TARJAN'S ALGORITHM IN COMPUTING PAGERANK

Ivana Pultarová

Department of Mathematics, Faculty of Civil Engineering,
Czech Technical University in Prague

Abstract

As a core problem in computing PageRank a stationary probability distribution vector is solved. We show, that using Tarjan's reordering in connection to iterative aggregation-disaggregation method can speed up the convergence significantly in comparison to standard methods.

1 Introduction

In this work, some new observations in computing PageRank are presented. Tarjan's algorithm in connection to an iterative aggregation-disaggregation method seems to result in a new promissing approach in computing a stationary probability distribution vector of a large-scale stochastic matrix which forms a core problem in PageRank computing.

It is shown that for some sort of stochastic matrices, the iterative aggregation-disaggregation method may yield a sequence of vectors which converges much faster to the exact stationary probability distribution vector than a sequence computed by power method. Such matrices can be obtained by Tarjan's reordering. Because the mentioned iterative method with preprocessing by Tarjan's algorithm can be very costly in comparison to the simple use of power method, we can consider also a threshold adaptation of Tarjan's algorithm.

The mentioned algorithms are compared in this paper when dealing with two real-looking problems representing parts of hyperlink structures of the Web. These are two matrices, Stanford and Stanford-Berkeley matrices [3], the well known examples for testing PageRank algorithms. The experiments are performed in Matlab using its tools for large sparse matrix computation.

The paper is organized as follows. The basic subjects used are shortly described in the next four sections. We introduce a definition of PageRank [1, 8] as a stationary probability distribution vector [12] of a convex combination of a rank one matrix and a matrix reflecting the hyperlink structure of (a part of) the Web. We describe a basic method (power method) for computing this vector and some of its properties. Then the iterative two-level algorithm (iterative aggregation-disaggregation method) [6, 7, 9, 10] and Tarjan's algorithm are briefly introduced. In the end of the paper, computational examples comparing the suggested methods and the power method are presented.

2 What is PageRank

One of the possibilities how to organize the list of Web pages when they are displayed by some Web search engine as a result of some client's query is to organize them according to their PageRank values. PageRank is a vector of positive numbers, each of them corresponds to a particular page and it is defined as a probability that a random Web surfer is visiting this Web page [1, 4, 5, 8]. It means that it is a stationary probability distribution vector [12] of a random process denoted as Markov chain [11]. This process is connected to a stochastic matrix [12] G, an element G_{ij} of which somehow corresponds to probability that a random surfer follows to page i via a single hyperlink from page j. In other words, PageRank is defined as a stationary probability vector of a convex combination of matrix G and some appropriate rank one stochastic matrix Q [8]. This means that PageRank equals to vector $\hat{x}$ such that

$$(\alpha G + (1-\alpha)Q)\hat{x} = \hat{x},$$

where $\alpha \in (0,1)$ is a suitable constant [5, 8]. All of the columns of Q are equal and can also characterize some kind of importance of pages. Let us denote

$$B = \alpha G + (1-\alpha)Q$$

in our further considerations.

3 Computing PageRank

Let us stress that due to irreducibility and primitivity of B there exists a unique stationary probability vector $\hat{x}$ of B. This follows from Perron-Frobenius theorem [12].

Computing PageRank is extremely large-scale problem. It is spoken about several billions of pages considered. So that one has to handle with matrix B of such size. Of course, B is sparse and should be stored and handled in an appropriate way.

The most popular and most simple method for computing stationary probability vector of a stochastic matrix is power method. It generates a sequence of approximations x^k given by formula

$$x^{k+1} = Bx^k$$

for any positive x^0 such that $||x^0|| = 1$. The limit of the sequence $\{x^k\}_{k=0}^{\infty}$ is unique and equals to the Perron-Frobenius eigenvector $\hat{x}$ of B if B is irreducible and primitive [12]. Moreover, the asymptotic convergence factor is equal to the magnitude of the second largest eigenvalue of B, which is α in the case of the PageRank matrix B [4, 9, 10]. It means that the approximation error reduces asymptotically approximately by factor α in each step.

