2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Distributed PageRank Computation with Link Failures

Hideaki Ishii and Roberto Tempo

Abstract— **The Google search engine employs the so-called PageRank algorithm for ranking the search results. This algorithm quantifies the importance of each web page based on the link structure of the web. In this paper, we continue our recent work on distributed randomized computation of PageRank, where the pages locally determine their values by communicating with linked pages. In particular, we propose a distributed randomized algorithm with limited information, where only part of the linked pages is required to be contacted. This is useful to enhance flexibility and robustness in computation and communication.**

Index Terms— **Distributed computation, Link failures, Multi-agent consensus, PageRank algorithm, Randomization, Stochastic matrices**

I. INTRODUCTION

The performance of search engines heavily relies on the capability of listing search results so that users can quickly have access to the desired information. One effective and objective way to quantify the importance or popularity of the web pages is by simply examining the link structure of the web. The so-called PageRank algorithm at Google follows such an idea and ranks pages higher when they have links from more important pages (see, e.g., [3], [4], [22]).

To execute the PageRank algorithm, however, the size of the web poses serious difficulties. Google is said to have over 8 billion web page indices and moreover computes the PageRank in a centralized fashion. In view of the rapid growth of the web, it is critical to develop more efficient computational methods. In this regard, a line of current research is towards distributed computation of the PageRank. In [28], block structures in the web are exploited to apply Markov chain methods while the work of [1] utilizes techniques from Monte Carlo simulation. In [7], [21], the application of numerical analysis methods known as asynchronous iterations [2] is discussed. Other works include [20], where adaptive methods allocate computational resources depending on the rate of convergence.

In our recent papers [16], [17], we developed a distributed randomized approach for the PageRank computation; for recent advances on probabilistic methods in systems and control, see [25]. The approach is distributed in that each page computes 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 to which it links and those linked to it. Randomization is with respect to the time that each page decides to initiate the communication. The time is randomly chosen and is independent among the pages. Hence, there is no need of a fixed order among the pages or a leader agent that specifies the pages to start updates. It is also stressed that relatively small communication and computation are required for the agents. On the other hand, in [18], we considered a centralized scheme for computing the bounds on the PageRank values when the web data contains uncertainties.

In this paper, we further explore the approach of [16], [17] to enhance flexibility and robustness under limited information. Specifically, we are interested in situations where each page initiating an update contacts only part of its linked pages. We continue to work in the probabilistic setting, and such pages are determined in a random manner. The links not used for communication at the time of updates will be referred to as the *failing* links. This feature would be useful, for example, when the computation/communication load among the pages must be reduced, but the rate of updates should be kept at the same level. In this respect, this scheme is more flexible than that in [17] because in addition to the the rate of updates for each page, the rate for link selection may be specified. Another situation where this algorithm can be applied is when communication is unreliable due to link failures and/or packet losses. In such a case, it may not be possible to contact all linked pages at the same time. The algorithm from our previous work does not function in this case and thus exhibits lack of robustness. A simple way to model packet losses is to consider them as an outcome of Bernoulli random processes. This approach has been widely adopted in the field of networked control as well as consensus; see, for example, [8]–[10], [13]–[15], [23]. This model of unreliable channels can be incorporated into the proposed scheme of the paper.

As discussed in [16], it is important to note that the proposed distributed randomized approach has been motivated by the recent development in the multi-agent problems. This aspect is exploited in this paper as well. In particular, among the many works in this field, our approach has strong ties with the stochastic versions of the consensus problems (e.g., [5], [11], [24], [26], [27]). From the viewpoint of consensus, it is natural to treat the web as a network of agents capable of local computation as well as communication with neighbors. It is further emphasized that there are similarities at the technical level. In the algorithm for PageRank computation with link failures, stochastic matrices play a crucial role, but in a slightly different form than consensus problems.

