Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

A Distributed Randomized Approach for the PageRank Computation: Part 1

Hideaki Ishii
Department of Computational Intelligence & Systems Science
Tokyo Institute of Technology, Yokohama 226-8502, Japan
ishii@dis.titech.ac.jp

Roberto Tempo
IEIIT-CNR, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
roberto.tempo@polito.it

Abstract— **In the search engine of Google, the PageRank algorithm plays a crucial role in ranking the obtained results. The algorithm quantifies the importance of each web page based on the link structure of the web. In this two-part paper, 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. 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, 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], [20], [21]

One of the main features of the PageRank algorithm is that it is based solely on the link structure of 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. This clearly necessitates more efficient computational methods.

In this regard, several approaches have recently been proposed. In [18], 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 examined in [3], while the work in [29] exploits the block structure of the web to apply techniques from the Markov chain literature. In [13], methods using asynchronous iterations [11] are discussed.

In this two-part paper with [16], we follow a distributed approach and, in particular, we develop several randomized algorithms for the PageRank computation; for recent advances on probabilistic approaches in systems and control, see [25]. These schemes are fully distributed and have 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. In this first part, we give a simple scheme and show its convergence properties. In the second part [16], this scheme is extended to improve its applicability.

We emphasize that the approach proposed here is particularly motivated by the recent research on multi-agent consensus, agreement, and formation problems [5], [7], [10], [14], [17], [19], [24], [26]–[28]. For additional details, we refer to [1], [2], [4] and also to the paper [6] which gives a summary of recent development of such problems along with some new results. Among such problems, the PageRank is especially related to the consensus where multiple agents exchange their values with neighboring agents so that all agents have the same value. The objective is clearly different from that for 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 of the same form as that for the

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under Grant-in-Aid for Scientific Research No. 17760344

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

PageRank computation except that the class of stochastic matrices is slightly different. We will make a comparison of the two problems in some detail.

The paper is organized as follows: In Section II, we provide an overview of the PageRank algorithm. The distributed approach is introduced in Section III. The proposed scheme is given and analyzed in Section IV; this is followed by a discussion on consensus problems in Section V. A numerical example is provided in Section VI to show the effectiveness of the scheme. We conclude the paper in Section VII.

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 $i = 1, \ldots, n$ and $j = 1, \ldots, m$; 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 $j = 1, \ldots, n$. Let $\mathbf{1} \in \mathbb{R}^n$ be the vector with all entries equal to 1 as $\mathbf{1} := [1 \ \cdots \ 1]^T$. 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 $\rho(X)$.

II. THE PAGERANK PROBLEM

We provide a brief introduction of the PageRank problem; this material can be found in, e.g., [8], [9], [20], [21].

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, $\mathcal{V} := \{1, 2, \ldots, n\}$ 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 equivalently, page j has an incoming link from page i. To simplify the discussion, we assume $n \geq 3$.

The objective of the PageRank algorithm is to provide some measure of importance to each web page based on the link structure of the web. 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 by the values is that a page having links from important pages is also important. This is realized by determining the value of a page as the sum of the contributions from all pages having 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 set of page indices that are 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, the PageRank problem can be restated as

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

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

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

Note that the value vector x^* is a nonnegative unit eigenvector corresponding to the eigenvalue 1 of the link matrix A. In general, however, for this eigenvector to exist and then to be unique, it is sufficient that the following conditions hold: (i) The so-called dangling nodes, which are pages having no links to others, do not exist, and (ii) the web as a graph is strongly connected[1], or equivalently the matrix A is irreducible[2].

In the real web, actually, dangling nodes are abundant. Such pages can be found, e.g., in the form of PDF document files having no outgoing links. Note that these pages introduce zero columns into the link matrix. To simplify the discussion, we redefine the graph (and the matrix A) by bringing in artificial links for all dangling nodes. As a result, the link matrix A becomes a stochastic matrix, that is, $\sum_{i=1}^n a_{ij} = 1$ for all j. This implies that it has at least one eigenvalue equal to 1.

The web is also known not to be strongly connected in general. To avoid this problem, 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

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

Notice that M is a positive stochastic matrix, which makes it an irreducible matrix. By the Perron-Frobenius Theorem [15], there exists a unique positive eigenvector for the eigenvalue 1. Hence, we redefine the value vector x^* by using M in place of A in (1) as

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

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

[1] A 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.

