# A Generalization of the PageRank Algorithm

Z. Bahrami Bidoni, R. George, K.A. Shujaee
Department of Computer and Information Systems
Clark Atlanta University
Atlanta, GA
{zeynab.bahrami, rgeorge, kshujaee}@cau.edu

*Abstract*— **PageRank is a well-known algorithm that has been used to understand the structure of the Web. In its classical formulation the algorithm considers only forward looking paths in its analysis- a typical web scenario. We propose a generalization of the PageRank algorithm based on both out-links and in-links. This generalization enables the elimination network anomalies- and increases the applicability of the algorithm to an array of new applications in networked data. Through experimental results we illustrate that the proposed generalized PageRank minimizes the effect of network anomalies, and results in more realistic representation of the network.**

***Keywords- Search Engine; PageRank; Web Structure; Web Mining; Spider-Trap; dead-end; Taxation;Web spamming.***

## I. INTRODUCTION

With the rapid growth of the Web, users can get easily lost in the massive, dynamic and mostly unstructured network topology. Finding users' needs and providing useful information are the primary goals of website owners. Web structure mining [1],[2],[3] is an approach used to categorize users and pages. It does so by analyzing the users' patterns of behavior, the content of the pages, and the order of the Uniform Resource Locator (URL) that tend to be accessed. In particular, Web structure mining plays an important role in guiding the users through the maze. The pages and hyperlinks of the World-Wide Web may be viewed as nodes and arcs in a directed graph. The problem is that this graph is massive, with more than a trillion nodes, several billion links, and growing exponentially with time. A classical approach used to characterize the structure of the Web graph through PageRank algorithm, which is the method of finding page importance.

The original PageRank algorithm [3],[4],[5] one of the most widely used structuring algorithms, states that a page has a high rank if the sum of the ranks of its backlinks is high. Google effectively applied the PageRank algorithm, to the Google search engine [4]. Xing and Ghorbani [6] enhanced the basic algorithm through a Weighted PageRank (WPR) algorithm, which assigns a larger rank values to the more important pages rather than dividing the rank value of a page evenly among its outgoing linked pages. Each outgoing link page gets a value proportional to its popularity (its number of in-links and out-links). Kleinberg [7] identifies two different forms of Web pages called hubs and authorities, which lead to the definition of an iterative algorithm called Hyperlink Induced Topic Search (HITS) [8].

Bidoki and Yazdani [9] proposed a novel recursive method based on reinforcement learning [10] that considers distance between pages as punishment, called "DistanceRank" to compute ranks of web pages in which the algorithm is less sensitive to the "rich-get-richer" problem [9],[11] and finds important pages faster than others. The DirichletRank algorithm has been proposed by X. Wang et al [12] to eliminate the zero-one gap problem found in the PageRank algorithm proposed by Brin and Page [4]. The zero-one gap problem occurs due to the ad hoc way of computing transition probabilities. They have also proved that this algorithm is more robust against several common link spams and is more stable under link perturbations. Singh and Kumar [13] provide a review and comparison of important PageRank based algorithms.

As search engines are used to find the way around the Web, there is an opportunity to fool search engines into leading people to particular page. This is the problem of web spamming [14], which is a method to maliciously induce bias to search engines so that certain target pages will be ranked much higher than they deserve. This leads to poor quality of search results and in turn reduces the trust in the search engine. Consequently, anti-spamming is a big challenge for all the search engines. Earlier Web spamming was done by adding a variety of query keywords on page contents regardless of their relevance. In link spamming [15], the spammers intentionally set up link structures, involving a lot of interconnected pages to boost the PageRank scores of a small number of target pages. This link spamming does not only increasing the rank gains, but also makes it harder to detect by the search engines. It is important to point out that link spamming is a special case of the spider-traps [16]. At the present time, the Taxation method [16] is the most significant way to diminish the influence of the spider-traps and dead-ends by teleporting the random surfer to a random page in each iteration.

This article has two main contributions: First, we present a generalized formulation of the PageRank algorithm based on transition probabilities, which takes both in-link and out-links of node and their influence rates into account in order to calculate PageRanks. This would permit the application of this approach to a wide variety of network problems that require consideration of the current state values (and PageRank) as a function of past state transitions. Second, we describe a novel approach of adding virtual edges to a graph that permits more realistic computations of PageRank,

 ISBN: 978-1-61208-324-7 

negating the effect of network anomalies such as spider-traps and dead-ends.