The organization of this paper is as follows: We first provide an overview of the PageRank problem in Section II. This is followed by Section III, where we summarize the distributed approach of [16], [17]. In Section IV, we present

Hideaki Ishii is with the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, 4259-J2-54, Nagatsuta-cho, Midori-ku, Yokohama 226-8502, Japan. E-mail: ishii@dis.titech.ac.jp

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

978-1-4244-4524-0/09/$25.00 ©2009 AACC

a distributed algorithm which allows for link failures and prove its convergence to the PageRank values. The results are illustrated through a numerical example in Section V. Finally, in Section VI, concluding remarks are given.

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 $\sum_{i=1}^n v_i = 1$. 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. 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 $\rho(X)$. For a discrete set $\mathcal{D}$, its cardinality is given by $|\mathcal{D}|$.

II. The PageRank problem

The PageRank problem is now briefly described based on, e.g., [3], [4], [22].

Consider a network of n web pages indexed from 1 to n. The 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 in other words, page j has an incoming link from page i.

The objective of the PageRank algorithm is to assign 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 denoted by $x_i^* \in [0, 1]$. The values are ordered: $x_i^* > x_j^*$ implies that page i is more important than page j.

The pages are ranked according to the rule that a page having links from important pages is also important. This is done in such a way that the value of one page equals the sum of the contributions from all pages that have links to it. Specifically, we define the value of page i by

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

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 as $\sum_{i=1}^n x_i^* = 1$.

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

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

The value vector x^* is a nonnegative unit eigenvector corresponding to the eigenvalue 1 of A. In general, however, for this eigenvector to exist and then to be unique, it is sufficient that (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].

To simplify the issue regarding dangling nodes, we redefine the graph by bringing in artificial links. As a result, the link matrix A becomes a stochastic matrix, having at least one eigenvalue equal to 1.

The web is known not to be strongly connected in general. To avoid this problem, a modified version of the values has been introduced in [3] as follows: Let $m \in (0, 1)$, and let the modified link matrix $M \in \mathbb{R}^{n \times n}$ be defined by

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

This matrix is clearly positive and is also stochastic being a convex combination of two stochastic matrices A and S/n. By the Perron-Frobenius Theorem [12], 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

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

We note that in the original paper [3], a typical value for m is chosen as $m = 0.15$. This value will be employed in the rest of the paper.

Because of the large dimension of the link matrix M, the computation of the value vector x^* relies on the power method. That is, x^* is computed through the recursion

$$\begin{aligned} x(k+1) &= Mx(k) \\ &= (1 - m)Ax(k) + \frac{m}{n}\mathbf{1}, \end{aligned} \tag{5}$$

where $x(k) \in \mathbb{R}^n$ and the initial condition $x(0) \in \mathbb{R}^n$ is a probability vector. Notice that the second equality can be established because both A and M are stochastic matrices.

The following lemma shows that, using this method, we can asymptotically find the value vector (e.g., [12]).

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 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 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 [22] that the structure of the link matrix M leads us to the bound

$$|\lambda_2(M)| \leq 1 - m. \tag{6}$$

For the value $m = 0.15$, the asymptotic rate of convergence is bounded by 0.85.

[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.

III. Distributed randomized approach

In this section, we summarize the distributed randomized algorithm for computing the PageRank values from [16], [17]. This approach is the basis for the computation scheme proposed in Section IV.

Consider the web with n pages described in Section II. The basic protocol employed in this scheme is as follows: At each time k, the page i initiates its PageRank value update (i) by sending its value to the pages to which it is linked and (ii) by requesting the pages that link to it for their values. All pages involved here update their values based on the newly available information.

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. The probability α is however a global parameter, and all pages in the web use the same value.

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

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

for $k \in \mathbb{Z}_+$, where their probability distributions are specified by the probability α as

$$\alpha = \text{Prob}\{\eta_i(k) = 1\}. \quad (7)$$

The process $\eta_i(k)$ is generated at the corresponding page i, and when its value is 1, then the page will follow the protocol outlined above so that an update is initiated. Let $\eta(k) := [\eta_1(k) \ \cdots \ \eta_n(k)]$ be the notation in a vector form.

