INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2013) Vol. 2 (1) 73−84

EVALUATION OF THE PAGERANK ALGORITHM EFFECTIVENESS

KAZIMIERZ WORWA, GUSTAW KONOPACKI

The Faculty of Cybernetics, Military University of Technology

In this paper the challenges in building good search engines are discussed. Many of the search engines use well-known information retrieval algorithms and techniques. They use Web crawlers to maintain their index databases amortizing the cost of crawling and indexing over the millions of queries received by them. Web crawlers are programs that exploit the graph structure of the Web to move from page to page. Paper analyses the PageRank algorithm one of these Web crawlers. The results of the impact of the PageRank parameter value on the effectiveness of determining the so-called PageRank vector are considered in the paper. Investigations are illustrated by means of the results of a some simulation experiments to analyze the PageRank algorithm efficiency for different density graph (representing analyzed part of www) coefficient values.

Keywords: Search engine, Crawling, PageRank algorithm

1. Introduction

One of the most popular services offered by modern Internet is www. Access to the Web resources is implemented mostly through search engines , whose functionality is growing. Users of the search engine form queries resulting in a list of websites containing the following keywords. Most of the search engines uses familiar, traditional algorithms and information retrieval techniques developed for searching a relatively small and thematically coherent collection, such as catalogs of books in the library. These methods are not effective enough for the needs of

Web search, which is a huge, much less consistent, very often changing its content and structure, and is spread over geographically distributed computers. For the purpose of searching the Internet is therefore required to improve the traditional information retrieval techniques or develop new ones. The research carried out in order to estimate the size of modern Internet shows that it consists of over one billion pages. Given that the average web page size is approximately 5-10 kilobytes size of the Internet can be estimated at tens of terabytes. The Internet is characterized by a very high dynamics of change in its size and structure. The research conducted by Lawrence and Giles [10] shows that the size of the Web has doubled in the last two years. Large is the dynamics of Internet content . In addition to the newly created pages, existing pages are constantly updated. Research carried out by Cho and Garcia -Molina [4] shows that about 23% of all the pages available on the Web is updated daily. Knowledge of the structure and size of the Internet and development of methods for Internet structure modeling is a number of ongoing studies [4].

There are two main reasons why the traditional information retrieval techniques may not be sufficiently effective in the exploration of the modern Internet. The first reason stems from the mentioned above very large size of the Internet and the very large dynamic changes in its structure and content. The second reason has to do with the existence of multiple systems describing the contents of individual Web pages, which can significantly impede analysis of their contents. A qualitative change in the efficiency of search algorithms on the Web was the result of the use of the results in their design analysis of the structure of links in the network. In particular, a link from page A to page B can be considered as a recommendation of the page B by the author of the page A. In recent years some new algorithms have been proposed based on the knowledge of the structure of Internet links. Practice shows that the effect of information retrieval algorithms of this class gives qualitatively better results than the results of the algorithms that implement the traditional methods and techniques of information retrieval.

Internet search engines use a variety of algorithms to sort Web pages based on their text content or on the hyperlink structure of the Web. This paper describes algorithms that use the hyperlink structure, called link-based algorithms: PageRank [12] and HITS [8]. The basic notion for these algorithms is the Web graph, which is a digraph with a node for each Web page and an arc between pages i and j if there is a hyperlink from page i to page j. Given a collection of Web pages linking to each other, the HITS and PageRank algorithms construct a matrix capturing the Web hyperlink structure and compute a measures of pages popularity (ranks) using linear algebra methods.

2. The PageRank algorithm

In well-known study Brin and Page [3] have proposed an algorithm for determining the ranking of Web pages called PageRank , which uses the term "weight of page". According to this proposal the weight of page depends on the number of others Web pages that point to it. The value of the weight can be used to rank the results of the query. This page rank, however, would be little resistance to a phenomenon known as spam, because it is quite easy to artificially create multiple pages pointing to the page [1]. To counteract such practices PageRank algorithm extends the basic idea of citations, taking into account the importance of each page that point to the analyzed page. This means that the definition of page weights (PageRank) is cyclic: the importance of page depends on the weight of pages pointing to it and at the same time affect the validity of the pages to which she points. Web model proposed in the work of Brin and Page [3] uses the link structure of Web site to the construction of a Markov chain with transition matrix *P*, whose elements are the probabilities pij of random events such that the user of page *i* indicates a link to the page *j*. The irreducibility of the chain guarantees that the long-run stationary vector $\boldsymbol{r}$, known as the PageRank vector, exists. Mathematically, we can think of this network as a graph, where each page is a vertex, and a link from one page to another is a graph edge. In the language of PageRank, vertices are nodes (Web pages), the edges from a node are forward links, and the edges into a node are backlinks.

