



IEIIT-CNR

Uncertainty and Randomization

The PageRank Computation in Google

Roberto Tempo

IEIIT-CNR

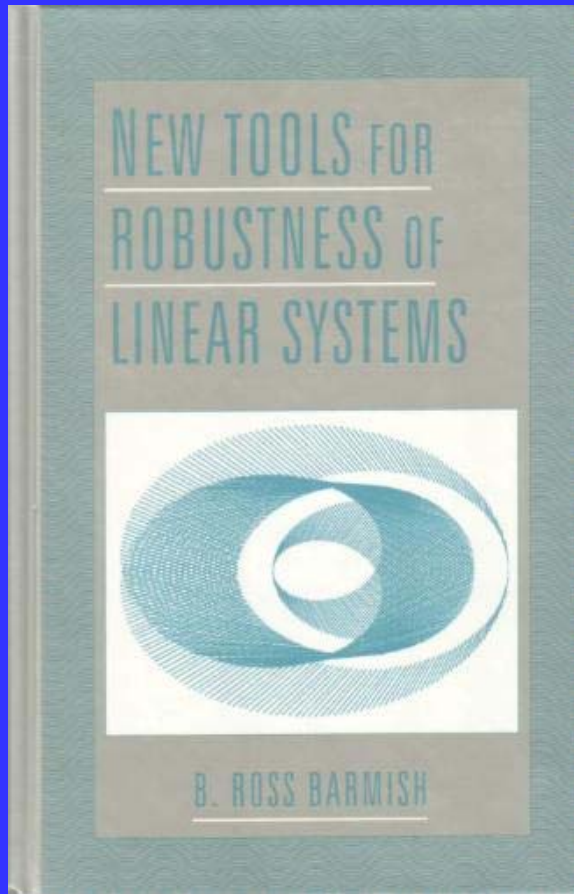
Politecnico di Torino

tempo@polito.it



IEIIT-CNR

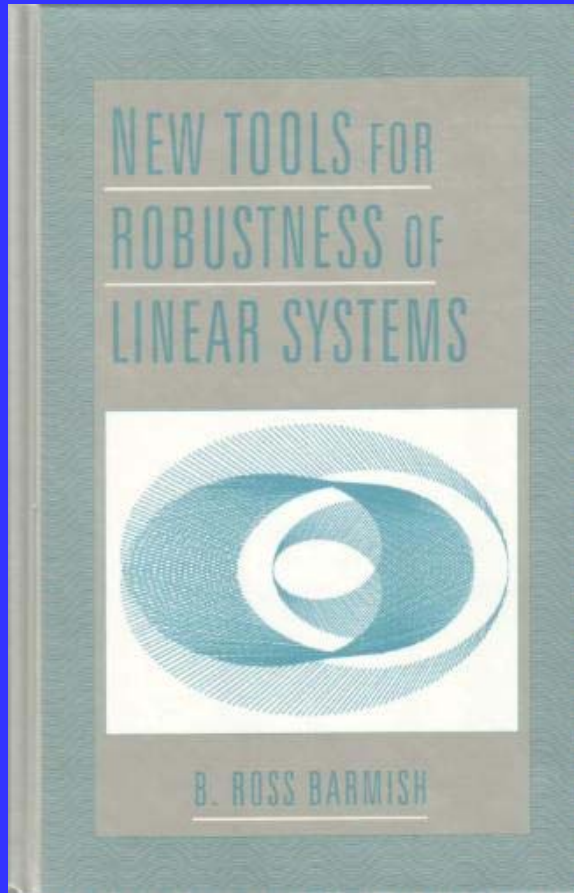
1993: Robustness of Linear Systems





IEIIT-CNR

1993: Robustness of Linear Systems



July 20, 1993 (Sydney)
Roberto: Friendship
and collaboration — scaling
ever higher peaks.
Bob



- ... still working on uncertain systems for modern applications (using different methodologies)
- On the theory side, randomization is now used in the control community as a powerful tool for “solving” problems otherwise intractable



IEIIT-CNR

16 Years Later...



- We are also dealing with very different applications...



- We are also dealing with very different applications...
- The PageRank Problem in Google
- Random Surfer Model and Teleportation Matrix
- A Distributed Randomized Algorithm for PageRank
- Uncertain Systems, Control and Randomization



IEIIT-CNR

The PageRank Problem in Google

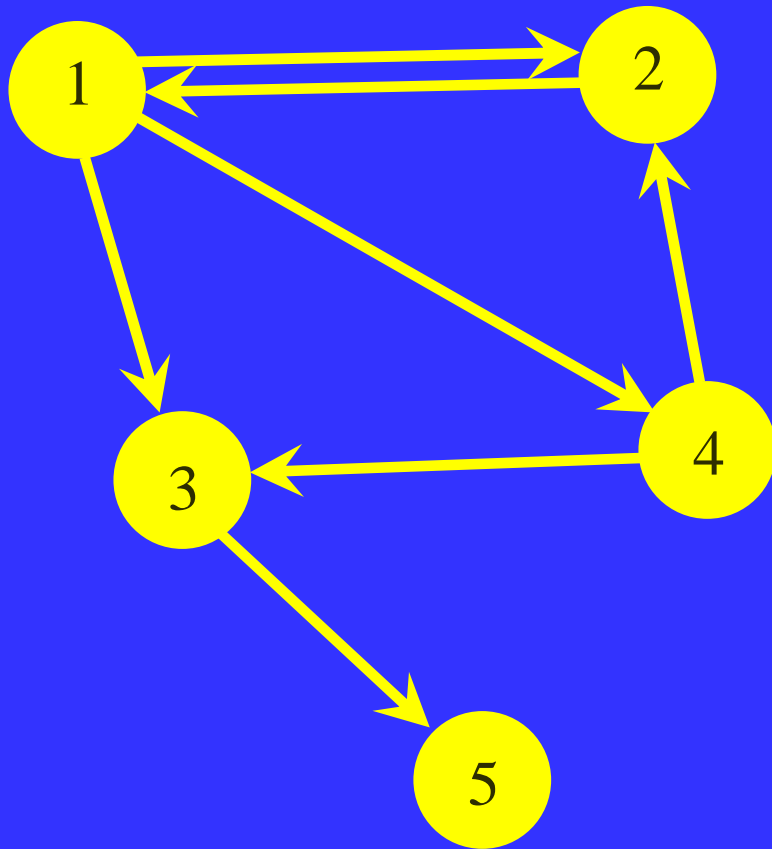


Random Surfer Model

- Web surfer moves along randomly following the hyperlink structure
- When arriving at a page with several outgoing links, one is chosen at random, then the random surfer hyperlinks to the new one, and so on...
- The time the random surfer spends on a page is a measure of the importance of the page
- If important pages point to your page, then your page becomes important. Need to rank the pages for facilitating the web search