Now, consider the distributed update scheme given by

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

where $x(k) \in \mathbb{R}^n$ is the state whose initial condition satisfies $x(0) \geq 0$ and $\sum_{i=1}^{n} x_i(0) = 1$; $\hat{m} \in (0,1)$ is the parameter used instead of m in the centralized case in (5). In particular, we take

$$\hat{m} = \frac{[1-(1-\alpha)^2]m}{1-m(1-\alpha)^2}. \quad (9)$$

The distributed link matrices A_q for $q \in \{0,1\}^n$ are given as follows:

$$\left(A_q\right)_{ij} := \begin{cases} a_{ij} & \text{if } q_i = 1 \text{ or } q_j = 1, \\ 1 - \sum_{h:\ q_h = 1} a_{hj} & \text{if } q_i = 0 \text{ and } i = j, \\ 0 & \text{if } q_i = q_j = 0 \text{ and } i \neq j, \end{cases} \quad i,j \in \mathcal{V}. \quad (10)$$

Clearly, there are 2^n such matrices. They have the property that (i) if $q_i = 1$, then the ith column and the ith row are the same as those in the original link matrix A, (ii) if $q_i = 0$, then the ith diagonal entry is chosen so that the entries of the ith column add up to 1, and (iii) all other entries are 0. Hence, these are constructed as stochastic matrices.

In this scheme, each page i also computes the time average of its own state x_i. Let $y(k)$ be the average of the sample path $x(0),\ldots,x(k)$ as

$$y(k) := \frac{1}{k+1}\sum_{\ell=0}^{k} x(\ell), \quad k \in \mathbb{Z}_+. \quad (11)$$

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 follows:

$$E\left[\left\|y(k) - x^*\right\|^2\right] \to 0, \quad k \to \infty. \quad (12)$$

This type of convergence is known as ergodicity for stochastic processes.

For completeness, we restate the main result of [17].

Theorem 3.1: Consider the distributed update scheme in (8). For any update probability $\alpha \in (0,1]$, the PageRank value x^* is obtained through the time average y as in (12).

We comment on the distributed update scheme described above. This scheme can be implemented decentrally. This can be seen in the expression (8), where it is clear that each page communicates only with those pages sharing direct links. Such links correspond to the nonzero entries of the link matrix A. The parameter α determines the probability of the updates to occur and thus the necessary load of communication among the pages. At page i, the amount of computation is fairly small: The update requires weighted additions of its own value x_i, the values x_j from the linked pages, and the constant $\hat{m}/n$; also the time average y_i is computed.

This distributed update scheme can also be viewed as a generalization of the original centralized scheme (5) in Section II. By using the update probability of $\alpha = 1$, all pages initiate their updates at all times. In this case, we have $\eta(k) \equiv [1 \ \cdots \ 1]$ and thus, the distributed link matrix $A_{\eta(k)}$ is equal to the original A. Furthermore, the parameter $\hat{m}$ coincides with m.

IV. A distributed scheme with link failures

In this section, we extend the distributed approach to handle the situation where only part of the links are used for communication each time a page initiates an update. That is, we examine how an update can be carried out when not all values from the linked pages are available; we say that link failures occur in this case. We continue to work in the probabilistic setting and assume that such links are randomly selected. This scheme would be useful when the communication load among the pages must be reduced or when some pages cannot be reached because of link failures and/or packet losses.

The set of failing links where no communication takes place at time k is denoted by $\Delta(k)$. This is a subset of the edges that link to or from the pages initiating the updates; we denote such a set by $\mathcal{E}_{\eta(k)}$, which is formally defined by

$$\mathcal{E}_q := \left\{(i,j) \in \mathcal{E} :\ q_i = 1 \text{ or } q_j = 1\right\}, \quad q \in \{0,1\}^n. \quad (13)$$

For the set $\Delta(k)$ at time k, we assume that if $(i,j) \in \Delta(k)$ and $(j,i) \in \mathcal{E}_{\eta(k)}$, then $(j,i) \in \Delta(k)$ for $(i,j) \in \mathcal{E}_{\eta(k)}$. This represents symmetry in the link failures; if a link from one page to another is failing at time k, then the link in the other direction between these pages must be failing as well. The set $\Delta(k)$ is a random process specified by the link failure probability $\delta \in [0,1)$ under the probability distribution

$$\delta = \text{Prob}\{(i,j) \in \Delta(k) \mid \eta(k) = q\}, \quad \forall (i,j) \in \mathcal{E}_q,\ q \in \{0,1\}^n,\ k \in \mathbb{Z}_+. \tag{14}$$

This shows that the links through which information of other pages is not transmitted are probabilistically selected under a fixed probability. Such a distribution for link failures has been employed in the context of networked control and consensus with limited information; see, e.g., [8]–[10], [13]–[15], [23].