2.1. The idea of PageRank model

We first present a simple definition of PageRank that captures the above intuition before describing a practical variant.

Let the pages on the Web be denoted by $1, 2, \ldots, m$. Let $N(i)$ denote the number of forward (outgoing) links from page i. Let $B(i)$ denote the set of pages that point to page i. For now, assume that the Web pages form a strongly connected graph (every page can be reached from any other page). The basic PageRank of page i, denoted by r_i, is nonnegative real number given by

$$r_i = \sum_{j \in B(i)} r_j / N(j), \quad i = 1, 2, \ldots, m . \tag{1}$$

The division by $N(j)$ captures the intuition that pages that point to page i evenly distribute their rank boost to all of the pages they point to. According to this definition, the PageRank of some page depends not only on the number of pages pointing to it, but also on their importance. The row vector $\boldsymbol{r}$ is called a PageRank vector and the value r_i is the PageRank of page i.

Effective, practical way to find PageRank vector $\boldsymbol{r}$ is using the language and methods of linear algebra. Using the linear algebra the PageRank vector $\boldsymbol{r}$ can be found by solving either the homogeneous linear system

$$(A^T - I)r^T = 0^T, \tag{2}$$

or by solving the eigenvector problem

$$r = r \cdot A, \tag{3}$$

where r^T is a column transposed vector to the row vector r, I is the identity matrix of order m, 0^T is the column vector of all 0's, and A^T is a transposed matrix of a square matrix $A = [a_{ij}]_{m \times m}$ which elements a_{ij} are defined as follows

$$a_{ij} = \begin{cases} \dfrac{1}{N(i)} & \text{if page } i \text{ points to page } j, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Both formulations are subject to an additional equation, the normalization equation $r \cdot 1^T = 1$, where 1^T is the column vector of all 1's.

Simple PageRank is well defined only if the link graph is strongly connected, where a graph is strongly connected when for each pair of nodes (i, j) there is a sequence of directed edges leading from i to j. One problem with solely using the Web's hyperlink structure to build the Markov matrix is apparent. Some rows of the matrix may contain all zeros. Thus, such a matrix is not stochastic. This occurs whenever a node contains no outlinks. Many such nodes exist on the Web. In particular, there are two related problems that arise on the real Web: rank sinks and rank leaks [1]. A group of pages pointing to each other could have some links going to the group but no links going out forms a rank sink. An individual page that does not have any outlinks constitutes a rank leak. Although, technically, a rank leak is a special case of rank sink, a rank leak causes a different kind of problem. In the case of a rank sink, nodes not in a sink receive a zero rank, which means we cannot distinguish the importance of such nodes.

Page et al. [12] suggest eliminating these problems in two ways. First, they remove all the leak nodes with out-degree 0. Second, in order to solve the problem of sinks, they introduce a decay coefficient α, $0 < \alpha < 1$, in the PageRank definition (1). In this modified definition, only a fraction α of the rank of a page is distributed among the nodes that it points to. The remaining rank is distributed equally among all the pages on the Web. Thus, the modified PageRank is [1]:

$$r_i = \alpha \sum_{j \in B(i)} r_j / N(j) + (1 - \alpha)/m, \quad i = 1, 2, ..., m \tag{5}$$

where m is the total number of nodes in the graph. Note that basic PageRank (1) is a special case of (5) that occurs when we take $\alpha = 1$.

Using the matrix A, defined by (4), is insufficient for the PageRank algorithm because the iteration using A alone might not converge properly. It can cycle or the limit may be dependent on the starting vector. Part of the explanation for this is that

the matrix $\boldsymbol{A}$ is not yet necessarily stochastic [6]. For example, if some page is a leak node than corresponding row of the matrix $\boldsymbol{A}$ contains all zeros (*0*).