[2] An irreducible matrix is a matrix that is not reducible. A matrix $X \in \mathbb{R}^{n\times n}$ is said to be reducible if either (i) $n = 1$ and $X = 0$ or (ii) $n \geq 2$ and there exist a permutation matrix $P \in \mathbb{R}^{n\times n}$ and an integer r with $1 \leq r \leq n - 1$ such that

$$P^T XP = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix},$$

where $B \in \mathbb{R}^{r\times r}$, $C \in \mathbb{R}^{r\times(n-r)}$, and $D \in \mathbb{R}^{(n-r)\times(n-r)}$.

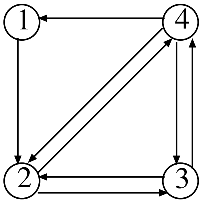

Fig. 1. A web with four pages

is based on the power method. That is, the value vector x^* is computed through the recursion

$$x(k+1) = Mx(k), \tag{5}$$

where $x(k) \in \mathbb{R}^n$ and the initial condition $x(0) \in \mathbb{R}^n$ is a probability vector. Using this method, we can asymptotically find the value vector as shown below; see, e.g., [15].

Lemma 2.1: For any initial condition $x(0)$, in the update scheme (5) using the modified link matrix M, it holds that $x(k) \to x^*$ as $k \to \infty$.

We next provide a simple example for illustration.

Example 2.2: 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 can easily be constructed by (2) as

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 1 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}. \tag{6}$$

For the modified link matrix M, we use $m = 0.15$, which is the value reported in the original algorithm [8]. Thus,

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

Then, the value vector x^* can be computed as

$$x^* = \begin{bmatrix} 0.119 & 0.331 & 0.260 & 0.289 \end{bmatrix}^T.$$

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 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 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

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

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

We now consider the distributed update scheme in the following form:

$$x(k+1) = A_{\theta(k)}x(k), \tag{8}$$

where $x(k) \in \mathbb{R}^n$ is the state whose initial condition $x(0)$ is a probability vector; $\theta(k) \in \{1, \dots, n\}$ is the mode of the system, and $A_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, n$, are called the distributed link matrices.

The objective here is to design the distributed update scheme (8) so that the PageRank values are computed through the time average of x. For this purpose, let $y(k)$ be the time average of the sample path $x(0), \dots, x(k)$ given by

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

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

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

This type of convergence is called ergodicity for stochastic processes [22]. In the next section, we specify the A_i matrices. We note that these are related to the original link matrix A rather than its modified version M. This approach allows us to simplify the discussion.

The problem is closely related to fixed point computations, where distributed methods known as asynchronous iterations have been developed [4], [13]. As we will see, the difference is that our approach is motivated by consensus problems and the use of randomization.

IV. PROPOSED SCHEME AND ITS ANALYSIS

A. Distributed link matrices

We now introduce the distributed link matrices A_i. For $i = 1, \dots, n$, the matrix $A_i \in \mathbb{R}^{n \times n}$ is obtained as follows: (i) The ith 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 equal to zero. More formally, we have

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

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

Example 4.1: We continue with the 4-page web of Example 2.2. The link matrices A_i can be found to be

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{2}{3} \end{bmatrix},$$
$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{2}{3} \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

To see why we employ the particular structure for the A_i matrices in (11), we may consider another set of these matrices. For example, let A_i be constructed by simply using the ith row of the original matrix A and 1 in the diagonal entries of other rows as follows:

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

This scheme can be realized in a distributed way. However, it will not lead us to the true values. In fact, each time an update takes place, one of the values is completely lost because each A_i contains a column with all zeros. ▽

B. The average link matrix

To clarify the properties of the distributed link matrices A_i, we consider the distributed update scheme in (8) and, in particular, its average dynamics. For this purpose, let $\overline{A}$ be the average matrix given by $\overline{A} := E[A_{\theta(k)}]$, where $E[\,\cdot\,]$ is the expectation with respect to the random variable $\theta(k)$. Due to the probability distribution of $\theta(k)$ in (7), we have

$$\overline{A} = \frac{1}{n} \sum_{i=1}^{n} A_i. \tag{12}$$

This matrix $\overline{A}$ is stochastic since all A_i are so.