To take account of the set $\Delta(k)$ of failing links, consider the distributed update scheme given by

$$x(k+1) = (1-\hat{m}) A_{\eta(k),\Delta(k)} x(k) + \frac{\hat{m}}{n}\mathbf{1}, \tag{15}$$

where $x(k) \in \mathbb{R}^n$, the initial condition $x(0) \geq 0$ satisfies $\sum_{i=1}^n x_i(0) = 1$, and $\hat{m} \in (0,1)$ is the parameter used instead of m in the centralized case. The matrices $A_{q,\mathcal{D}}$ for $q \in \{0,1\}^n$ and $\mathcal{D} \subset \mathcal{E}_q$ are the distributed link matrices with link failures.

The objective here is to design this distributed update scheme by finding the appropriate link matrices $A_{q,\mathcal{D}}$ and the parameter $\hat{m}$ so that the PageRank values are computed through the time average y of the state x. We follow an approach similar to that in Section III and, in particular, construct the link matrices so that they possess the stochastic property.

A. Distributed link matrices and their average

The first step in the design is to introduce the distributed link matrices and analyze their properties.

Let the distributed link matrix with link failures be given as follows:

$$\left(A_{q,\mathcal{D}}\right)_{ij} := \begin{cases} 0 & \text{if } (j,i) \in \mathcal{D}, \\ (A_q)_{ij} & \text{if } (j,i) \notin \mathcal{D} \\ & \quad \text{and } i \neq j, \\ 1 - \sum_{\substack{h \in \mathcal{V},\ h \neq j \\ (j,h) \notin \mathcal{D}}} (A_q)_{hj} & \text{if } i = j \end{cases} \tag{16}$$

for $q \in \{0,1\}^n$, $\mathcal{D} \subset \mathcal{E}_q$, and $i,j \in \mathcal{V}$. Note that by definition, $(i,i) \notin \mathcal{E}_q$ for any i and q.

By the definition of link failures, if the link $(j,i) \in \mathcal{E}$ is failing, then the (i,j) entry of the link matrix must be equal to zero. The link matrices defined above take account of such zero entries, but are still designed to be stochastic. This property is critical in showing the convergence of the scheme. In practice, this structure implies that if page j initiates an update and sends its value to page h over a link that is potentially failing, it must know whether page h received the value (and used it for its own update) or not. This can be observed in the (j,j) entry in (16) since it consists of the (h,j) entry of A_q.

We now analyze the average dynamics of the distributed update scheme determined by the link matrices just introduced. We define the average link matrix by

$$\overline{A} := E\big[A_{\eta(k),\Delta(k)}\big], \tag{17}$$

where $E[\,\cdot\,]$ is the expectation with respect to the processes $\eta(k)$ and $\Delta(k)$. This matrix $\overline{A}$ is nonnegative and stochastic because all $A_{\eta(k),\Delta(k)}$ share this property.

The following proposition shows that the average link matrix $\overline{A}$ has a clear relation to the original link matrix A.

Proposition 4.1: (i) The average link matrix $\overline{A}$ given in (17) can be expressed as

$$\overline{A} = \big[1 - \delta - (1-\delta)(1-\alpha)^2\big] A + \big[\delta + (1-\delta)(1-\alpha)^2\big] I. \tag{18}$$

(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 $\overline{A}$.