4 Iterative aggregation-disaggregation method

An alternative approach in computing stationary probability distribution vector insists in an iterative two level algorithm for solving

$$Bx = x.$$

It is called the iterative aggregation-disaggregation (IAD) method. The set of events is partitioned into subgroups and some corresponding smaller size problem is solved. Then the obtained solution is prolonged to the original size and corrected by several steps of some basic iteration method, e.g. power method, block Jacobi od block Gauss-Seidel methods [11]. One of the basic questions is the choice of the aggregation groups. In [6] was shown that for some special structure of B one can obtain the exact solution after at most two iterations of the IAD method. In [7] this statement was generalized, still the desired structure property may be undetectable in practical large-scale computing. The only tool for analysing the mentioned structure is Tarjan's algorithm. Basically, Tarjan's algorithm finds a symmetric permutation of a matrix leading to an upper triangular form with irreducible diagonal blocks [2].

Now we introduce the IAD method. Let $G_1, \ldots, G_n$, $n \leq N$, be the *aggregation groups* of events which are numbered with $1, 2, \ldots, N$. All of pairs of the sets G_i, $i = 1, \ldots, n$, are assumed to be disjoint and all of these sets are covering the whole set of events, $\cup_{i=1}^{n} G_i = \{1, 2, \ldots, N\}$. Let us define the *restriction* (*aggregation*) $n \times N$ matrix R, $R_{ij} = 1$ if $j \in G_i$ and $R_{ij} = 0$ otherwise. For any positive x the *prolongation* (*disaggregation*) $N \times n$ matrix $S(x)$ is defined by

$$S(x)_{ij} = \frac{x_i}{\sum_{k \in G_j} x_k}$$

if $i \in G_j$ and $S(x)_{ij} = 0$ otherwise. Let $P(x)$ be a projection matrix given by

$$P(x) = S(x)R.$$

Note that $RS(x) = I$, I is the identity matrix here. Finally let $T = M^{-1}W$ be a matrix given by some splitting of $I - B$, $I - B = M - W$, which is of weak nonnegative type, i.e. $M^{-1} \geq 0$ and $M^{-1}W \geq 0$ [12].

The IAD method consists of several repeating steps. In this part the upper vector index denotes the order of a vector in a sequence, while the upper matrix index is an exponent.

IAD algorithm.

Step 1. An elementwise positive initial approximation x^0, $||x^0|| = 1$, is selected. The value k is set to 0.

Step 2. A positive integer s is chosen and an $n \times n$ *aggregated* matrix

$$RB^sS(x^k)$$

is constructed. The associated problem is solved, i.e. a vector z is found, which fulfilles

$$RB^sS(x^k)z = z,$$

$||z|| = 1$. This step can be called *solution on the coarse level*.

Step 3. A prolonged vector $x^{k+1,1}$ of the original size N is computed by

$$x^{k+1,1} = S(x^k)z.$$

Step 4. The next approximation x^{k+1} is computed by $x^{k+1} = T^t x^{k+1,1}$ for an appropriate positive integer t, where $T = M^{-1}W$. This step can be called *the smoothing* step or *the correction on the fine level*.

Step 5. The test for convergence is evaluated and then the algorithm finishes with the approximate solution x^{k+1} as a result or it continues with *Step 2* and with k increased by 1.

Note that the all computed vectors x^k are positive and $||x^k|| = 1$. For any positive x, the aggregated matrix $RB^sS(x)$ is stochastic and primitive. Computing z in *Step 2* is assumed to be carried out exactly. In the place of the iteration matrix T one can choose any nonnegative matrix with the properties $T\hat{x} = \hat{x}$ and $I - B = M(I - T)$ with some invertible M.

5 Tarjan's reordering

A structure of nonzero elements of a nonnegative matrix G can be connected with a graph, where an edge leading from node j to node i represents positivity of element G_{ij}. Then finding the symmetric permutation resulting in block upper diagonal form with irreducible diagonal blocks corresponds to finding all of the strong components of the graph. This problem can be solved by Tarjan's algorithm [2]. We can also work with a threshold adaptation of this method where a lower limit for considering a number for nonzero is used.