Thus, to ensure that matrix $\boldsymbol{A}$ is stochastic, we must ensure that every row sums to $\boldsymbol{1}$. It can be proved that from matrix $\boldsymbol{A}$, we can obtain the stochastic matrix $\boldsymbol{S}$ as follows [6]:

$$\boldsymbol{S = A + (b^T \cdot 1)/m}\,, \tag{6}$$

where $\boldsymbol{b^T}$ is a column vector such that

$$\boldsymbol{b_i} = \begin{cases} \boldsymbol{1} & \boldsymbol{if \ \sum_{j=1}^{m} a_{ij} = 0, \ i.e., \ page \ i \ is \ a \ leak \ node,} \\ \boldsymbol{0} & \boldsymbol{otherwise.} \end{cases} \tag{7}$$

where *i=1, 2, …, m* and $\boldsymbol{1}$ is a row vector of all *1's*.
Given any stochastic matrix $\boldsymbol{S}$ we can obtain irreducible matrix $\boldsymbol{G}$ as follows [6]:

$$\boldsymbol{G = \alpha S + (1-\alpha) E}\,, \tag{8}$$

where $\boldsymbol{0<\alpha<1}$, $\boldsymbol{E = (1^T \cdot 1)/n}$ and $\boldsymbol{1^T}$, $\boldsymbol{1}$ are, respectively, the column and row vectors of all *1*'s.

Because $\boldsymbol{G}$ is stochastic (i.e., the entries in each column sum to *1*), the dominant eigenvalue of $\boldsymbol{G}$ is *1* [11]. Notice, also, that matrix $\boldsymbol{G}$ is completely positive, i.e. all elements of $\boldsymbol{G}$ are positive, although the probability of transitioning may be very small in some cases, it is always nonzero. The irreducibility adjustment insures that matrix *G* is primitive, where a nonnegative, irreducible matrix is primitive if it has only one eigenvalue on its spectral circle [10]. The matrix irreducibility implies that the power method will converge to the stationary PageRank vector $\boldsymbol{r}$. It can be shown that

$$\boldsymbol{r = r \cdot G}\,. \tag{9}$$

2.2. Computational aspects of PageRank

Although PageRank can be described using equation (1), the summation method is neither the most interesting nor the most illustrative of the algorithm's properties [1]. The preferable method is to compute the principal eigenvector of the stochastic and irreducible matrix *G* defined by (8).

One of the simplest methods for computing the principal eigenvector of a matrix is called power iteration. In power iteration, an arbitrary initial vector is multiplied repeatedly with the given matrix, until it converges to the principal eigenvector [6]. The idea of power iteration algorithm to compute the PageRank vector $\boldsymbol{r}$ is given below [1]:

1) $\boldsymbol{s} \leftarrow$ initial vector;
2) $\boldsymbol{r} \leftarrow \boldsymbol{s \cdot G;}$

3) if $\|\boldsymbol{r}-\boldsymbol{s}\|<\varepsilon$ then *end*; $\boldsymbol{r}$ is the PageRank vector;

4) $\boldsymbol{s}\leftarrow\boldsymbol{r}$;

5) goto 2,

where $\|\cdot\|$ is the measure of difference of successive iterates and ε is predetermined tolerance level (computational accuracy).

In order for the power iteration to be practical, it is not only necessary that it converge to the PageRank, but that it does so in a few iterations [1]. Theoretically, the convergence of the power iteration for a matrix depends on the eigenvalue gap, which is defined as the difference between the modulus of the two largest eigenvalues of the given matrix. Page et al. [12] claim that this is indeed the case, and that the power iteration converges reasonably fast (practically in no more than in 100 iterations). It is worth noting that in practice we are more interested in the relative ordering of the pages induced by the PageRank (since this is used to rank the pages) than the actual PageRank values themselves [1]. Thus, we can terminate the power iteration once the ordering of the pages becomes reasonably stable. Experiments [7] indicate that the ordering induced by the PageRank converges much faster than the actual PageRank.