B. Mean-square convergence of the distributed scheme

In order to show the convergence property of the distributed update scheme, we now introduce the modified version of the link matrices. First, we rewrite the update scheme of (15) in its equivalent form as

$$x(k+1) = M_{\eta(k),\Delta(k)} x(k), \tag{19}$$

where the matrices $M_{q,\mathcal{D}}$ for $q \in \{0,1\}^n$ and $\mathcal{D} \subset \mathcal{E}_q$ are given by

$$M_{q,\mathcal{D}} := (1-\hat{m}) A_{q,\mathcal{D}} + \frac{\hat{m}}{n} S. \tag{20}$$

These matrices are called the modified distributed link matrices. This equivalent form of (19) can be obtained because the link matrices A_q are stochastic matrices; thus, the state $x(k)$ remains a probability vector for all k, which implies $Sx(k) \equiv \mathbf{1}$.

Also, let the average matrix of $M_{\eta(k),\Delta(k)}$ be

$$\overline{M} := E[M_{\eta(k),\Delta(k)}]. \tag{21}$$

Here, the distributed link matrices are positive stochastic matrices, which means that the average matrix $\overline{M}$ enjoys the same property.

The next step in designing the update scheme is to determine the parameter $\hat{m}$. The specific aim here is to show that the average of the modified distributed link matrices 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^*.

Similarly to the case in Section III, the parameter $\hat{m}$ is chosen differently from m in the centralized scheme. Let $\hat{m}$ be given by[2]

$$\hat{m} = \frac{\big[1 - \delta - (1-\delta)(1-\alpha)^2\big] m}{1 - m\big[\delta + (1-\delta)(1-\alpha)^2\big]}. \tag{22}$$

[2]Recall that the parameter m is chosen to be $m = 0.15$ in this paper.

The next lemma states an important property regarding the modified link matrices for this value $\hat{m}$.

Lemma 4.2: The parameter $\hat{m}$ in (22) and the average link matrices $\overline{M}$ in (21) have the following properties:

(i) $\hat{m} \in (0,1)$ and $\hat{m} \leq m$.
(ii) $\overline{M} = \frac{\hat{m}}{m} M + \left(1 - \frac{\hat{m}}{m}\right) I$.
(iii) The value x^* in (4) is the unique eigenvector of the average matrix $\overline{M}$ corresponding to the eigenvalue 1.

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 $E[x(k)] = \overline{M}^k x(0) \to x^*$ as $k \to \infty$.

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

Theorem 4.3: Consider the distributed scheme with link failures in (15). For any update probability $\alpha \in (0,1]$ and link failure probability $\delta \in [0,1)$, the PageRank value x^* is obtained through the time average y in (11) as $E\big[\|y(k) - x^*\|^2\big] \to 0$, $k \to \infty$.

The proof follows along similar lines as that for Theorem 3.1 in [17]. Specifically, one way to establish the convergence property is by the general Markov chain results of, e.g., [6]. Another approach is to employ the proof developed in the papers [16], [17] by adapting it to the current update scheme. This proof is found to be useful to study the rate of convergence and to include an update termination feature. Under this termination feature, each page is allowed to stop its update when an approximate value is obtained; this is important because computation as well as communication loads can be reduced.

We have remarks on the asymptotic rate of convergence for the average state $E[x(k)]$. Similarly to the discussion in Section II, the convergence rate is dominated by the second largest eigenvalue $\lambda_2(\overline{M})$ in magnitude. By (6) and (ii) in Lemma 4.2, this eigenvalue can be bounded as

$$\begin{aligned} |\lambda_2(\overline{M})| &= \frac{\hat{m}}{m}|\lambda_2(M)| + 1 - \frac{\hat{m}}{m} \\ &\leq \frac{1-m}{1-m\big[\delta + (1-\delta)(1-\alpha)^2\big]}. \end{aligned}$$

It is clear that this bound is a monotonically decreasing function of α and a monotonically increasing function of δ. That is, higher probability α in updates and/or smaller δ results in faster average convergence. Faster convergence is, nevertheless, realized by additional computation and communication, which are affected by both α and δ.

V. NUMERICAL EXAMPLE

In this section, we present a numerical example to verify the efficacy of the results.