6 Numerical experiments

We provide a numerical comparison of the introduced algorithms. Stationary probability distribution vectors are computed by power method and by IAD algorithm with preprocessing by Tarjan's algorithm and without it. In tests we deal with data frequently used for PageRank computation. The matrices representing parts of the Web are Stanford and Stanford-Berkeley matrices [3]. Let us abbreviate their names to S-matrix and SB-matrix, respectively. To illustrate the properties of the data, we introduce some of their characteristics. The sizes of the matrices are 281 903 $\times$ 281 903 and 683 446 $\times$ 683 446, respectively. The numbers of nonzero

elements are 2 312 669 and 7 588 111, respectively. Thus the densities of nonzero elements are about 0.0029% and 0.0016%, respectively, and the average numbers of nonzero elements per column is 8.2 and 11.1, respectively. The sparsity patterns of them is shown on Figures 1 and 2. On right hand sides of the figures, the structures of some diagonal blocks are projected. Let us notice that the S-matrix does not seem to possess any structure, while the SB-matrix looks like in some way ordered data both in coarse and fine levels.

The results of Tarjan's reordering algorithm are displayed on Figures 3 and 4. On left hand sides, the original 20000×20000 diagonal blocks of S-matrix and SB-matrix are shown and on right hand sides the resulting permuted matrices are presented, both from the point of view of their sparsity structure. Let us observe again the almost regular structure of SB-matrix.

Now we introduce some tests on S-matrix and SB-matrix. In each test, a stationary probability distribution vector of matrix B is computed, where

$$B = 0.85G + 0.15E,$$

E is a matrix of ones divided by N and G is S-matrix in Tests 1 and 2 and it is SB-matrix in Tests 3 and 4, respectively.

Test 1. In the first test we compare the rate of convergence of power method and the IAD method, where the parameters of the IAD agorithm are $t = 1$, $s = 1$ and T corresponds to block Jacobi iteration, i.e. T is the product of the inverse of the block diagonal of $I - B$ and the block nondiagonal part of B. The diagonal blocks correspond to the aggregation groups. Due to the extremely large scale of the problem, we perform the computations only with a part of S-matrix. We adapted a diagonal square submatrix with indexes $20\,001 - 40\,000$, i.e. we set ones to the diagonal positions of empty columns, then we normalized the matrix. There are 100 aggregation groups ($n = 100$), each of the size 200. The errors of approximations of the exact solution measured in 1-norm computed for S-matrix is shown in Figure 5. The red line denotes the errors of power method and the red circles denote the errors of power method after Tarjan's permuting with the threshold 0. Let us see that Tarjan's reordering has no effect on the convergence of power method. The blue line means the errors obtained by the IAD method and the blue circles denote the errors of the IAD method preprocessed by Tarjan's permuting.

Test 2. The same computation as in Test 1 is performed, but the number of smoothing steps is 5, $t = 5$. Resulting errors are displayed on Figure 6. Here the error of power method is displayed after each five steps, in order the comparison to IAD was more realistic. The coloring corresponds to the Test 1.

Test 3. The same computation as in Test 1 is done for SB-matrix, see Figure 7.

Test 4. Finally, in this test five steps of smoothing is done in each IAD iteration for SB-matrix. The other items are identical to Test 2. The resulting errors are shown on Figure 8.

7 Conclusion

We show an efficient iterative method for computing a stationary probability vector of a stochastic matrix suitable for solving large-scale problems. We show that in the case of computing PageRank, some properties of the corresponding stochastic matrix can be exploited for obtaining faster convergence. This is the sparsity of matrix G. The Tarjan's algorithm can be used to reorder the set of events (pages) in a more appropriate way. For such data the IAD method can converge significantly faster than power method and also than unpreprocessed IAD method, see [6, 10] for more detailed analysis.

Acknowledgement. This research was supported by project CEZ MSM 6840770001.