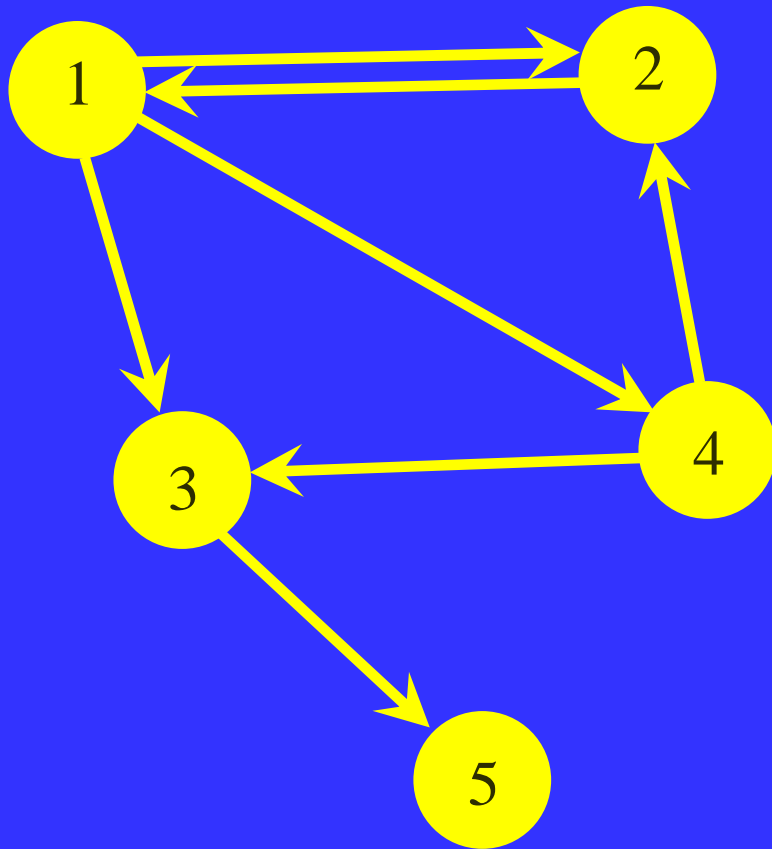
Graph Representation



- Directed graph with nodes (pages) and links representing the web
- Graph is not necessarily strongly connected
- Graph is constructed using crawlers and spiders which move continuously along the web



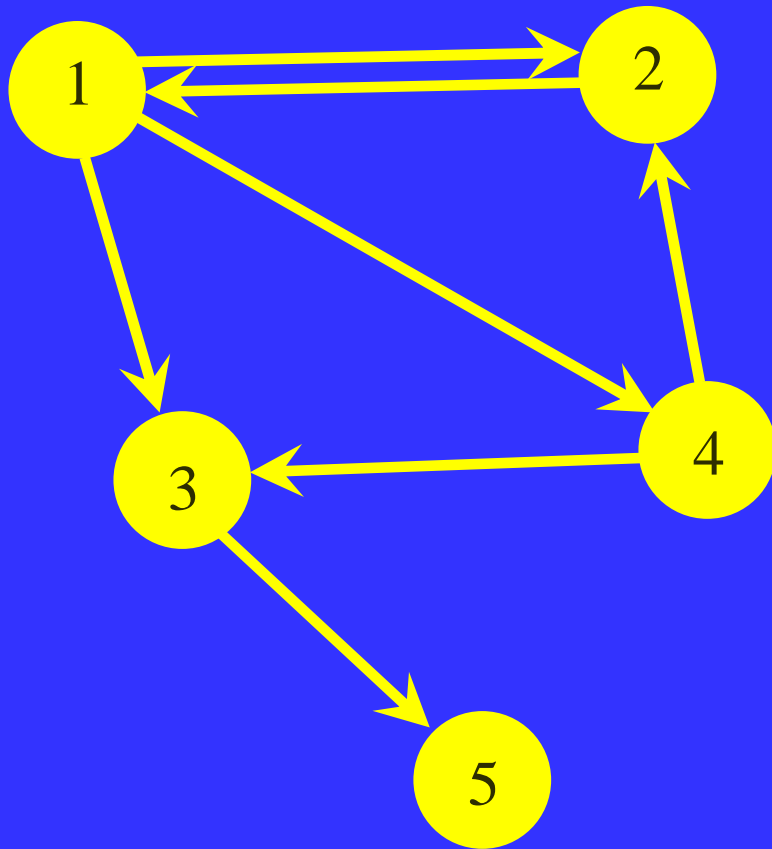
Hyperlink Matrix



- For each node we count the number of outgoing links and normalize them to 1
- Hyperlink matrix is a nonnegative (column) substochastic matrix



Hyperlink Matrix



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



PageRank: Bringing Order to the Web^[1,2]

- Need to rank pages in order of importance
- The PageRank x^* is defined as

$$x^* = Ax^* \quad \text{where} \quad x^* \in [0,1]^n \quad \text{and} \quad \sum_i x_i^* = 1$$

- x^* is a nonnegative unit eigenvector corresponding to the eigenvalue 1 for the hyperlink matrix A
- The question is when x^* exists and it is unique

[1] S. Brin, L. Page (1998)

[2] S. Brin, L. Page, R. Motwani, T. Winograd (1999)