We generated a web with 1,000 pages ($n = 1,000$), where the links among the pages were randomly determined. The first ten pages are designed to have high PageRank values and are linked from over 90% of the pages. For other pages, the numbers of links are between 2 and 333. The parameter m was taken as $m = 0.15$.

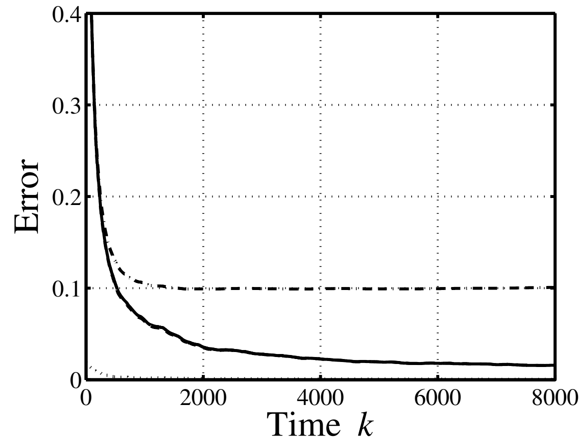

Fig. 1. The error $e(k)$ in the PageRank: $\|e(k)\|_1$ for the original scheme without link failures (dashed line), the original scheme (dash-dot line), and for the proposed scheme (solid line); $\|e(k)\|_\infty$ for the proposed scheme (dotted line).

Simulations were carried out using three algorithms: The first one is the original distributed scheme in Section III, which is run without any link failures and is for reference. The second one is the same original scheme, but failing links are present, that is, when some values from linked pages are not available, they are considered to be zero. The last one is the proposed update scheme with link failures in Section IV. For all three cases, the probability of update for the pages was taken as $\alpha = 0.01$ throughout the simulation.

First, we executed the three algorithms with the link failure probability $\delta = 0.02$. Sample paths of the state x were computed from time 0 to 8,000. The initial state $x(0)$ was taken the same for all algorithms and was randomly chosen as a probability vector. We computed the error $e(k) := y(k) - x^*$ in the PageRank value estimate. In Fig. 1, we show the ℓ_1 norm of $e(k)$. The original scheme without link failures (dashed line) and the proposed scheme (solid line) have comparable performance; the difference is minor and actually is not visible in the plot. In contrast, for the original scheme with failing links, the error almost stops decreasing and stays at a relatively high level of 0.1. This is interesting because the probability δ of link failures is quite small, but has a significant effect. One reason for this is that the original link matrices are in effect no longer stochastic. As a result, the final value $y(k)$ at $k = 8,000$ for this scheme is not a probability vector and, in fact, we obtained $\sum_i y_i(k) = 0.900$. We also plotted in Fig. 1 the ℓ_∞ norm of the error $e(k)$ for the proposed case. This corresponds to the maximum individual error. We observe that it rapidly decreases.

In Fig. 2, the values computed for the first twenty pages are plotted. The true PageRank values are given by $\times$, and the final values of $y_i(k)$ at $k = 8,000$ for the original scheme with link failures are marked as $\triangle$ and and those for the proposed scheme are shown by $\bigcirc$. Overall, the errors for the proposed scheme are fairly small.

Finally, we examined the case where the probability of link failures is $\delta = 0.1$. We generated sample paths similarly

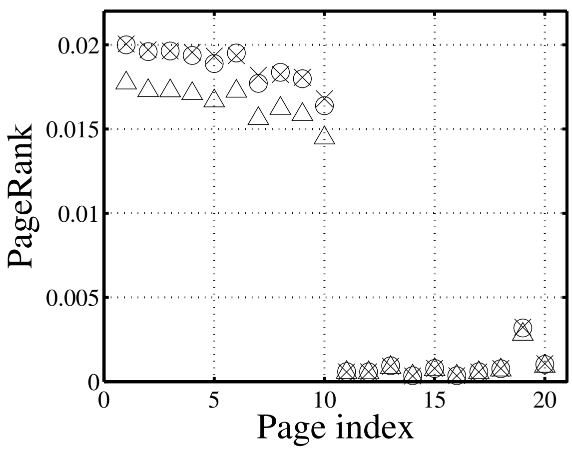

Fig. 2. The PageRank values x_i^* (marked as $\times$), y_i from the original scheme with link failures (marked as $\triangle$), and y_i from the proposed scheme (marked as $\bigcirc$) for $i = 1, \ldots, 20$, at $k = 8,000$.