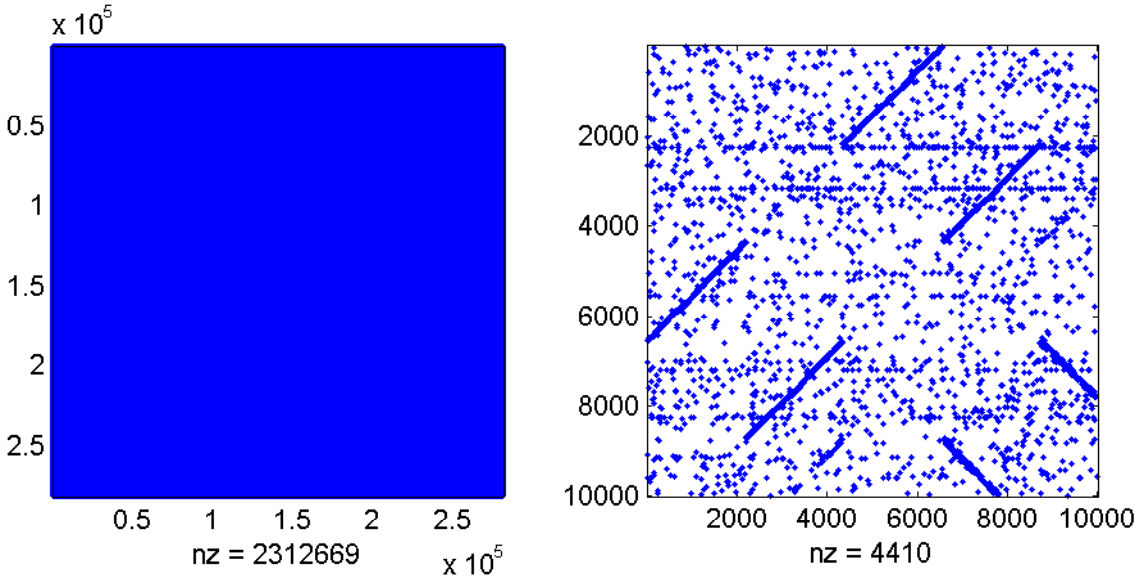

Figure 1: Sparsity structure of S-matrix. The whole matrix (resolution is not fine enough) and a diagonal submatrix with indices $1 - 10\,000$.

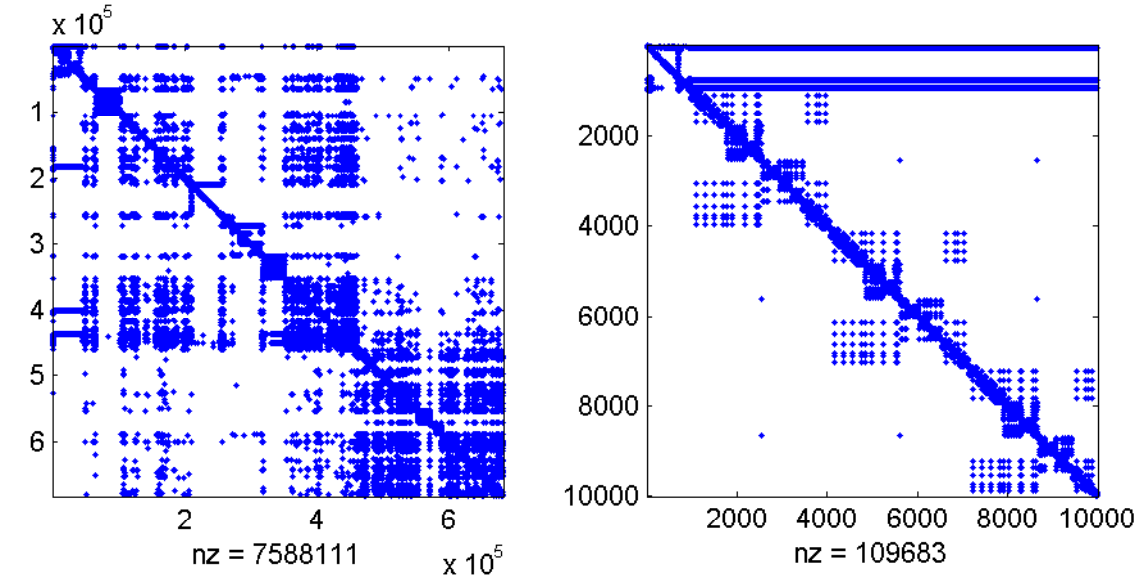

Figure 2: Sparsity structure of SB-matrix. The whole matrix and a diagonal submatrix with indices $1 - 10\,000$.

