\ell)^T e(\ell)] + 2 \sum_{\ell=0}^{k-1} \sum_{r=1}^{k-\ell} E [e(\ell)^T e(\ell+r)] \right\}. \quad (54)$$

In what follows, we must evaluate the two summation terms on the right-hand side.

First, we claim that $e(k)$ is uniformly bounded and in particular, for each $k \geq 0$

$$\|e(k)\|_1 \leq \|x_{\mathcal{N}}(0)\|_1 + \frac{\varepsilon}{\hat{m}} + \|\tilde{x}_{\mathcal{N}}\|_1 =: \gamma \quad (55)$$

where $\varepsilon := \max_{p \in \{0,1\}^n} \|\tilde{A}_{p,\mathcal{N}\mathcal{C}} \tilde{x}_{\mathcal{C}} + \hat{m}/n \mathbf{1}\|_1$. Notice that

$$\|e(k)\|_1 = \|x_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}\|_1 \leq \|x_{\mathcal{N}}(k)\|_1 + \|\tilde{x}_{\mathcal{N}}\|_1. \quad (56)$$

From the distributed update law in (36), it easily follows that:

$$x_{\mathcal{N}}(k+1) = \tilde{A}_{\eta(k),\mathcal{N}\mathcal{N}} x_{\mathcal{N}}(k) + \tilde{A}_{\eta(k),\mathcal{N}\mathcal{C}} \tilde{x}_{\mathcal{C}} + \frac{\hat{m}}{n} \mathbf{1}.$$

By definition, $\tilde{A}_{\eta(k),\mathcal{N}\mathcal{N}} = (1 - \hat{m}) A_{\eta(k),\mathcal{N}\mathcal{N}}$ and $A_{\eta(k),\mathcal{N}\mathcal{N}}$ is a submatrix of the stochastic matrix $A_{\eta(k)}$. Consequently, we have $\|\tilde{A}_{\eta(k),\mathcal{N}\mathcal{N}}\|_1 \leq 1 - \hat{m}$. Using this bound, we obtain

$$\begin{aligned} \|x_{\mathcal{N}}(k+1)\|_1 &\leq \left\| \tilde{A}_{\eta(k),\mathcal{N}\mathcal{N}} \right\|_1 \cdot \|x_{\mathcal{N}}(k)\|_1 \\ &\quad + \left\| \tilde{A}_{\eta(k),\mathcal{N}\mathcal{C}} \tilde{x}_{\mathcal{C}} + \frac{\hat{m}}{n} \mathbf{1} \right\|_1 \\ &\leq (1 - \hat{m}) \|x_{\mathcal{N}}(k)\|_1 + \varepsilon. \end{aligned}$$

Thus

$$\begin{aligned} \|x_{\mathcal{N}}(k)\|_1 &\leq (1 - \hat{m})^k \|x_{\mathcal{N}}(0)\|_1 + \varepsilon \sum_{\ell=0}^{k-1} (1 - \hat{m})^\ell \\ &\leq \|x_{\mathcal{N}}(0)\|_1 + \frac{\varepsilon}{\hat{m}}. \end{aligned}$$

Therefore, substituting this into (56), we have shown (55).

Now, with the bound (55), the first summation term in (54) can be upper bounded as

$$\sum_{\ell=0}^k E [e(\ell)^T e(\ell)] \leq (k+1) \gamma^2. \quad (57)$$

We next look at the the second summation term of (54). The summands can be written as

$$\begin{aligned} E [e(\ell)^T e(\ell+r)] &= E [e(\ell)^T [0 \ I] (x(\ell+r) - \tilde{x})] \\ &= E \left[e(\ell)^T [0 \ I] \left(\tilde{A}_{\eta(\ell+r-1)} \cdots \tilde{A}_{\eta(\ell)} x(\ell) \right. \right. \\ &\quad \left. \left. + \sum_{j=\ell}^{\ell+r-1} \tilde{A}_{\eta(\ell+r-1)} \cdots \tilde{A}_{\eta(j+1)} \frac{\hat{m}}{n} \tilde{s} - \tilde{x} \right) \right]. \end{aligned}$$