When dealing with data sets as large as Google uses (more than eight billion web pages [5]), it is unrealistic to form a matrix $\boldsymbol{G}$ and find its dominant eigenvector. It is more efficient to compute the PageRank vector using the power method variant, where we can compute the PageRank vector $\boldsymbol{r}$ in k iteratations, $k = 1, 2, \ldots$, with the matrix $\boldsymbol{A}$ which elements are defined by (4) instead matrix $\boldsymbol{G}$ [6]:

$$\boldsymbol{r}^{(k)}=\alpha\boldsymbol{r}^{(k-1)}\boldsymbol{A}+\left[(\alpha\boldsymbol{r}^{(k-1)}\boldsymbol{b}^{T}+(1-\alpha))/m\right]\cdot\boldsymbol{1}. \tag{10}$$

One of the benefits of using the above power method variant to compute the PageRank vector is the speed with which it converges. Specifically, the power method on matrix $\boldsymbol{G}$ converges at the rate at which a quantity α^{k} goes to zero. This gives the ability to estimate the number of iterations required to reach a tolerance level measured by $\left\|\boldsymbol{r}^{(k)}-\boldsymbol{r}^{(k-1)}\right\|$. The number of needed iterations $\boldsymbol{k}$ is approximately $\log\varepsilon/\log\alpha$, where ε the tolerance level [9].

It is worth noting that the founders of Google, Lawrence Page and Sergey Brin, use $\alpha=0.85$ and find success with only **50** to **100** power iterations [9].

3. Test the effectiveness of the PageRank algorithm

3.1. General assumptions

Using an iterative algorithm, in practice, according to the formula (7) is conditioned to its efficiency, which in this case is measured by the number of iterations to be done to accuracy that is required for elements of $\boldsymbol{r}$ vector for a fixed value of the α coefficient. The independent parameters of simulation experiments were the number of Web pages and their links and the density of these links. In accordance with what has been said, a network of websites is mapped in the form of a directed graph without loops, where the arc shows the indication (the link) from one page to another, such as a linked thematically. As a measure of the density of links between Web pages for the simulation experiments the λ coefficient is assumed, hereafter referred to as the density coefficient adjacency matrix of the Web pages graph comprising $\boldsymbol{m}$ websites, determined from the following relationship:

$$\lambda = \frac{\sum_{i=1}^{m} N(i)}{m^2 - m}. \tag{11}$$

Experiments were performed on randomly generated adjacency matrix with a predetermined value λ coefficient. Due to the limited possibility of presentation of the results of experiments will be based at most 20 Web pages networks (20 dimensional adjacency matrix), which does not detract from the generality of observations and conclusions.

Experiments conducted to evaluate the effectiveness of an iterative algorithm of determining the $\boldsymbol{r}$ vector were aimed at:

- assessment of the number of iterations of the algorithm and the clarity of the resulting $\boldsymbol{r}$ vector depending α values at a fixed value of the λ coefficient for the Web with a fixed number of pages,
- assessment of the number of iterations of the algorithm, depending on the values of the coefficients α and λ for the Web with a fixed number of pages,
- assessment of the impact of coefficients α and λ for the Web fixed number of pages on the number of iterations of the algorithm required to achieve of $\boldsymbol{r}$ vector of highest distinctness,
- assessment of the impact the accuracy of determining the elements of the $\boldsymbol{r}$ vector on the number of iterations of the algorithm.

3.2. Assessment of the number of iterations of the algorithm and the clarity of the resulting *r* vector depending α values at a fixed value of the λ coefficient for the Web with a fixed number of pages

The research was conducted with the following assumptions:

- *20* Web pages were considered,
- for considered Web the adjacency matrix is a description of a graph without loops, with the density values $\boldsymbol{\lambda = 0.1}$.

Fig. 1 shows graphs of the PageRank coordinates of $\boldsymbol{r}$ vector for three values of the coefficient $\boldsymbol{\alpha}$, equal to 0.1, 0.5 and 0.99, respectively.