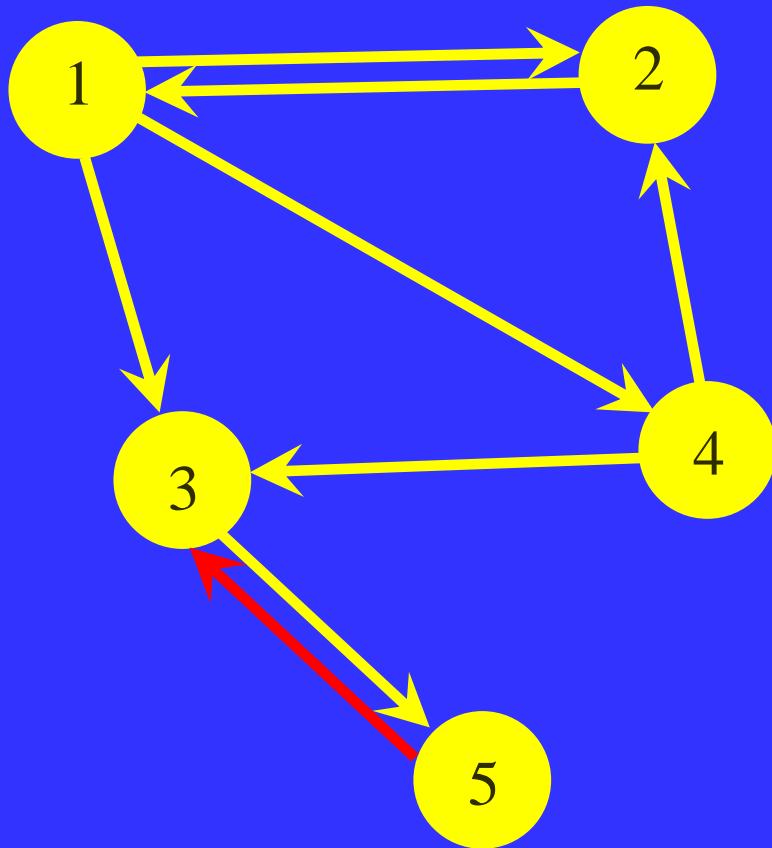


Issue of Dangling Nodes

- First issue: We have *dangling nodes*
- Random surfer gets “stuck” when visiting a pdf file
- In this case the “back button” of the browser is used
- Mathematically, the hyperlink matrix is nonnegative and (column) substochastic
- Easy fix: Add artificial links to make the matrix stochastic



- Page 5 is a dangling node
- We add an outgoing link



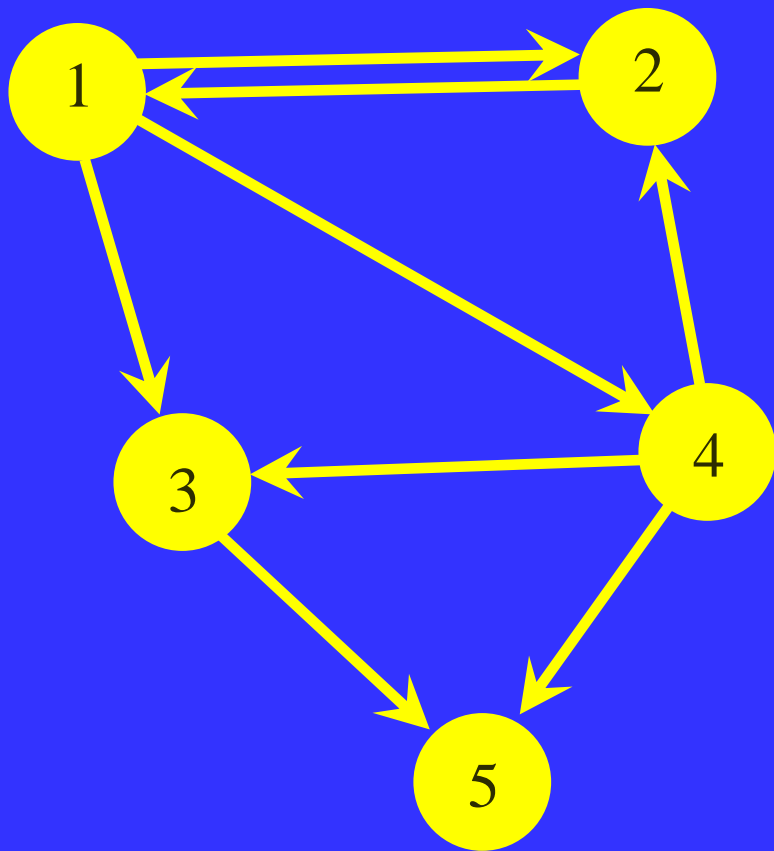
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/2 & 1 \\ 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$



But in General the Fix is not so Easy...



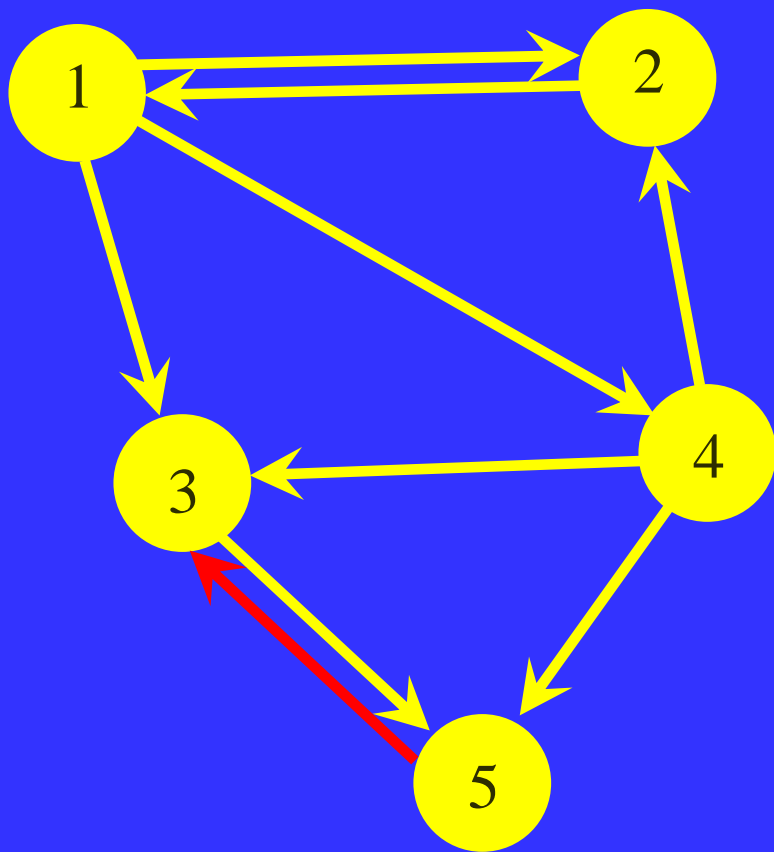
- Page 5 has two incoming links





But in General the Fix is not so Easy...