Here, by taking the expectation of the matrix products $\tilde{A}_{\eta(\ell+r-1)} \cdots \tilde{A}_{\eta(\ell+j)}$, $j = 0, 1, \dots, r-1$, with respect to the random variables $\eta(\ell+r-1), \dots, \eta(\ell+j)$, we have

$$\begin{aligned} E [e(\ell)^T e(\ell+r)] &= E [e(\ell)^T [0 \ I] \left(E [\tilde{A}_{\eta(\ell+r-1)} \cdots \tilde{A}_{\eta(\ell)}] x(\ell) \right. \\ &\quad \left. + \sum_{j=\ell}^{\ell+r-1} E [\tilde{A}_{\eta(\ell+r-1)} \cdots \tilde{A}_{\eta(j+1)}] \frac{\hat{m}}{n} \tilde{s} - \tilde{x} \right)] \\ &= E [e(\ell)^T [0 \ I] \left(E [\tilde{A}_{\eta(\ell+r-1)}] \cdots E [\tilde{A}_{\eta(\ell)}] x(\ell) \right. \\ &\quad \left. + \sum_{j=\ell}^{\ell+r-1} E [\tilde{A}_{\eta(\ell+r-1)}] \cdots E [\tilde{A}_{\eta(j+1)}] \frac{\hat{m}}{n} \tilde{s} - \tilde{x} \right)] \\ &= E \left[e(\ell)^T [0 \ I] \left(\hat{A}^r x(\ell) + \sum_{j=\ell}^{\ell+r-1} \hat{A}^{\ell+r-1-j} \frac{\hat{m}}{n} \tilde{s} - \tilde{x} \right) \right] \end{aligned}$$

where the second and third equalities, respectively, follow from the independence of $\eta(\ell+r-1), \dots, \eta(\ell+j)$ and the definition of \hat{A} in (39). As \tilde{x} is an equilibrium of the average system in (38), it can be shown that $E[e(\ell)^T e(\ell+r)] = E[e(\ell)^T [0 \ I] \hat{A}^r (x(\ell) - \tilde{x})]$. Here, we have $x(\ell) - \tilde{x} = [0^T e(\ell)^T]^T$ and use the norm relation $\|z\| \leq \|z\|_1$ for all $z \in \mathbb{R}^n$ [15] to derive

$$\begin{aligned} E [e(\ell)^T e(\ell+r)] &= E \left[e(\ell)^T [0 \ I] \hat{A}^r \begin{bmatrix} 0 \\ e(\ell) \end{bmatrix} \right] \\ &= E [e(\ell)^T \hat{A}_{\mathcal{N}\mathcal{N}}^r e(\ell)] \\ &\leq E \left[\|e(\ell)\| \cdot \left\| \hat{A}_{\mathcal{N}\mathcal{N}}^r e(\ell) \right\| \right] \\ &\leq E \left[\|e(\ell)\|_1 \cdot \left\| \hat{A}_{\mathcal{N}\mathcal{N}}^r e(\ell) \right\|_1 \right] \\ &\leq E \left[\|e(\ell)\|_1^2 \cdot \left\| \hat{A}_{\mathcal{N}\mathcal{N}} \right\|_1 \right] \\ &\leq \gamma^2 \|\hat{A}_{\mathcal{N}\mathcal{N}}\|_1^r \end{aligned} \quad (58)$$

where in the last inequality, we used $\|e(\ell)\|_1 \leq \gamma$ from (55). The average matrix $\hat{A}_{\mathcal{N}\mathcal{N}}$ can be bounded as $\|\hat{A}_{\mathcal{N}\mathcal{N}}\|_1 \leq 1 - \hat{m}$ because $\hat{A}_{\mathcal{N}\mathcal{N}} = (1 - \hat{m}) \bar{M}_{\mathcal{N}\mathcal{N}}$, and $\bar{M}_{\mathcal{N}\mathcal{N}}$ is a submatrix of a stochastic matrix. Thus, we arrive at

$$E [e(\ell)^T e(\ell+r)] \leq \gamma^2 (1 - \hat{m})^r. \quad (59)$$

Finally, by substituting (57) and (59) into (54) and by $\hat{m} \in (0, 1)$ from Lemma 5.3 (i)

$$\begin{aligned} E \left[\|y_{\mathcal{N}}(k) - \tilde{x}_{\mathcal{N}}\|^2 \right] & \\ & \leq \frac{1}{(k+1)^2} \left\{ (k+1)\gamma^2 + 2 \sum_{\ell=0}^{k-1} \sum_{r=1}^{k-\ell} \gamma^2 (1-\hat{m})^r \right\} \\ & \leq \frac{\gamma^2}{k+1} \left(1 + 2 \frac{1-\hat{m}}{\hat{m}} \right). \end{aligned}$$