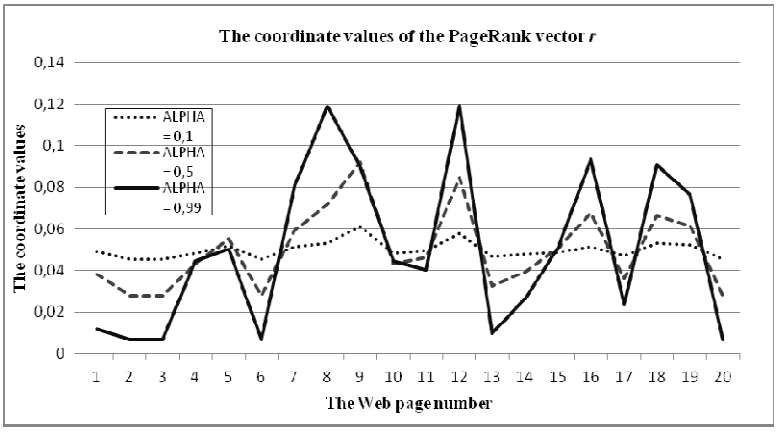

Figure 1. Graphs coordinate values of the PageRank vector r for values $\alpha = 0.1$, 0.5 and 0.99

Analysis of the results of the research confirm the supposition any increased expressiveness assessment of Web pages by PageRank algorithm with increasing $\boldsymbol{\alpha}$ coefficient, wherein the assessments expressiveness was measured using well-known in statistics, the coefficient of variation (ratio of the standard deviation of the coordinate vector $\boldsymbol{r}$ to their mean value), as:

$$\boldsymbol{V_r = \frac{s_r}{\bar{r}}}. \tag{12}$$

The values of the variation coefficient of $\boldsymbol{r}$ vector depending on the value of the $\boldsymbol{\alpha}$ coefficient shows the table 1.

Table 1. The values of the coefficient of variation of *PageRank* $\boldsymbol{r}$ vector depending on the value of the $\boldsymbol{\alpha}$ coefficient

α	0,1	0,5	0,99
V_r	0,0816	0,3768	0,7528

The desired increase of expressiveness coefficient of the $\boldsymbol{r}$ vector by increasing the value of the $\boldsymbol{\alpha}$ coefficient results in undesirable exponential increase of the number of iterations of the algorithm of calculating the $\boldsymbol{r}$ vector, as shown in Fig. 2.

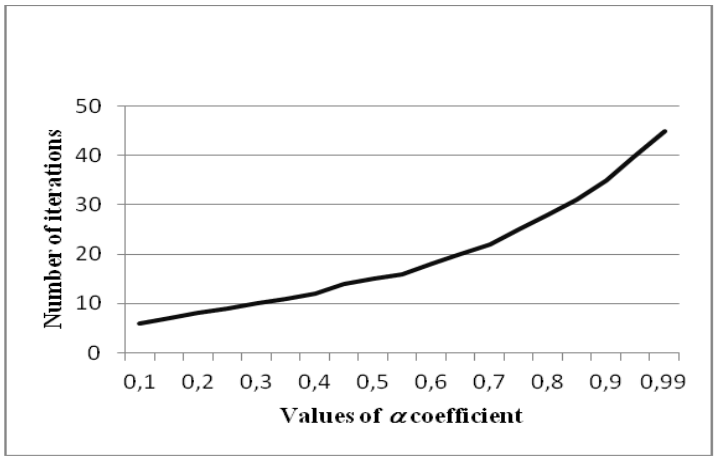

Figure 2. Plot of the number of iterations of the PageRank algorithm in the process of determining the $\boldsymbol{r}$ vector for $\boldsymbol{\alpha}$ values
Source: own preparation

For the experiment, the number $\boldsymbol{L}$ of iterations PageRank algorithm depending on $\boldsymbol{\alpha}$ values can be estimated with high accuracy by using the following relationship:

$$\boldsymbol{L = 5{,}6815 \cdot e^{0{,}107\,\alpha}} \tag{13}$$

3.3. Assessment of the number of iterations of the algorithm, depending on the values of the coefficients α and λ for the Web with a fixed number of pages

Experiments were performed for adjacency matrices of fixed dimensions (20×20) and changing values of $\boldsymbol{\lambda}$ coefficient ranging from 0.1 to 0.9 in steps of 0.1 and for fixed values of $\boldsymbol{\alpha}$ coefficient. The number of iterations needed to determine the $\boldsymbol{r}$ vector for the assumed accuracy of its coordinates have been measured. The results are shown in Table 2.