The following lemma shows some properties of the matrix $\overline{A}$ that are useful in subsequent developments.

Lemma 4.2: For the average matrix $\overline{A}$ given in (12), we have the following:

(i) $\overline{A} = \frac{2}{n} A + \frac{n-2}{n} I$.

(ii) There exists $z_0 \in \mathbb{R}^n_+$ which is an eigenvector corresponding to the eigenvalue 1 for both A and $\overline{A}$.

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

C. A modified distributed update scheme

As in the case with the original link matrix A, for the average matrix $\overline{A}$, 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 version of the distributed link matrices.

To this end, we first consider the modified distributed update scheme given by

$$x(k+1) = M_{\theta(k)} x(k), \tag{13}$$

where $x(k) \in \mathbb{R}^n$, the initial condition $x(0)$ is a probability vector, and the mode $\theta(k) \in \mathcal{V}$ is specified in (7).

The problem at this stage is as follows: Find the modified link matrices M_i 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 $\overline{x}(k) := E[x(k)]$ be the mean of the state $x(k)$ of the system (13), where the expectation is with respect to $\theta(0), \ldots, \theta(k-1)$. Its dynamics is then expressed as

$$\overline{x}(k+1) = \overline{M}\,\overline{x}(k), \tag{14}$$

where $\overline{x}(0) = x(0)$ and the average matrix $\overline{M}$ is given by

$$\overline{M} := E[M_{\theta(k)}] = \frac{1}{n} \sum_{i=1}^{n} M_i. \tag{15}$$

A simple way of defining M_i would be, as in the case with M, via the relation $M_i = (1-m)A_i + \frac{m}{n} S$. However, in this case, it can be shown that there is no clear relation between the original matrix M and the average matrix $\overline{M}$ such as that between A and $\overline{A}$ as seen in Lemma 4.2.

Therefore, we introduce an additional parameter $\hat{m} \in (0,1)$, and let the matrices M_i be defined as

$$M_i := (1-\hat{m}) A_i + \frac{\hat{m}}{n} S, \quad i = 1, \ldots, n. \tag{16}$$

Note that M_i and hence $\overline{M}$ in (15) are positive stochastic matrices. Specifically, let the parameter $\hat{m}$ be given by

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

For this choice of $\hat{m}$, the following result holds.

Lemma 4.3: For the scalar $\hat{m}$ given in (17) and the average matrix $\overline{M}$ in (15), we have the following:

(i) $\hat{m} \in (0,1)$ and $\hat{m} < m$.

(ii) $\overline{M} = \frac{\hat{m}}{m} M + \left(1 - \frac{\hat{m}}{m}\right) I$.

(iii) The value vector x^* is the unique eigenvector of the average matrix $\overline{M}$ corresponding to the eigenvalue 1.

From the lemma, the value vector x^* can be obtained by the power method, i.e., by the average system in (14) as

$$\overline{x}(k) \to x^*, \quad k \to \infty. \tag{18}$$

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 multiple matrices M_i.

However, this property turns out not to be sufficient to guarantee convergence of $x(k)$ to the true value x^*. From an argument based on weak ergodicity [23], we can show that for any sequence $\{\theta(k)\}$, there exists a sequence of probability vectors $\{v(k)\}$ such that, for any $x(0)$, $x(k) - v(k)s^T x(0) = x(k) - v(k) \to 0$ as $k \to \infty$. The vectors $v(k)$ in general do not converge. 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 the paper. It shows that the time average indeed converges to the value vector in the mean-square sense.

Theorem 4.4: In the distributed update scheme in (13), the PageRank value x^* is obtained through the time average y in (9) as $E\left[\left\|y(k) - x^*\right\|^2\right] \to 0$, $k \to \infty$.

The theorem shows that the proposed distributed update scheme has an ergodic property. This theorem can be shown by general Markov process results in, e.g., [12]. We also note that the convergence is of order $1/k$.