The far right-hand side converges to zero as $k \rightarrow \infty$, which completes the proof. \square

ACKNOWLEDGMENT

The authors wish to thank E.-W. Bai, B. Ross Barmish, T. Başar, F. Dabbene, S. Dasgupta, S. Hara, Z. Qu, M. Vidyasagar, and Y. Yamamoto, for their helpful comments and discussions on this work.

REFERENCES

- [1] "Special section on complex networked control systems," *IEEE Control Syst. Mag.*, vol. 27, no. 4, 2007.
- [2] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, V. S. Mirrokni, and S.-H. Teng, "Local computation of PageRank contributions," in *Algorithms and Models for the Web-Graph*. Berlin, Germany: Springer, 2007, vol. 4863, Lect. Notes Comp. Sci., pp. 150–165.
- [3] P. J. Antsaklis and J. Baillieul, Eds., "Special issue on the technology of networked control systems," *Proc. IEEE*, vol. 95, no. 1, Jan. 2007.
- [4] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte Carlo methods in PageRank computation: When one iteration is sufficient," *SIAM J. Numer. Anal.*, vol. 45, pp. 890–904, 2007.
- [5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [6] D. P. Bertsekas and J. N. Tsitsiklis, "Comments on "Coordination of groups of mobile autonomous agents using nearest neighbor rules";" *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 968–969, May 2007.
- [7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
- [8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks ISDN Syst.*, vol. 30, pp. 107–117, 1998.
- [9] K. Bryan and T. Leise, "The \$25,000,000,000 eigenvector: The linear algebra behind Google," *SIAM Rev.*, vol. 48, pp. 569–581, 2006.
- [10] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Symmetries in the coordinated consensus problem," in *Networked Embedded Sensing and Control Workshop*, P. J. Antsaklis and P. Tabuada, Eds. Berlin, Germany: Springer, 2005, vol. 331, Lect. Notes Contr. Info. Sci., pp. 25–51.
- [11] R. Cogburn, "On products of random stochastic matrices," *Contemporary Math.*, vol. 50, pp. 199–213, 1986.
- [12] D. V. de Jager and J. T. Bradley, "Asynchronous iterative solution for state-based performance metrics," in *Proc. ACM SIGMETRICS*, 2007, pp. 373–374.
- [13] D. J. Hartfiel, *Markov Set-Chains*. Berlin, Germany: Springer, 1998, vol. 1695, Lect. Notes Math..
- [14] Y. Hatano and M. Mesbahi, "Agreement over random networks," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1867–1872, Nov. 2005.
- [15] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
- [16] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
- [17] H. Ishii and R. Tempo, "A distributed randomized approach for the PageRank computation: Parts 1 and 2," in *Proc. 47th IEEE Conf. Decision Control*, 2008, pp. 3523–3528.
- [18] H. Ishii and R. Tempo, "Computing the PageRank variation for fragile web data," *SICE J. Control, Meas., Sys. Integr.*, vol. 2, pp. 1–9, 2009.
- [19] H. Ishii and R. Tempo, "Distributed PageRank computation with link failures," in *Proc. Amer. Control Conf.*, 2009, pp. 1976–1981.
- [20] H. Ishii, R. Tempo, E.-W. Bai, and F. Dabbene, "Distributed randomized PageRank computation based on web aggregation," in *Proc. 48th IEEE Conf. Decision Control*, 2009, pp. 3026–3031.
- [21] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
- [22] S. Kamvar, T. Haveliwala, and G. Golub, "Adaptive methods for the computation of PageRank," *Linear Algebra Appl.*, vol. 386, pp. 51–65, 2004.
- [23] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, pp. 1192–1203, 2007.
- [24] G. Kollias, E. Gallopoulos, and D. B. Szyld, "Asynchronous iterative computations with Web information retrieval structures: The PageRank case," in *Parallel Computing: Current and Future Issues of High-End Computing*, G. R. Joubert, Ed. et al. Jülich, Germany: John von Neumann-Institut für Computing, 2006, vol. 33, NIC Series, pp. 309–316.
- [25] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton, NJ: Princeton Univ. Press, 2006.
- [26] Z. Lin, B. A. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Trans. Autom. Control*, vol. 50, no. 1, pp. 121–127, Jan. 2005.
- [27] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 1963–1974, Nov. 2004.
- [28] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.
- [29] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York: McGraw Hill, 2002.
- [30] Z. Qu, J. Wang, and R. A. Hull, "Cooperative control of dynamical systems with application to autonomous vehicles," *IEEE Trans. Autom. Control*, vol. 53, no. 4, pp. 894–911, May 2008.
- [31] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.
- [32] E. Seneta, *Non-Negative Matrices and Markov Chains*. New York: Springer, 1981.
- [33] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 791–795, Apr. 2008.
- [34] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. London, U.K.: Springer, 2005.
- [35] R. Tempo and H. Ishii, "Monte Carlo and Las Vegas randomized algorithms for systems and control: An introduction," *Eur. J. Control*, vol. 13, pp. 189–203, 2007.
- [36] C. W. Wu, "Synchronization and convergence of linear dynamics in random directed networks," *IEEE Trans. Autom. Control*, vol. 51, no. 7, pp. 1207–1210, Jul. 2006.
- [37] C. Yu, J. M. Hendrickx, B. Fidan, B. D. O. Anderson, and V. D. Blondel, "Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence," *Automatica*, vol. 43, pp. 387–402, 2007.
- [38] Y. Zhu, S. Ye, and X. Li, "Distributed PageRank computation based on iterative aggregation-disaggregation methods," in *Proc. 14th ACM Conf. Inform. Knowledge Manag.*, 2005, pp. 578–585.