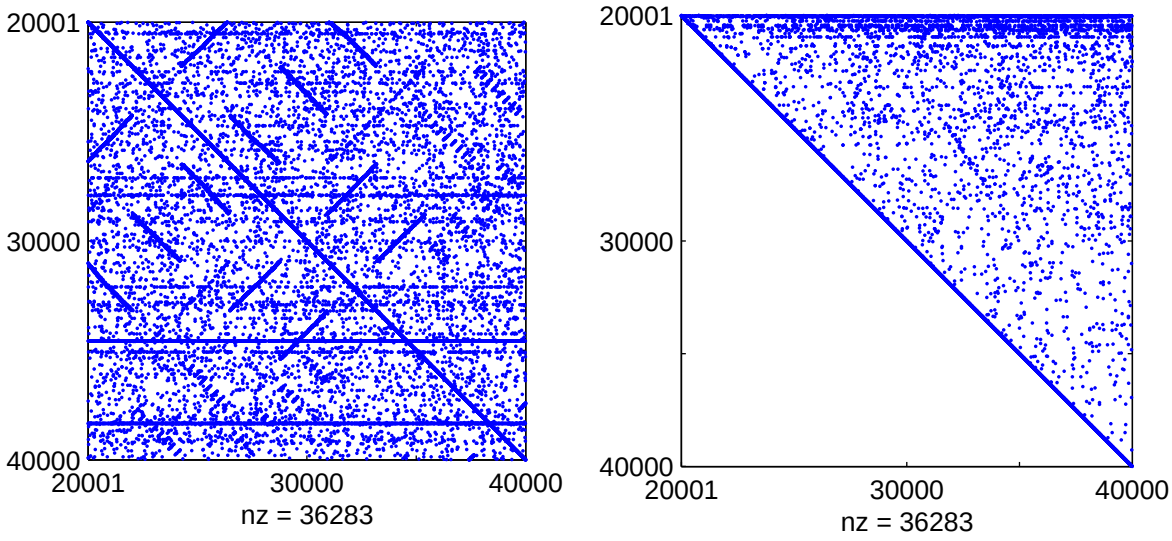

Figure 3: Tarjan's permuting (right) of a diagonal 20 000 × 20 000 submatrix (left) of the S-matrix.