The paper is organized as follows. In Section 2, a brief background review of the basic concepts for computing PageRanks based on transition probabilities is presented and the problems related to network anomalies such as spider-traps and dead-ends together with their solution method based on Taxation is stated. In Section 3, we introduce the proposed general approach for determining PageRank. In Section 4, we apply our PageRank method to a typical graph with all types of possible structures and inter/ intra-correlations and compare our results with the baseline technique. In Section 5, we conclude by describing the contribution of our method and discuss its results.

## II. OVERVIEW ON THE PAGERANK APPROACH BASED ON TRANSITION PROBABILITIES

PageRank is a function that assigns a real number to each page in the Web. We begin by defining the basic, idealized PageRank, and follow it by modifications that are necessary for dealing with some real-world problems concerning the structure of the Web. Imagine surfing the Web, going from page to page by randomly (random surfer) choosing an outgoing link from one page to get to the next. This can lead to dead-ends at pages with no outgoing links, or cycles around cliques of interconnected pages. This theoretical random walk is known as a Markov chain or Markov process [16],[17].

In general, we can define the transition matrix of the Web to describe what happens to random surfers after one step. This matrix *M* has *n* rows and columns, if there are *n* pages. The element $m_{ij}$ in row *i* and column *j* has value *1/k* if page *j* has *k* arcs out, and one of them is to page *i*. Otherwise, $m_{ij} = 0$. The probability distribution for the location of a random surfer can be described by a column vector whose *j*th component is the probability that the surfer is at page *j*. This probability is the (idealized) PageRank function.

Suppose we start a random surfer at any of the *n* pages of the Web with equal probability. Then the initial vector $v_0$ will have *1/n* for each component. If *M* is the transition matrix of the Web, then after one step, the probability distribution of the surfer place will be $Mv_0$, after two steps it will become $M\left(Mv_0\right) = M^2 v_0$, and so on. In general, multiplying the initial vector $v_0$ by *M* a total of i times will give us the distribution of the surfer after i steps.

This sort of behavior is an example of a Markov processes. It is known that the distribution of the surfer approaches a limiting distribution *v* that satisfies $v = Mv$, provided two conditions are met:

*1) The graph is strongly connected; that is, it is possible to get from any node to any other node.*

*2) There are no dead-ends: nodes that have no arcs out.*

In fact, because *M* is stochastic, meaning that each of its columns adds up to 1, v is the principal eigenvector. Note also that, because *M* is stochastic, the eigenvalue associated with the principal eigenvector is 1.The principal eigenvector of *M* tells us where the surfer is most likely to be after infinite steps *i*. The intuition behind PageRank is that the more likely a surfer is to be at a page, the more important the page is. We can compute the principal eigenvector of *M* by starting with the initial vector $v_0$ and multiplying by *M* some number of times, until the vector we get shows little change at each round. In practice, for the Web itself, 50–75 iterations are sufficient to converge to within the error limits of double-precision arithmetic.

### A. Structure of the Web

It would be nice if Web pages were strongly connected. However, it is not the case in practice. An early study of the Web found it to have the structure shown in Figure 1. There is a large strongly connected component (SCC), but there were several other portions that were almost as large [18].

- The **in-component**, consisting of pages that could reach the SCC by following links, but were not reachable from the SCC.
- The **out-component**, consisting of pages reachable from the SCC but unable to reach the SCC.
- **Tendrils**, which are of two types. Some tendrils consist of pages reachable from the in-component but not able to reach the in-component. The other tendrils can reach the out-component, but are not reachable from the out-component.