Table 2 shows that the increase in the value of $\boldsymbol{\lambda}$ coefficient of the adjacency matrix (increasing the number of links between the pages) will reduce the number of iterations of the PageRank algorithm to determine the $\boldsymbol{r}$ vector desired accuracy for fixed $\boldsymbol{\alpha}$ coefficient. Number of iterations of the algorithm varies exponentially for rare adjacency matrix ($\boldsymbol{\lambda} = 0.1$) by changing the linear for the adjacency matrix

of λ = 0.5, up to by parabola negative coefficient directional - for a dense matrix, i.e. for λ = 0.9. However, it seems that the actual Web networks are rather rare, characterized by the values of the coefficient λ < 0.5, therefore, to be expected in such cases, the exponential increase in the number of iterations of the PageRank algorithm to achieve the desired $\mathbf{r}$ vector with increasing α values.

Table 2. Number of iterations of the *PageRank* algorithm as a function of the α and λ coefficients

α	λ coefficient								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	6	5	5	5	4	4	4	4	3
0.2	8	6	6	5	5	5	5	4	4
0.3	10	8	7	6	6	6	5	5	4
0.4	12	9	8	7	7	7	6	5	5
0.5	14	11	9	8	7	7	6	6	5
0.6	18	12	10	9	8	8	7	6	5
0.7	21	15	11	10	9	9	7	7	6
0.8	26	17	12	11	9	9	8	7	6
0.9	34	20	14	12	10	10	8	7	6
0.99	44	24	15	12	11	11	9	8	6

3.4. Assessment of the impact of coefficients α and λ for the Web fixed number of pages on the number of iterations of the algorithm required to achieve of *r* vector of highest distinctness

Evaluation of the impact speed for obtaining the highest expressiveness of the $\mathbf{r}$ vector by the algorithm based on the change both the value of the α and λ coefficients was made indirectly through the distances analysis of $\mathbf{r}$ vectors obtained for different values of the α coefficient from the vector which is characterized by the greatest expressiveness, i.e. the vector obtained for α = 0.99. Among the known distance measures between numerical vectors in experiment selected 7 following, the most frequently used in practice: Euclidean, Chebyshev, Manhattan, Pearson, tangents, angular and exponential module. The research was conducted for the adjacency matrix of fixed dimensions (20×20) and selected values of λ coefficient. Fig. 3 shows the changes in the Euclidean distances between the $\mathbf{r}$ vectors and the vector with the highest expressiveness (for α = 0.99) as a function of the α coefficient for the adjacency matrix of values with λ coefficient equals to 0.1, 0.5 and 0.9.

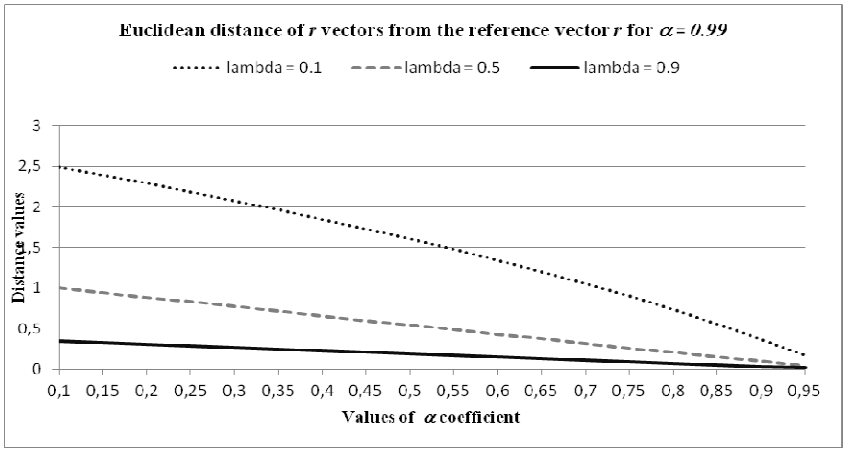

Figure 3. Changes of the Euclidean distance of $\boldsymbol{r}$ vectors to the vector with the greatest distinctness (for $\alpha = 0.99$) as a function of the α coefficient for the adjacency matrix with λ coefficient equals to 0.1, 0.5 and 0.9