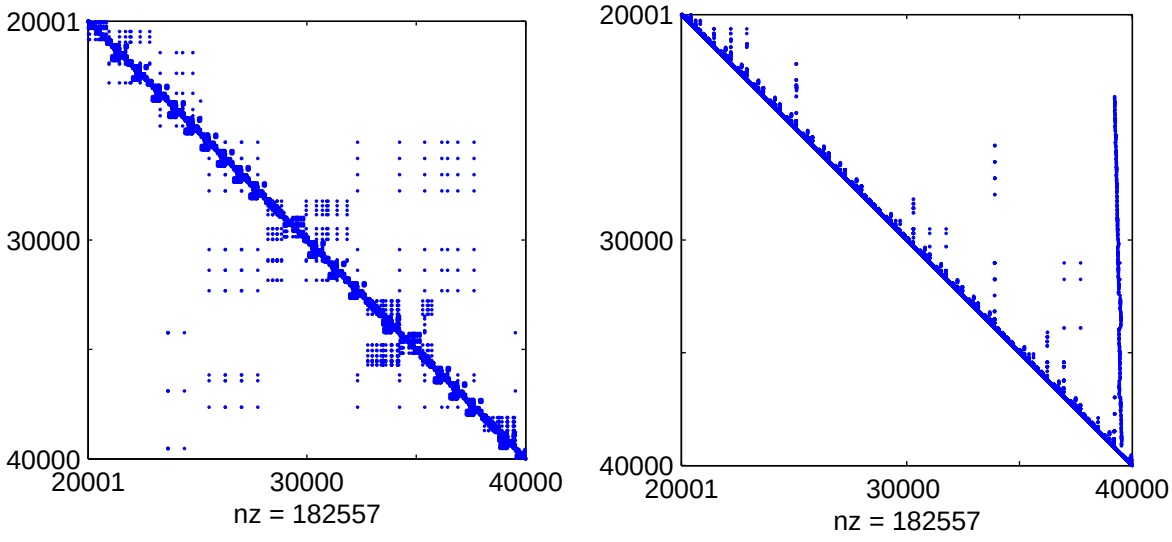

Figure 4: Tarjan's permuting (right) of a diagonal 20 000 × 20 000 submatrix (left) of the SB-matrix.

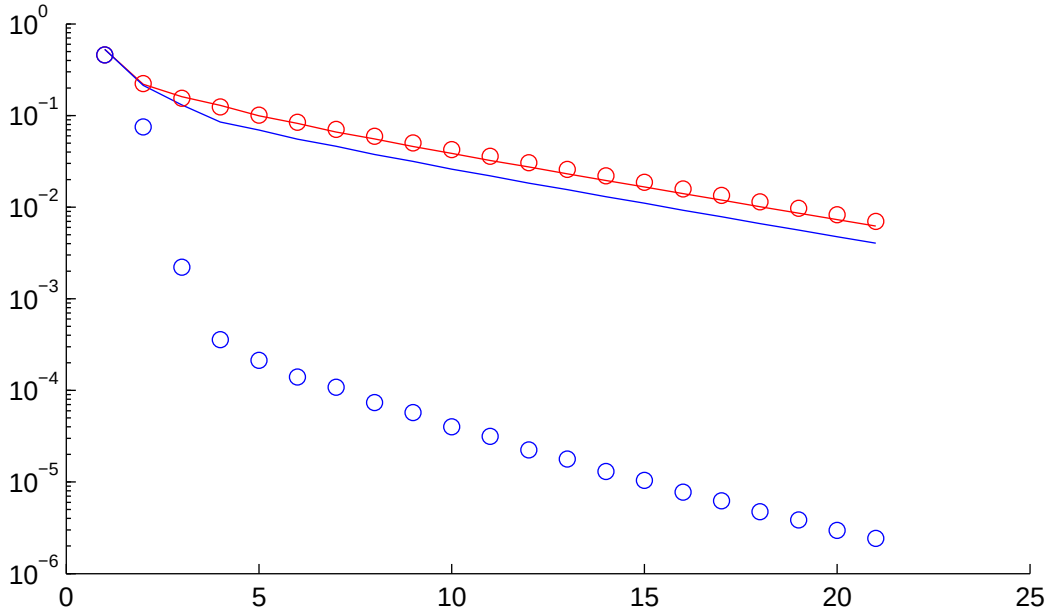

Figure 5: Graphical plot of the errors for Test 1 - 20000×20000 submatrix of S-matrix, $s = t = 1$, $n = 100$. Red line - power method, blue line - IAD method. Red circles - power method after Tarjan's reordering, blue circles - IAD method after Tarjan's reordering.