- We add an outgoing link from 5 to 3...

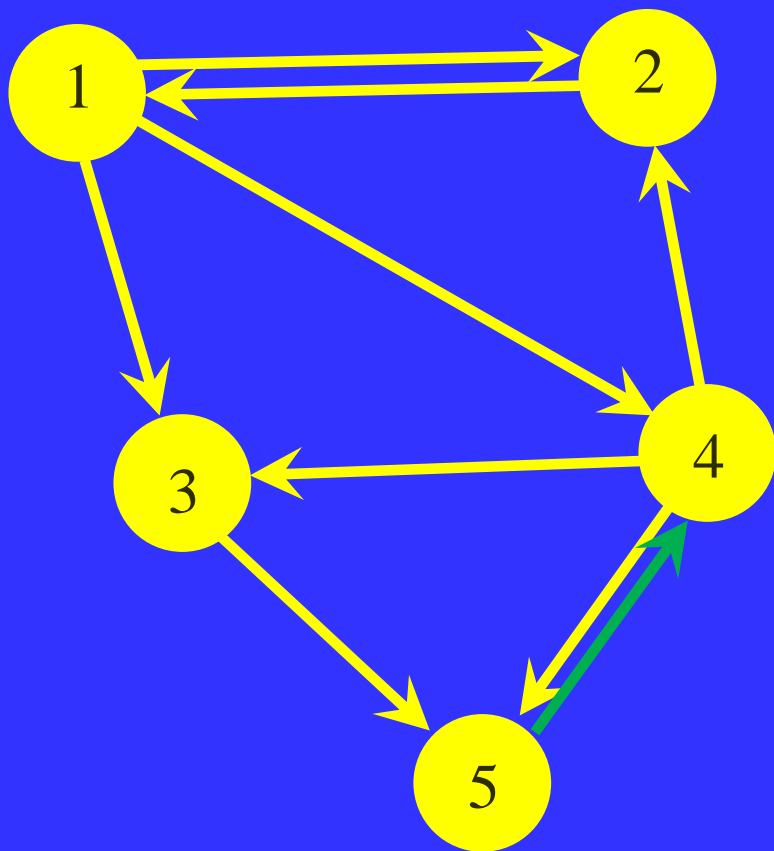


$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1 \\ 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1/3 & 0 \end{bmatrix}$$



But in General the Fix is not so Easy...

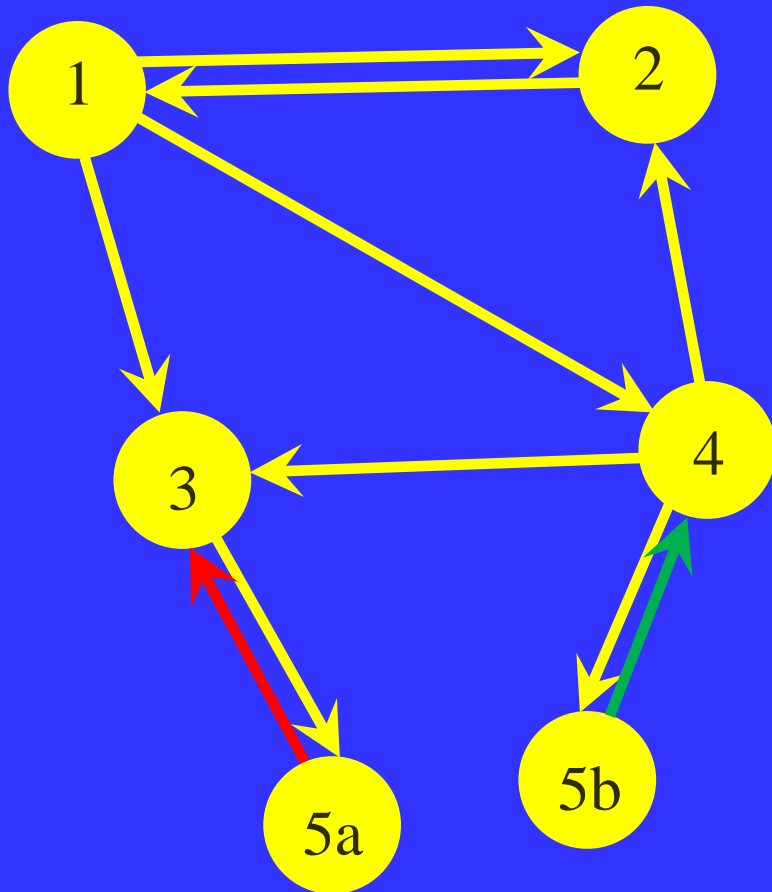
- ... or we add an outgoing link from 5 to 4?