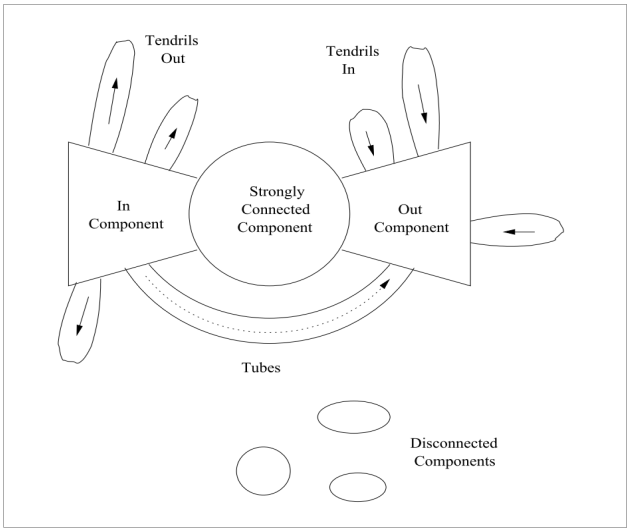


Figure 1. The "bowtie" representation of the Web [22]

In addition, there were small numbers of pages found either in

- Tubes, which are pages reachable from the in-component and able to reach the out-component, but unable to reach the SCC or be reached from the SCC.
- Isolated components that are unreachable from the large components (the SCC, in- and out-components) and unable to reach those components.

As a result, PageRank is usually modified to prevent such anomalies. There are, in principle, two problems we need to avoid. First, is the dead-end - a page that has no links out- which will bring a zero column in the forward transition matrix, and consequently it will cause all PageRanks to become zero. The second problem is groups of pages that all have out-links but they never link to any other pages. These structures are called spider-traps. Both these problems are solved by a method called "taxation," where we assume a random surfer has a finite probability of leaving the Web at any step, and new surfers are started at each page.

### B. Taxation

To avoid the problem of spider-trap or dead-end, we modify the calculation of PageRank by allowing each random surfer a small probability of teleporting to a random page, rather than following an out-link from their current page. The iterative step, where we compute a new vector estimate of PageRanks $v'$ from the current PageRank estimate $v$ and the transition matrix $M$ is

$$v' = \beta M v + (1-\beta)e / n \tag{1}$$

Where $\beta$ is a chosen constant, usually in the range 0.8 to 0.9, $e$ is a vector of all 1's with the appropriate number of components, and n is the number of nodes in the Web graph. The term $\beta Mv$ represents the case where, with probability $\beta$, the random surfer decides to follow an out-link from their present page. The term $(1-\beta)e/n$ is a vector each of whose components has value $(1-\beta)/n$ and represents the introduction, with probability $1-\beta$, of a new random surfer at a random page.

Although by employing this formulation, the effect of spider-trap and dead-end is controlled and the PageRank is distributed to each of other nodes, components of spider-trap still are managed to get most of the PageRank for themselves. Therefore, the PageRanks of nodes are still unreasonable. For instance, in Figure 2. , C is a simple spider trap of one node and the transition matrix is as follows:

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

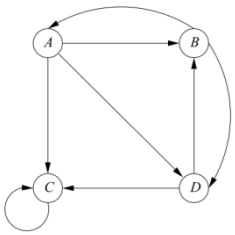


Figure 2. A graph with a one-node spider trap

If we perform the usual iteration to compute the PageRank of the nodes, we get

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 11/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 5/48 \\ 7/48 \\ 29/48 \\ 7/48 \end{bmatrix} \begin{bmatrix} 21/288 \\ 31/288 \\ 205/288 \\ 31/288 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{3}$$

As predicted, all the PageRank is at C, since once there a random surfer can never leave. To avoid the problem illustrated, we modify the calculation of PageRank by the Taxation method. Thus, the equation for the iteration becomes

$$\mathbf{v}' = \begin{bmatrix} 0 & 2/5 & 0 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 4/5 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 1/20 \\ 1/20 \\ 1/20 \\ 1/20 \end{bmatrix} \tag{4}$$

Notice that we have incorporated the factor $\beta$ into $M$ by multiplying each of its elements by *4/5*. The components of the vector $(1-\beta)e/n$ are each *1/20*, since $1-\beta = 1/5$ and $n=4$. The first iteration:

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix} \begin{bmatrix} 41/300 \\ 53/300 \\ 153/300 \\ 53/300 \end{bmatrix} \begin{bmatrix} 543/4500 \\ 707/4500 \\ 2543/4500 \\ 707/4500 \end{bmatrix} \cdots \begin{bmatrix} 15/148 \\ 19/148 \\ 95/148 \\ 19/148 \end{bmatrix} \tag{5}$$

By being a spider trap, C has still managed to get more than half of the PageRank for itself. However, the effect has been limited, and each of the nodes gets some of the PageRank.