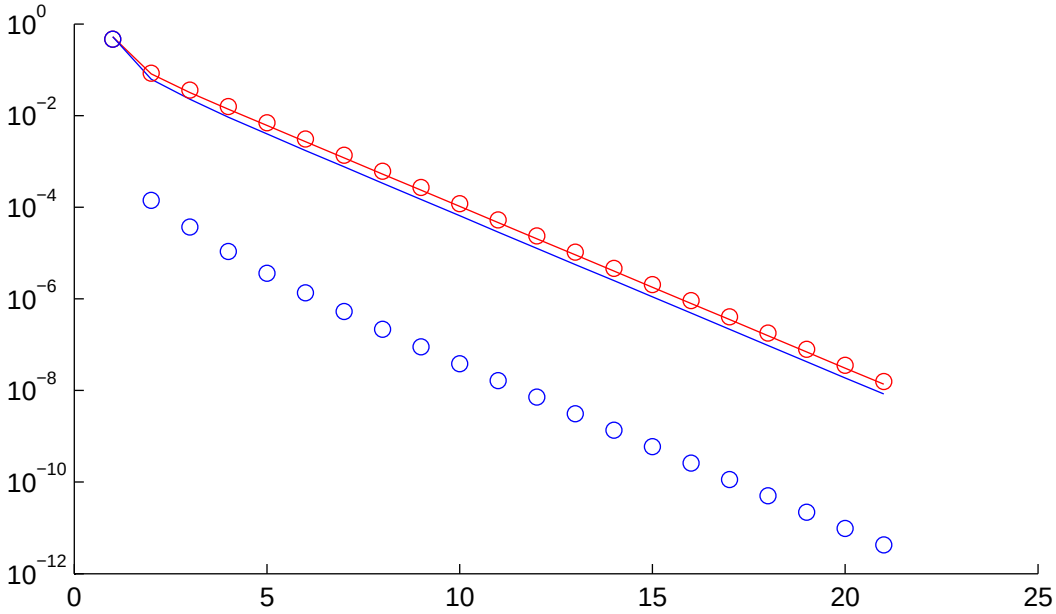

Figure 6: Graphical plot of the errors for Test 2 - 20000×20000 submatrix of S-matrix, $s = 1$, $t = 5$, $n = 100$. Red line - power method, blue line - IAD method. Red circles - power method after Tarjan's reordering, blue circles - IAD method after Tarjan's reordering.

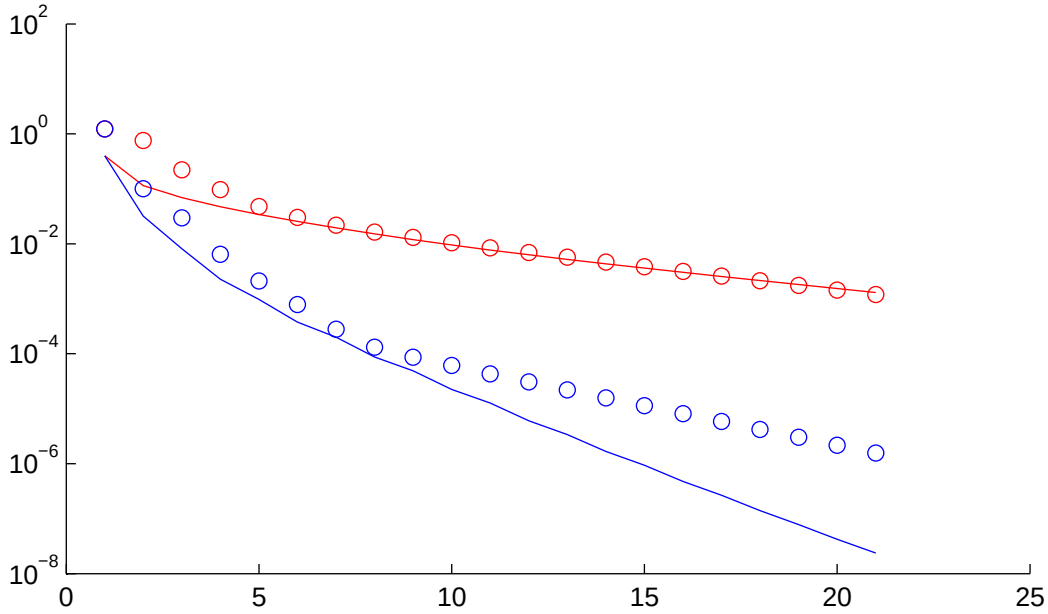

Figure 7: Graphical plot of the errors for Test 3 - 20000 × 20000 submatrix of SB-matrix, $s = t = 1$, $n = 100$. Red line - power method, blue line - IAD method. Red circles - power method after Tarjan's reordering, blue circles - IAD method after Tarjan's reordering.