The waveforms similar to shown in Fig. 3 was also observed if the distance between $\boldsymbol{r}$ vectors was measured by using the other distance measures. Thus justified hypothesis that for the rare adjacency matrix ($\lambda = 0.1$) the approximation of the $\boldsymbol{r}$ vectors (decreasing distances), calculated for increasing values of α coefficient from the reference vector is much faster than for the denser of adjacency matrix. Based on the results of the experiment can be concluded that the dense adjacency matrix ($\lambda = 0.9$) the $\boldsymbol{r}$ vector (for small values of α obtained using a small number of iterations of the investigated algorithm) will be a good approximation of the high expressiveness $\boldsymbol{r}$ vector, obtained for the high value of α coefficient, but at the expense of a larger number of iterations. This conclusion may have important practical significance when examined pages ranking algorithm would be used in large networks with highly dynamic changes in the density of the relationship between the Web pages.

4. Conclusions

Many of today's search engines use a two-step process to retrieve pages related to a user's query. In the first step, traditional text processing is done to find all documents using the query terms, or related to the query terms by semantic meaning. This can be done by a lookup into an inverted file, with a vector space method, or with a query expander that uses a thesaurus. With the massive size of the Web, this first step can result in thousands of retrieved pages related to the query.

To make this list manageable for a user, many search engines sort this list by some ranking criterion. One popular way to create this ranking is to exploit the additional information inherent in the Web due to its hyperlinking structure. Thus, link analysis has become the means to ranking. One successful and well-publicized link-based ranking system is PageRank, the ranking system used by the Google search engine [2].

From the foregoing considerations, it follows that there is possibility of practical achieve time savings associated with the ranking Web pages, by substituting the result (page ranking), obtained through the implementation of the PageRank algorithm, by the approximate ranking of these pages, based on the analysis of their input stages, i.e., the number of appeals from other pages.

REFERENCES

[1] Arasu A., Cho J., Garcia-Molina H., Paepcke A., Raghavan S. (2001) *Searching the Web*. ACM Transactions on Internet technology, 1 (1): 2-43.

[2] Blachman N., Fredricksen E. Schneider F. (2003) *How to Do Everything with Google.* McGraw-Hill.

[3] Brin S., Page L. (1998) *The anatomy of a large-scale hypertextual Web search engine*. Comput. Netw. ISDN Syst. 30: 107–117.

[4] Cho J., Garcia-Molina H. (2003) *Estimating frequency of change*. Journal ACM Transactions on Internet Technology, 3 (3): 256- 90.

[5] Coughran B. (2005) *Google's index nearly doubles*, Google Inc. http://googleblog.blogspot.com/2004/11/googles-index-nearly-doubles.html

[6] Golub G., Van Loan C.F. (1989) *Matrix Computations*. 2nd ed. Johns Hopkins University Press, Baltimore.

[7] Havelivala T. (1999) *Efficient computation of PageRank*. Tech. Rep. 1999-31. Computer Systems Laboratory, Stanford University, Stanford, CA. http://dbpubs.stanford.edu/ pub/1999-31

[8] Kleinberg J.M. (1999) *Authoritative Sources in a Hyperlinked Environment*. Journal of ACM, 46(5): 604–632.

[9] Langville A.N., Meyer C.D. (2004) *The Use of the Linear Algebra by Web Search Engines*, http://meyer.math.ncsu.edu/Meyer/PS_Files/IMAGE.pdf.

[10] Lawrence S., Giles C. (1999) *Accessibility of information on the web*. Nature 400, 107–109.

[11] Meyer C. D. (2000) *Matrix Analysis and Applied Linear Algebra*. The Society for Industrial and Applied Mathematics, Philadelphia: 490–693.

[12] Page L., Brin S., Motwani R., Winograd T. (1998) *The PageRank Citation Ranking: Bringing Order to the Web*. Tech. Rep. Computer Systems Laboratory, Stanford University, Stanford, CA.