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 0 \\ 1/3 & 0 & 0 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1/3 & 0 \end{bmatrix}$$



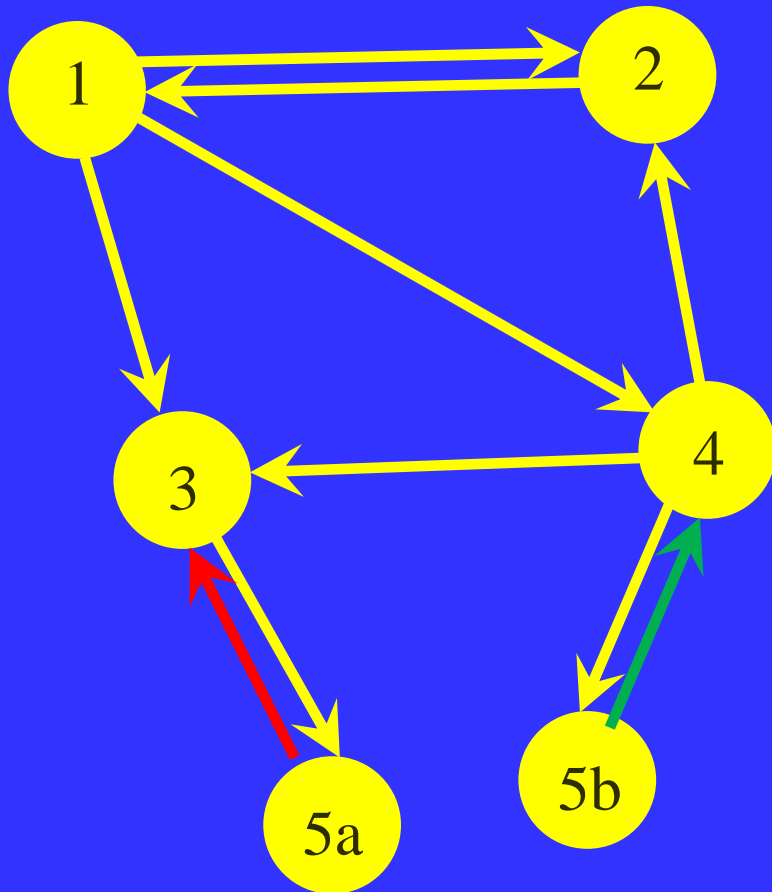
Modified Hyperlink Matrix



- A solution may be to break page 5 into two pages 5a and 5b
- This artificially changes the number of pages (not only the number of links) and the topology of the network



Modified Hyperlink Matrix



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 0 \end{bmatrix}$$



Assumption: No Dangling Nodes

- We assume that there are no dangling nodes
- This implies that A is a nonnegative stochastic matrix (instead of substochastic) having at least one eigenvalue equal to one
- This eigenvalue is not necessarily unique



Teleportation Matrix

- Second issue: The random surfer may get bored after a while, and decides to “jump” to another page not directly connected to that currently visited
- Instead of A we consider a matrix M defined as

$$M = (1 - m) A + m/n S \quad m \in (0,1)$$

where S is a matrix with all entries equal to 1 and n is the number of pages

- The value $m = 0.15$ is proposed and used at Google^[1]

[1] S. Brin, L. Page (1998)



Matrix M and Perron-Frobenius Theorem

- M is positive stochastic (convex combination of two stochastic matrices and $m \in (0,1)$)
- M is irreducible and the corresponding graph is strongly connected (every page is directly connected to every page)
- M is primitive because M^k is positive for some k ($k = 1$)
- Perron-Frobenius Theorem: For a positive stochastic matrix M there exists a unique positive eigenvector for the eigenvalue 1



IEIIT-CNR

PageRank Computation



PageRank Computation

- PageRank is computed with the power method

$$x(k+1) = M x(k)$$

- Convergence of this recursion is guaranteed by Perron-Frobenius Theorem for any initial condition $x(0)$ because M is a positive stochastic matrix

$$x(k) \rightarrow x^* \quad \text{for } k \rightarrow \infty$$

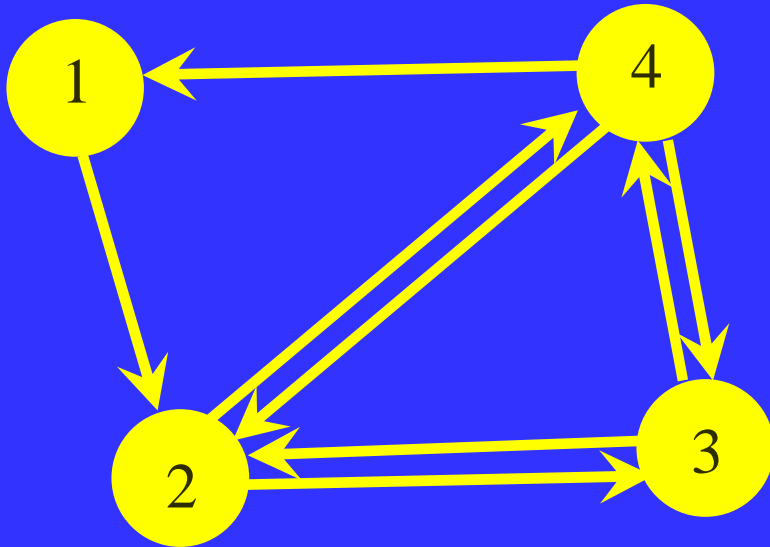
provided that $\sum_i x_i(0) = 1$

- **Remark:** PageRank computation can be interpreted as finding the stationary point of a Markov Chain



IEIIT-CNR

PageRank Computation with Power Method



$$A = \begin{bmatrix} 0 & 0 & 0 & 1/3 \\ 1 & 0 & 1/2 & 1/3 \\ 0 & 1/2 & 0 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix} \quad m=0.15$$

$$M = \begin{bmatrix} 0.038 & 0.037 & 0.037 & 0.321 \\ 0.887 & 0.037 & 0.462 & 0.321 \\ 0.037 & 0.462 & 0.037 & 0.321 \\ 0.037 & 0.462 & 0.462 & 0.037 \end{bmatrix}$$

$$x^* = [0.12 \quad 0.33 \quad 0.26 \quad 0.29]^T$$



IEIIT-CNR

Size of the Web



- The size of M is 8 billion!
- The PageRank computation requires 50-100 iterations
- This takes about a week and it is performed centrally at Google once a month
- More and more computing power is needed



IEIIT-CNR

Columbia River, The Dalles, Oregon





IEIIT-CNR

Randomized Decentralized Approach



IEIIT-CNR

Randomized Decentralized Approach

- Main idea: Develop a decentralized approach for computing PageRank (instead of a centralized approach which involves the entire web)