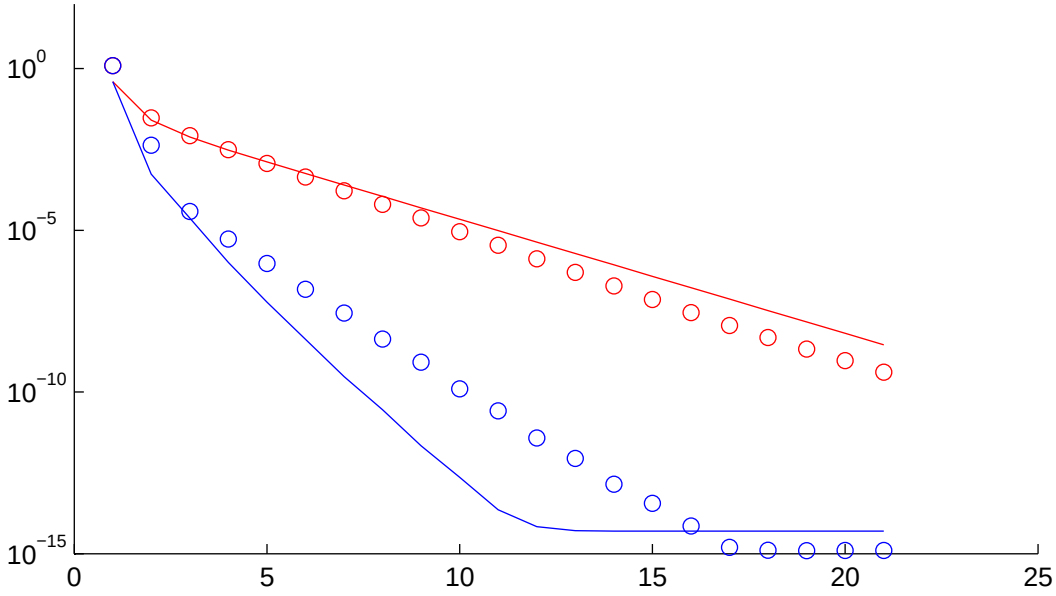

Figure 8: Graphical plot of the errors for Test 4 - 20000 × 20000 submatrix of SB-matrix, $s = 1$, $t = 5$, $n = 100$. Red line - power method, blue line - IAD method. Red circles - power method after Tarjan's reordering, blue circles - IAD method after Tarjan's reordering.

References

[1] S. Brin, L. page, R. Motwami, T. Winograd. *The PageRank citation ranking: Bringing order to the web.* Technical report, Computer Science Department, Stanford University, 1998.

[2] I. S. Duff, J. K. Reid. *An implementation of Tarjan's algorithm for the block triangularization of a matrix.* ACM Transactions on Mathematical Software 4 (1978) 137-147.

[3] S. Kamvar. Data sets of Stanford Web Matrix and Stanford-Berkeley Web Matrix, http://www.stanford.edu/sdkamvar/research.html.

[4] A. N. Langville, C. D. Meyer. *Deeper Inside PageRank.* Internet Mathematics 1 (2005) 335-380.

[5] A. N. Langville, C. D. Meyer. *A Survey of Eigenvector Methods of Web Information Retrieval.* The SIAM Review 47 (2005) 135-161.

[6] I. Marek, P. Mayer. *Convergence theory of some classes of iterative aggregation-disaggregation methods for computing stationary probability vectors of stochastic matrices.* Linear Algebra and Its Applications 363 (2003) 177-200.

[7] I. Marek, I. Pultarová. *A note on local and global convergence analysis of iterative aggregation-disaggregation methods.* To appear in Linear Algebra and its Applications.

[8] C. Moler. *The world's largest matrix computation.* Matlab News & Notes, October 2002.

[9] I. Pultarová. *Google search engine - some computing experience.* Proceedings of Seminar on Numerical Analysis, Modeling and Simulation of Challenging Engineering Problems, Ostrava, 2005.

[10] I. Pultarová. *Iterative aggregation-disaggregation methods in computing Markov chains.* not published.

[11] W. J. Stewart. *Introduction to the numerical solutions of Markov chains.* Princeton University Press, Princeton, 1994.

[12] R. S. Varga. *Matrix iterative analysis.* 2nd ed., Springer Series in Computational Mathematics 27 , Springer-Verlag, New York, 2000.

Address, e-mail and phone. Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, Praha 6, Czech Republic; ivana@mat.fsv.cvut.cz; +420 22435 4408.