We have several remarks. In practice, each page needs to communicate with the pages that are directly connected by incoming or outgoing links. In this regard, it is important to note that the recursion in (13) can be expressed as

$$x(k+1) = (1 - \hat{m})A_{\theta(k)}x(k) + \frac{\hat{m}}{n}\mathbf{1}. \tag{19}$$

This is because for any k, $x(k)$ is a probability vector, and thus $Sx(k) = \mathbf{1}$. Hence, at time k, communication is required only among the pages corresponding to the nonzero entries in the distributed link matrix $A_{\theta(k)}$ (and not those in $M_{\theta(k)}$). As can be seen in (19), each page then performs weighted addition of its own value, the values just received, and the extra term $\hat{m}/n$. Hence, we observe that the amount of computation required at each page is limited at any time.

V. Relation 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., [14], [24], [27]; see also [26].

Consider a set $\mathcal{V} = \{1, 2, \ldots, n\}$ 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.

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

$$x(k+1) = A_{\theta(k)}x(k), \tag{20}$$

where $\theta(k) \in \{1, \ldots, d\}$ 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 $i \in \{1, \ldots, d\}$ 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 $\mathrm{Prob}\{\theta(k) = i\} = 1/d$ for $i \in \{1, \ldots, d\}$, $k \in \mathbb{Z}_+$.

We say that consensus is obtained if for any initial condition $x(0)$, it holds that

$$|x_i(k) - x_j(k)| \to 0, \quad k \to \infty \tag{21}$$

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

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

where $\mathcal{L}_{ij} := \{\ell : (\ell, j) \in \mathcal{E}_i\}$ is the set of agents that transmit their values to agent j and n_{ij} is its cardinality.

Lemma 5.1: Under the scheme of (20), consensus is obtained in the sense of (21).

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

(i) The graph is assumed to be strongly connected.
(ii) The goal is that all values $x_i(k)$ become equal. The values need not converge to a constant (according to (21)), 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.
(iv) The matrices A_i are row stochastic. The consensus problem can be restated in terms of infinite product of stochastic matrices. Hence, the coefficient of ergodicity is useful; see, e.g., [24].

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 neighboring agents.
(2) At the technical level, it is especially important to impose stochasticity on the distributed link matrices.

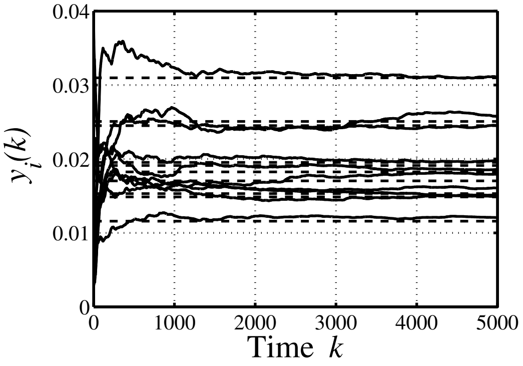

Fig. 2. Sample paths of $y_i(k)$ (solid lines) and the PageRank values x_i^* (dashed lines) for $i = 1, 2, \ldots, 10$.

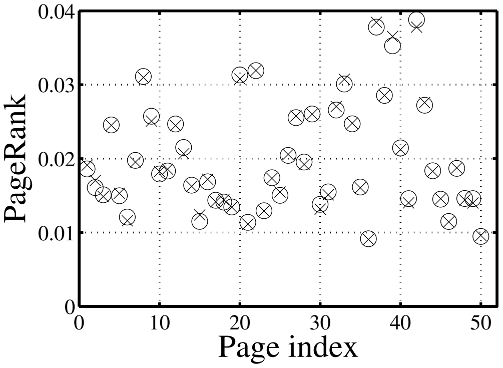

Fig. 3. The PageRank values x_i^* (marked as $\times$) and $y_i(k)$, $i = 1, \ldots, 5$, at $k = 5000$ (marked as $\bigcirc$)

For the distributed PageRank computation, very few works exploit such viewpoints.

VI. Numerical example

We present an example with 50 web pages ($n = 50$). The links among the pages were randomly generated, and for each page, there were between 2 and 13 links. The parameter m of the matrices M in (3) and M_i in (16) was taken as $m = 0.15$. Note that the probability of a page to update at time k is $1/n = 0.02$.

We computed a sample path of the state $x(k)$ of the distributed scheme (13). The initial state $x(0)$ was taken as a probability vector that was randomly generated. In Fig. 2, the time average $y(k)$ is shown for the ten pages $i = 1, \ldots, 10$. We observe their convergence to the PageRank values, which are shown in dashed lines. In Fig. 3, the PageRank values are marked as $\times$ and the values of $y_i(k)$ at time $k = 5,000$ are plotted as $\bigcirc$. We see that the errors are fairly small.