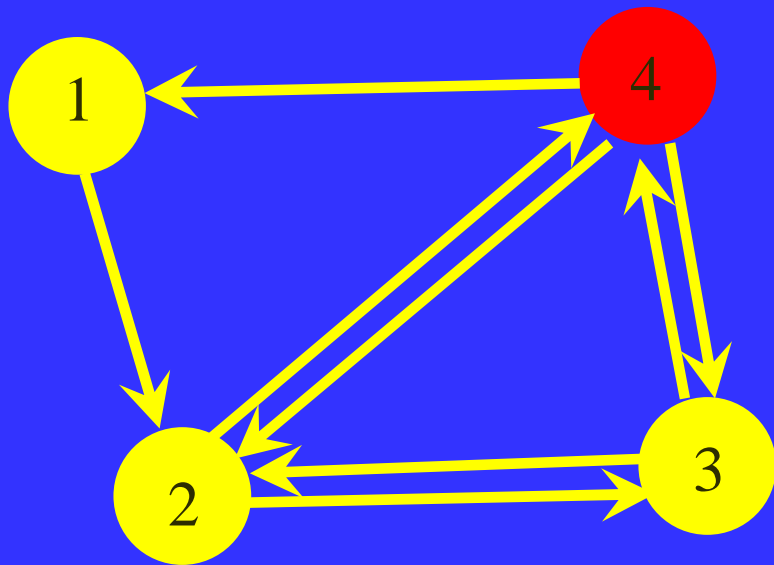


Randomized Decentralized Approach

- Main idea: Develop a decentralized approach for computing PageRank (instead of a centralized approach which involves the entire web)
- Approach is randomization-based (Las Vegas type)



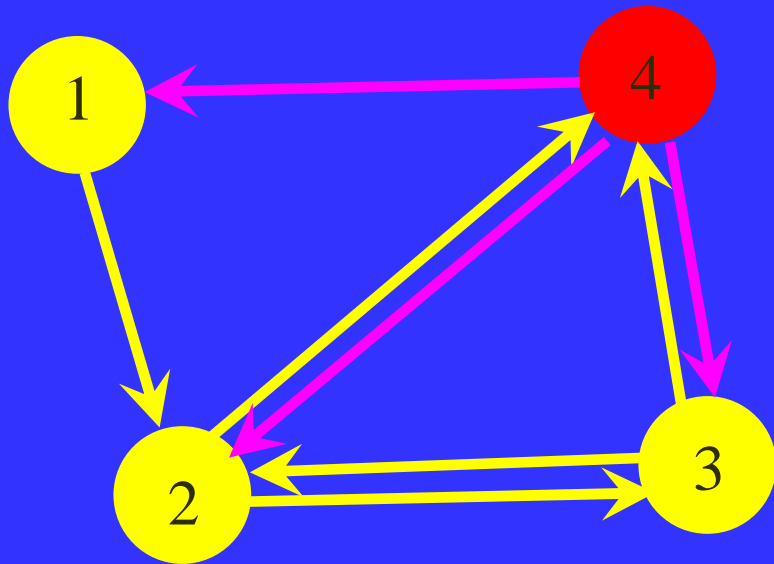
Basic Communication Protocol



Basic communication protocol:
at time k the randomly selected
page i initiates the PageRank
update as follows:



Basic Communication Protocol

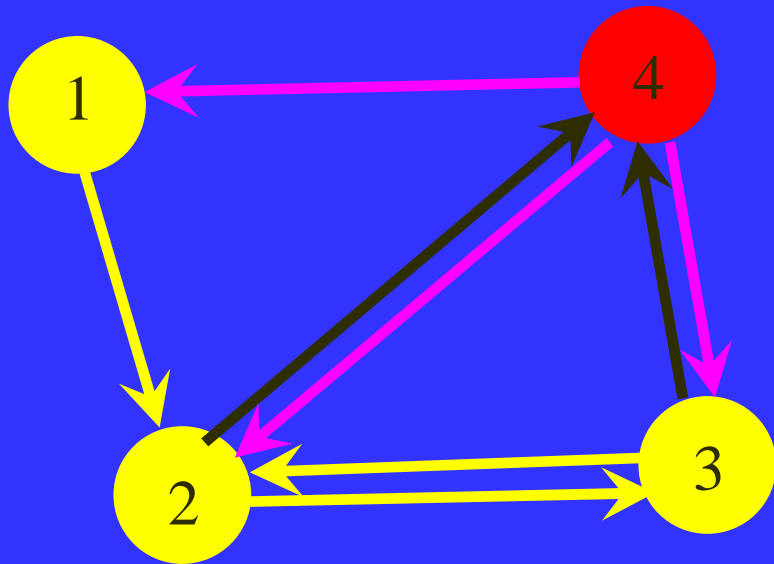


Basic communication protocol:
at time k the randomly selected page i initiates the PageRank update as follows:

1. by sending the value of page i to the outgoing pages that are linked to i



Basic Communication Protocol



Basic communication protocol:
at time k the randomly selected page i initiates the PageRank update as follows:

1. by sending the value of page i to the outgoing pages that are linked to i
2. by requesting their values from the incoming pages that are linked to page i



Las Vegas Randomized Approach

- The pages taking action are determined via a random process $\theta(k)$
- At time k page i initiates PageRank update with uniform probability

$$\text{Prob}\{\theta(k)=i\} = 1/n$$



Distributed Randomized Update Scheme

- We consider the randomized update scheme

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

where $A_{\theta(k)}$ are the distributed link matrices (example next)

- Consider the time average

$$y(k) = 1/(k+1) \sum_i x(i)$$



Distributed Link Matrices - 1

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

$$A_4 = \begin{bmatrix} & & & 1/3 \\ & & & 1/3 \\ & & & 1/3 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$



Distributed Link Matrices - 2

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

$$A_4 = \begin{bmatrix} 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$



Distributed Link Matrices - 3

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

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 1/3 \\ 0 & 1/2 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$