## III. A Generalized Method

In web arena, a link by important pages will impact on significance of a page. However, there are other networks in which not just in-link but out-links are also weighty. For instance, in social networks, connecting to eminent people (out-link) is as crucial as being connected by key persons (in-link) in evaluating the degree of prominence of a member. Therefore, sometimes sorting and grading nodes of a graph only based on in-links will result in an incorrect evaluation. So, we take out-links and the rate of their impacts with respect to in-links into our computations.

### A. Algorithm

Suppose we start as a random surfer at any of the $n$ pages of the Web with equal probability. Then the initial vector will have *1/n* for each component. If $M_f$ is the forward transition matrix of the Web, then after one forward step, the probability distribution of the next surfer place will be $M_f v_0$ and if $M_b$ is the backward transition matrix of the Web, then after one backward step, the probability distribution of the previous surfer place will became $M_b v_0$. Also, we consider the importance weight factor of both in-links ($\beta$) and out-links ($1-\beta$).

Note that equation $\left(\beta M_f + (1-\beta) M_b\right)$ is the linear combination of both next and previous surfer place, and it is

also stochastic because it is a linear combination of two stochastic matrices. So its eigenvalue associated with the principal eigenvector will be 1. The principal eigenvector of $(\beta M_f + (1-\beta) M_b)$ tells us where the surfer is most likely to be after a long time. Recall that the intuition behind PageRank is that the more likely a surfer is to be at a page, the more important the page is. We can compute the principal eigenvector of $(\beta M_f + (1-\beta) M_b)$ by starting with the initial vector $v_0$ and multiplying by $(\beta M_f + (1-\beta) M_b)$ some number of times, until the vector we get shows little change at each round. Considering this matrix instead of $M_f$ has two advantages: First, in computing PageRank of a node, the importance of its neighbors with both types of relationship (out-link and in-link) and their arbitrary impact rates (parameter $\beta$) have taken into account. Second, by using this method, we do not have the problems about dead-ends and spider-traps because we take the linear combination of entering probability from and exiting probability to other nodes in our computation. Therefore, in case $\beta \neq 0$ and $\beta \neq 1$, the columns related to dead-ends are not completely zero. Likewise, for the spider-trap columns, probabilities related to other nodes are not zero and they cannot absorb more unreasonable rank to themselves. About cases $\beta = 1$ or $\beta = 0$, in the following, we proposed another idea (adding virtual edges) by which the random surfer can exit from dead-ends and spider-traps.

The proposed algorithm is as follows:

**Step 1: finding Forward and Backward transition matrices.**

**Step 2: considering appropriate formula and keep iterating until it gets converged.**

In this step, three possible conditions can exist which are characterized as following:

**Case 1:** $\beta \neq 0$ **and** $\beta \neq 1$**.** It means that both forward and backward trends are important to calculate PageRanks. Thus, we only need to calculate the eigenvector of matrix $(\beta M_f + (1-\beta) M_b)$.

**Case 2:** $\beta = 1$ So, we need only the forward matrix to calculate PageRanks. If there are not a dead-end or a spider-trap in the graph, the vector of PageRanks is the eigenvector of $M_f$. If there are dead-ends or spider-traps, the eigenvector of $M_f$ assigns most of PageRank to spider-traps and dead-ends that is not real. Thus we add enough virtual out-links to remove these spider and dead-end situations. For each dead-end and spider-trap, we will consider a virtual edge in which source of them are dead-ends and one member of each spider-traps, respectively. Also, their destinations can be any arbitrary nodes, excepting those of dead-end and spider-traps (see Figure 3. Green color edges). Hence, If assumed $v$ is eigenvector of matrix $M_f^{'}$ (forward transition matrix after adding virtual links), in order to find final PageRanks of vertices, we have to remove effect of these virtual links on PageRanks by calculating the following equation $v - (M_f^{'} - M_f)v$ .