VII. Conclusion

In this paper, we have first given an overview of the PageRank problem which is critical in making accurate search results at Google. The main result is the distributed computation approach based on randomization at each page. We also clarified its relations to the consensus type problems. Further extensions of this approach are presented in [16].

Acknowledgement: We are thankful to B. Ross Barmish, Tamer Başar, Shinji Hara, Zhihua Qu, and Yutaka Yamamoto for their helpful comments and discussions on this work.

References

[1] Special section on complex networked control systems. *IEEE Control Systems Magazine*, 27(4), 2007.

[2] P.J. Antsaklis and J. Baillieul, Guest Editors. Special Issue on the Technology of Networked Control Systems. *Proc. IEEE*, 95(1), 2007.

[3] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte Carlo methods in PageRank computation: When one iteration is sufficient. *SIAM J. Numer. Anal.*, 45:890–904, 2007.

[4] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[5] D.P. Bertsekas and J.N. Tsitsiklis. Comments on "Coordination of groups of mobile autonomous agents using nearest neighbor rules". *IEEE Trans. Autom. Control*, 52:968–969, 2007.

[6] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. 44th IEEE Conf. on Decision and Control and European Control Conf.*, pages 2996–3000, 2005.

[7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inform. Theory*, 52:2508–2530, 2006.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comp. Networks & ISDN Systems*, 30:107–117, 1998.

[9] K. Bryan and T. Leise. The $25,000,000,000 eigenvector: The linear algebra behind Google. *SIAM Rev.*, 48:569–581, 2006.

[10] R. Carli, F. Fagnani, M. Focoso, A. Speranzon, and S. Zampieri. Symmetries in the coordinated consensus problem. In P. J. Antsaklis and P. Tabuada, editors, *Networked Embedded Sensing and Control Workshop*, volume 331 of *Lect. Notes Contr. Info. Sci.*, pages 25–51. Springer, Berlin, 2005.

[11] D. Chazan and W. L. Miranker. Chaotic relaxation. *Linear Algebra and its Applications*, 2:199–222, 1969.

[12] R. Cogburn. On products of random stochastic matrices. *Contemporary Mathematics*, 50:199–213, 1986.

[13] D. V. de Jager and J. T. Bradley. Asynchronous iterative solution for state-based performance metrics. In *Proc. ACM SIGMETRICS*, pages 373–374, 2007.

[14] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Trans. Autom. Control*, 50:1867–72, 2005.

[15] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge Univ. Press, 1985.

[16] H. Ishii and R. Tempo. A distributed randomized approach for the PageRank computation: Part 2. In *Proc. 47th IEEE Conf. on Decision and Control*, 2008.

[17] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, 48:988–1001, 2003.

[18] S. Kamvar, T. Haveliwala, and G. Golub. Adaptive methods for the computation of PageRank. *Linear Algebra Appl.*, 386:51–65, 2004.

[19] A. Kashyap, T. Başar, and R. Srikant. Quantized consensus. *Automatica*, 43:1192–1203, 2007.

[20] A. N. Langville and C. D. Meyer. A survey of eigenvector methods for Web information retrieval. *SIAM Rev.*, 47:135–161, 2005.

[21] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton Univ. Press, 2006.

[22] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes,* 4th edition. McGraw Hill, New York, 2002.

[23] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer, New York, 1981.

[24] A. Tahbaz-Salehi and A. Jadbabaie. A necessary and sufficient condition for consensus over random networks. *IEEE Trans. Autom. Control*, 53:791–795, 2008.

[25] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems*. Springer, London, 2005.

[26] R. Tempo and H. Ishii. Monte Carlo and Las Vegas randomized algorithms for systems and control: An introduction. *European J. Control*, 13:189–203, 2007.

[27] C. W. Wu. Synchronization and convergence of linear dynamics in random directed networks. *IEEE Trans. Autom. Control*, 51:1207–1210, 2006.

[28] 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*, 43:387–402, 2007.

[29] Y. Zhu, S. Ye, and X. Li. Distributed PageRank computation based on iterative aggregation-disaggregation methods. In *Proc. 14th ACM Conf. Info. and Knowledge Management*, pages 578–585, 2005.