Distributed Link Matrices - 4

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

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/3 \\ 0 & 0 & 1/2 & 2/3 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 1/3 \\ 0 & 1/2 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$



Distributed Link Matrices - 5

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

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 1/3 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2/3 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1/2 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 0 & 2/3 \end{bmatrix}$$



Modified Distributed Update Scheme

- Recall that we need to work with positive stochastic matrices
- We consider the modified distributed update scheme

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

where $M_{\theta(k)}$ are the modified distributed link matrices computed as

$$M_i = (1-r) A_i + r/n S \quad i = 1, 2, \dots, n$$

and $r \in (0,1)$ is a design parameter



- Theorem: Take $r = 2m/(n - mn + 2m) \in (0,1)$
- Using the modified distributed update scheme the PageRank is obtained through the time average y

$$E[\|y(k) - x^*\|^2] \rightarrow 0 \text{ for } k \rightarrow \infty$$

provided that $\sum_i x_i(0) = 1$

- Proof: Based on the theory of ergodic matrices
- Remark: The algorithm is a LVRA

[1] H. Ishii, R. Tempo (2008)



- The average $y(k)$ can be computed recursively in terms of $y(k-1)$

- Sparsity of the matrix A_i can be preserved because

$$x(k+1) = M_i x(k) = (1-r) A_i x(k) + r/n \mathbf{1}$$

where $\mathbf{1}$ is a vector with all entries equal to one

- Recursion $x(k+1) = (1-r) A_1 x(k)$ can be carried on as

$$x_3(k+1) = (1-r) x_3(k)$$

$$z(k+1) = (1-r) B z(k) \quad z = [x_1, x_2, x_4]^T$$

- Convergence rate is $1/k$



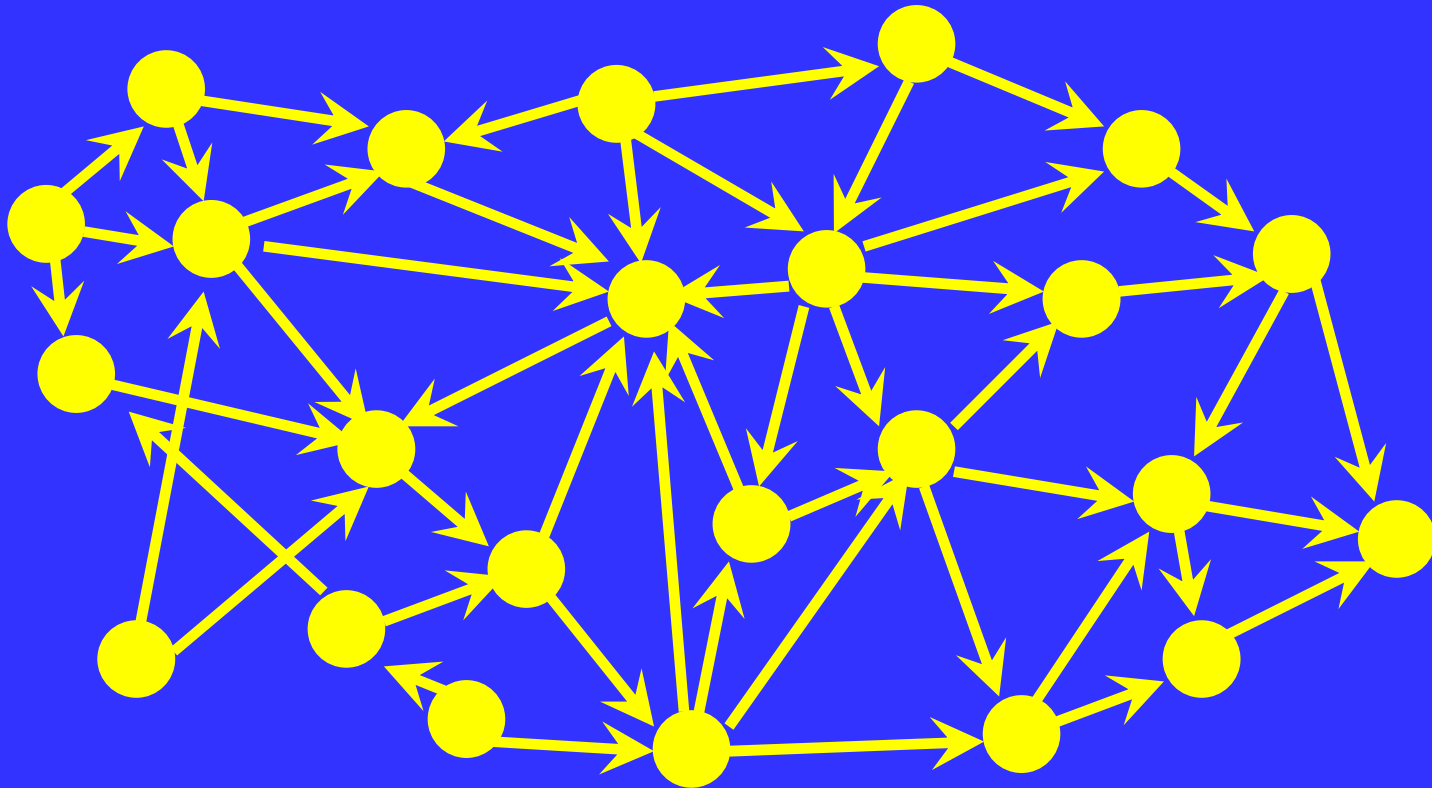
More Deeply into Distributed Randomized Schemes - 1

- Different update schemes based only on outgoing links (not incoming): similar convergence results
- Stopping criteria to compute approximately PageRank
- Need to improve convergence because power method is exponential ($|\lambda_2| \leq 1-m$)