**Case 3:** $\beta = 0$. Here only backward trend (out-links) is important to consider for calculation of PageRanks. So we only need backward matrix to determine PageRanks. If there are not in-component or in-tendril vertices in the graph, vector of PageRanks is eigenvector of $M_b$. If there are in-component or in-tendril vertices, eigenvector of $M_b$ assigns most of PageRank to in-component and in-tendril vertices, which is not real. Thus we add enough virtual in-links to remove these in-component and in-tendril situations then after computing eigenvector of new backward matrix $M_b^{'}$, we have to remove effect of these virtual links on PageRanks (see Figure 3. Red color edges). If suppose $v$ is eigenvector of matrix $M_b^{'}$ (backward transition matrix after adding virtual links). The final PageRanks of vertices would be $v - (M_b^{'} - M_b)v$ .

**Step 3: normalize PageRank vector to find distribution probability of vertices.**

As shown below, if we consider a matrix include the importance of pairwise comparison of vertices (A), eigenvector of this matrix would be distribution probability of vertices.

Note that, W is vector distribution probability of vertices that sum of its components is 1 and also $w_i$ is amount of vertex i's importance. So, instead of $w_i / w_j$ in matrix A, we let $p_i / p_j$ , which $p_i$ , $p_j$ are PageRanks of nodes i, j. We calculate eigenvector of matrix A and to get the distribution probability of vertices.

$$AW = \begin{bmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \vdots & \cdots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = n * \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = nW \quad (6)$$

*B. Biased Random Walk*

In order to bias the rank of all nodes with respect to a special subset of nodes, we use the Biased Random Walk method in which the random surfer, in each iteration, will jump on one of the member of the subset with equal probability. Its most important application is topic-sensitive PageRank [19] in search engines. The consequence of this approach is that random surfers are likely to be at an identified page, or a page reachable along a short path from

one of these known pages, because the pages they link to are also likely to be about the same topic. The mathematical formulation for the iteration that yields topic-sensitive PageRank is similar to the equation we used for general PageRank. The only difference is how we add the new surfers. Suppose S is a set of integers consisting of the row/column numbers for the pages we have identified as belonging to a certain topic (called the teleport set). Let $e_s$ be a vector that has 1 in the components in *S* and 0 in other components. Then the topic-sensitive PageRank for *S* is the limit of the iteration

$$v' = \alpha(\beta M_f + (1-\beta)M_b)v + (1-\alpha)e_s / |s| \quad 0.8 \le \alpha \le o.9 \tag{7}$$

Here, as usual, *M* is the transition matrix of the Web, and |*S*| is the size of set *S*.

## IV. THE EXPERIMENT

Figure 3. is a graph with 20 vertices that include all kinds of network artifacts mentioned in section 2.

SCC:{1,2,4,5,7,8,9,10,15,17,18,20} TUBE:{16-6}

OUT-COMPONENT:{6,11,12} IN-COMPONENT:{3,13,16}
OUT-TENDRIL:{14} IN-TENDRIL:{19}

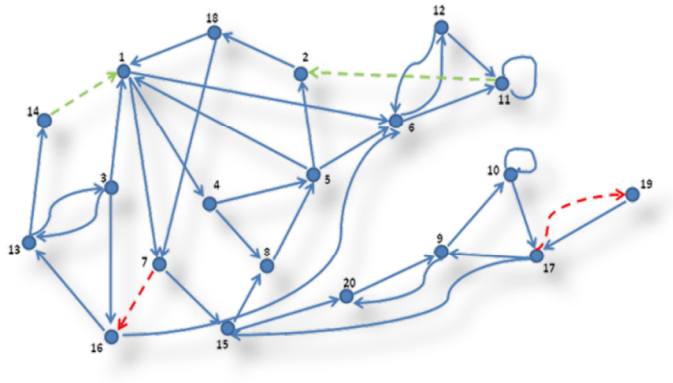

Figure 3. Synthetic Graph Example

In case 2 ( $\beta = 1$ ), there are a dead-end situation on vertex 14 and a spider-trap situation on set of vertices {6, 11, 12}, and in order to remove the dead-end and the spider-trap consider 2 virtual out-link (green edges) on these vertices. Also in case 3 ( $\beta = 0$ ), there are in-component situation on set of vertices {3, 13, 16}, and in order to remove negative PageRank consider 2 virtual in-link (red edges) on these vertices. For completeness, we also compute the biased random walk on case1. Comparing the results with case1, TABLE I. , it is clear that PageRanks are biased on set S={2, 4, 7, 18}. As we expect, rank of nodes of set *S* and nodes that are pointed by set *S* get higher ranks.