Hideaki Ishii (S'97–M'02) received the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2002.

He was a Postdoctoral Research Associate of the Coordinated Science Laboratory, the University of Illinois at Urbana-Champaign from 2001 to 2004, and a Research Associate of the Department of Information Physics and Computing, The University of Tokyo, Tokyo, Japan, from 2004 to 2007. He

is currently an Associate Professor of the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama, Japan. He is a co-author of the book *Limited Data Rate in Control Systems with Networks* (Berlin, Germany: Springer, 2002, vol. 275, Lecture Notes in Control and Information Sciences). He serves as an Associate Editor for *Automatica*. His research interests are in networked control systems, multi-agent systems, hybrid systems, and probabilistic algorithms.

Dr. Ishii is an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.



Roberto Tempo (F'00) is currently a Director of Research of Systems and Computer Engineering at the IEIT Institute, National Research Council (CNR), Politecnico di Torino, Italy. He has held visiting positions at various institutions, including Kyoto University, University of Illinois at Urbana-Champaign, German Aerospace Research Organization in Oberpfaffenhofen and Columbia University in New York. He is author or co-author of more than 150 research papers published in international journals, books and conferences. He is also a co-author of the book *Randomized Algorithms for Analysis and Control of Uncertain Systems*, (Berlin, Germany: Springer-Verlag, 2005). He is currently an Editor and Deputy Editor-in-Chief of *Automatica*. His research activities are mainly focused on analysis and design of complex systems with uncertainty, and related control applications.

Dr. Tempo received the "Outstanding Paper Prize Award" from the International Federation of Automatic Control (IFAC) for a paper published in *Automatica*, and the "Distinguished Member Award" from the IEEE Control Systems Society. He is a Fellow of the IFAC. He has been Editor for Technical Notes and Correspondence of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He has served as Program Chair of the first joint IEEE Conference on Decision and Control and European Control Conference, Seville, Spain, 2005. He is the 2010 President of the IEEE Control Systems Society.