More Deeply into Distributed Randomized Schemes - 2

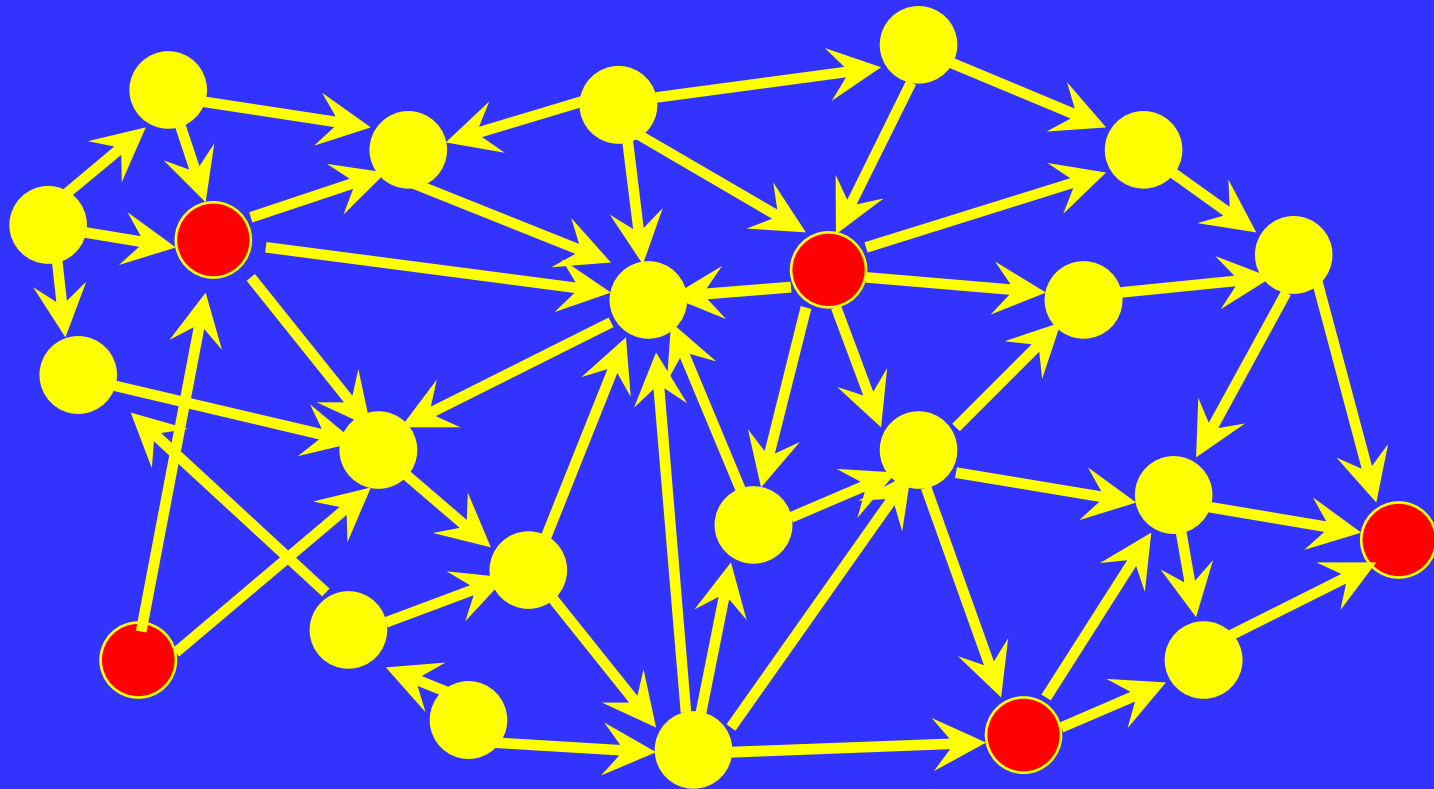
- Simultaneous random update of multiple webpages





More Deeply into Distributed Randomized Schemes - 2

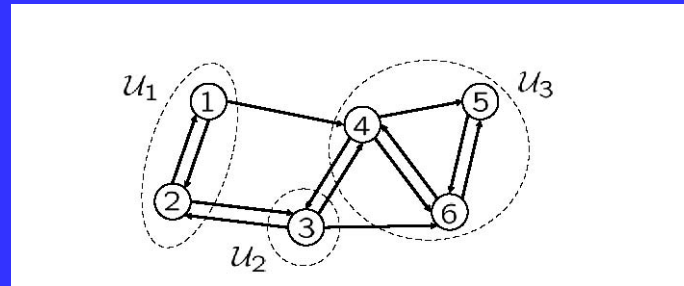
- Simultaneous random update of multiple webpages



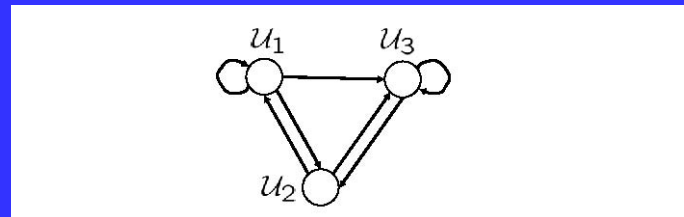


More Deeply into Distributed Randomized Schemes - 3

■ Webpage aggregation and clustering



original web



aggregated web



IEIIT-CNR

Uncertain Systems, Control, Randomization and Google



PageRank and Consensus

- For consensus problems (stochastic version) we consider a graph representing a network of agents x_i


Consensus	PageRank
All agent values become equal	Page values converge to constant
Graph is strongly connected	Web is not strongly connected
Convergence w.p.1 for all x_i, x_j $ x_i(k) - x_j(k) \rightarrow 0, k \rightarrow \infty$	MSE convergence for y $E[\ y(k) - x^*\ ^2] \rightarrow 0, k \rightarrow \infty$
Matrices A_i are row stochastic	Matrices A_i are column stochastic



PageRank and Uncertain Systems

- PageRank computation in the presence of uncertain, time-varying and broken links (LP solution)

Page not found - connection failure



Oops! This link appears broken.

Suggestions:

- Go to www.navy.mil
- Search on Google:

[Google Toolbar Help - Why am I seeing this page?](#)

©2009 Google - [Google Home](#)



IEIIT-CNR

Acknowledgment

- Acknowledgment: Research on PageRank is joint work with Hideaki Ishii
- Subsequent work with Er-Wei Bai, Fabrizio Dabbene, Shinji Hara

<http://staff.polito.it/roberto.tempo/>