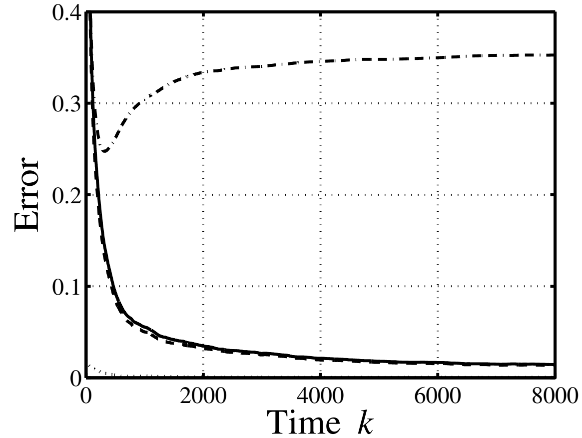

Fig. 3. The error $e(k)$ in the PageRank: $\|e(k)\|_1$ for the original scheme without failing links (dashed line), the original scheme (dash-dot line), and for the proposed scheme (solid line); $\|e(k)\|_\infty$ for the proposed scheme (dotted line).

to the first case and, in Fig. 3, the errors in the PageRank estimation are given. Also, in this case, we obtained good performance for the proposed scheme (solid line) as the error is only slightly larger than the original scheme without failing links (dashed line). When failure is present, the original scheme does not converge, and in fact, the error grows (dash-dot line).

VI. Conclusion

In this paper, we studied extensions of the distributed randomized approach for the PageRank computation proposed in [16], [17]. We considered the effect of link failures under which not all the links are used for communication in the update. This scheme is, in particular, useful to model failures in the network as well as to reduce the communication/computation load for the pages. In future research, we will address issues related to aggregation of webpages for PageRank computation [19].

References

[1] 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.

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

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

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

[5] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44:671–684, 2008.

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

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

[8] N. Elia. Remote stabilization over fading channels. *Systems & Control Letters*, 54:238–249, 2005.

[9] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM J. Contr. Optim.*, to appear, 2008.

[10] V. Gupta, N. C. Martins, and J. Baras. Observing a linear process over analog erasure channels using multiple sensors: Necessary and sufficient conditions for mean-square stability. In *Proc. 46th IEEE Conf. on Decision and Control*, pages 659–664, 2007.

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

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

[13] O.Ç. Imer, S. Yüksel, and T. Başar. Optimal control of LTI systems over unreliable communication links. *Automatica*, 42:1429–1439, 2006.

[14] H. Ishii. H^∞ control with limited communication and message losses. *Systems & Control Letters*, 57:322–331, 2008.

[15] H. Ishii and S. Hara. A subband coding approach to control under limited data rates and message losses. *Automatica*, 44:1141–1148, 2008.

[16] H. Ishii and R. Tempo. A distributed randomized approach for the PageRank computation: Part 1. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 3523–3528, 2008.

[17] H. Ishii and R. Tempo. A distributed randomized approach for the PageRank computation: Part 2. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 3529–3534, 2008.

[18] H. Ishii and R. Tempo. Computing the PageRank variation for fragile web data. *SICE J. Control, Measurement, and System Integration*, 2:1–9, 2009.

[19] H. Ishii, R. Tempo, E.W. Bai, and F. Dabbene. Distributed randomized PageRank computation based on web aggregation. Submitted, 2009.

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

[21] G. Kollias, E. Gallopoulos, and D. B. Szyld. Asynchronous iterative computations with Web information retrieval structures: The PageRank case. In G. R. Joubert *et al.*, editor, *Parallel Computing: Current and Future Issues of High-End Computing*, volume 33 of *NIC Series*, pages 309–316. John von Neumann-Institut for Computing, Julich, Germany, 2006.

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

[23] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control*, 49:1453–1464, 2004.

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