TABLE I. PAGERANK VECTOR AT CASES 1, 3, AND BIASED RANDOM WALK.

| **Results of case 1 ( $\beta = 0.7$ )** | | **Results of the biased random walk on case1** | | **Results of case 3 ( $\beta = 0$ )** | |
|---|---|---|---|---|---|
| ***Nodes number*** | ***PageRank*** | ***Nodes number*** | ***PageRank*** | ***Nodes number*** | ***PageRank*** |
| 11 | 0.945 | 5 | 0.9937 | 17 | 0.57916 |
| 12 | 0.2177 | 11 | 0.9878 | 10 | 0.38611 |
| 6 | 0.1767 | 18 | 0.9703 | 13 | 0.36037 |
| 9 | 0.0703 | 1 | 0.9432 | 1 | 0.27028 |
| 10 | 0.0632 | 7 | 0.9013 | 3 | 0.27028 |
| 5 | 0.0601 | 15 | 0.8513 | 5 | 0.25741 |
| 1 | 0.0543 | 2 | 0.7444 | 9 | 0.25741 |
| 20 | 0.0527 | 4 | 0.6847 | 7 | 0.24454 |
| 15 | 0.0495 | 6 | 0.65 | 4 | 0.19305 |
| 17 | 0.045 | 8 | 0.6414 | 19 | 0.19305 |
| 8 | 0.036 | 9 | 0.5045 | 16 | 0.18018 |
| 7 | 0.029 | 20 | 0.4878 | 2 | 0.16731 |
| 4 | 0.0272 | 12 | 0.3659 | 18 | 0.16731 |
| 18 | 0.025 | 10 | 0.3204 | 8 | 0.1287 |
| 3 | 0.0237 | 17 | 0.2976 | 15 | 0.1287 |
| 13 | 0.023 | 3 | 0.1628 | 20 | 0.1287 |
| 16 | 0.0223 | 13 | 0.1144 | 12 | 1.14E-17 |
| 2 | 0.0216 | 16 | 0.0923 | 6 | 7.34E-18 |
| 14 | 0.0081 | 19 | 0.0386 | 11 | 0 |
| 19 | 0.0068 | 14 | 0.035 | 14 | 0 |

TABLE II. COMPARING RESULTS OF THE ALGORITHM AND TAXATION METHOD TO AVOID ANOMALIES IN CASE 2 ( $\beta = 1$ )

| **Using virtual edges** | | **Taxation** | |
|---|---|---|---|
| ***nodes no*** | ***PageRank*** | ***nodes no*** | ***PageRank*** |
| 9 | 0.508068237 | 11 | 0.83086 |
| 10 | 0.508068237 | 9 | 0.25352 |
| 20 | 0.381051178 | 10 | 0.22903 |
| 2 | 0.265581124 | 20 | 0.19944 |
| 17 | 0.254034118 | 15 | 0.15968 |
| 15 | 0.254034118 | 6 | 0.1495 |
| 5 | 0.173205081 | 5 | 0.14569 |
| 18 | 0.161658075 | 17 | 0.14155 |
| 8 | 0.15011107 | 8 | 0.11547 |
| 1 | 0.138564065 | 1 | 0.11197 |
| 6 | 0.138564065 | 7 | 0.08907 |
| 7 | 0.127017059 | 12 | 0.08748 |
| 11 | 0.103923048 | 18 | 0.07921 |
| 12 | 0.069282032 | 2 | 0.06521 |
| 4 | 0.046188022 | 4 | 0.05567 |
| 3 | 7.50E-17 | 13 | 0.0528 |
| 13 | 2.12E-17 | 3 | 0.04612 |
| 16 | 1.16E-17 | 14 | 0.04612 |
| 14 | 1.02E-17 | 16 | 0.0369 |
| 19 | 0 | 19 | 0.02386 |

Comparing the results of the Taxation method and our proposed method, TABLE II. , obviously we can realize that our approach produces more reasonable outcomes. Because, as it is shown in the TABLE II, node 9 is the junction of two cycles, all nodes of these cycles are from SCC part of the graph, so the random surfer is most likely on it. The nodes 10 and 20 have higher rank after 9, because they have in-link from the node 9. The rank of node 5 cannot be higher than 17 because the node 17 is a member of the cycle consist of

 ISBN: 978-1-61208-324-7 

node 9 and 10. In Taxation result, the nodes with spider-trap situation such as 6 and 11 got higher and vertices 2 and 18 got lower PageRank than our proposed approach results. Also, for other vertices, their ranks are either the same or very close to each other's.

## V. CONCLUSION

In this paper, the fundamental idea of Web Structure mining and Web Graph is explained in detail to have a generic understanding of the data structure used in web. The main purpose of this paper is to present the new PageRank based algorithms and compare that with the previous algorithms.

The proposed method generalizes the approach of finding PageRank based on transition probabilities by considering the arbitrary impact rates of both out-links and in-links, in order to include all possible cases because there are some conditions in which out-links have also an influence on PageRank of nodes. Moreover, it prevents that spider-traps and dead-ends have a high unreasonable rank and assign higher PageRanks to themselves. The noticeable weak point of previous method is that it assigns more unreasonable PageRank to spider-traps and dead-ends, and also reduces PageRank of SCC vertices. But in our approach this problem has been solved, because by adding virtual edges, random surfers will not stop on spider-traps and dead-ends. According to [13], DirichletRank has been so far the best method amongst previous methods, capable of diminishing the impact of link spamming (a special case of spider-traps) and dead-end problem that is, however, only applicable to backward analysis. Our approach in comparison with their method is general for more types of networks and simpler to understand and implement. Also, by using ideas suggested in this paper, in any possible cases, PageRanks is insulated from the influence of anomalies including in/out-tendrils and in/out-components.

The generalization of the PageRank algorithm to include forward and backward links into a node makes this approach applicable to new domains beyond web mining and search engines. We are currently exploring the application of the new generalized algorithm to the analysis of network data for instance using PageRank as a measurement of node's activity score [20] to find communities.

## ACKNOWLEDGMENT

This research is funded in part by the Army Research Laboratory under Grant No: W911NF-12-2-0067 and Army Research Office under Grant Number W911NF-11-1-0168. Any opinions, findings, conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of the sponsor.

## REFERENCES

[1] R. Kosala and H. Blockeel, "Web mining research: A survey," ACM SIGKDD Explorations, 2(1), 2000, pp. 1–15.

[2] S. Madria, S. S. Bhowmick, W. K. Ng, and E.-P. Lim, "Research issues in web data mining," In Proceedings of the Conference on Data Warehousing and Knowledge Discovery, 1999, pp. 303–319.

[3] S. Pal, V. Talwar, and P. Mitra, "Web mining in soft computing framework : Relevance, state of the art and future directions," IEEE Trans. Neural Networks, 13(5), 2002, pp. 1163–1177.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Technical report,Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[5] C. Ridings and M. Shishigin, "Pagerank uncovered," Technical report, 2002.

[6] W. Xing and A. Ghorbani, "Weighted PageRank Algorithm," Proc. of the Second Annual Conference on Communication Networks and Services Research (CNSR '04) IEEE, 2004 , pp. 305-314, 0-7695-2096-0/04 .

[7] J. Kleinberg, "Authoritative Sources in a Hyper-Linked Environment", Journal of the ACM 46(5), 1999, pp. 604-632.

[8] S. Chakrabarti, et al. "Mining the Web's link structure." Computer 32.8 ,1999, pp. 60-67.

[9] A. M. Zareh Bidoki and N. Yazdani, "DistanceRank: An intelligent ranking algorithm for web pages," Information Processing and Management, Vol 44, No. 2, 2008, pp. 877-892.

[10] R.S. Sutton and A.G. Barto, "Reinforcement Learning: An Introduction," Cambridge, MA: MIT Press, 1998.

[11] J. Cho, S. Roy and R. E. Adams, "Page Quality: In search of an unbiased web ranking," Proc. of ACM International Conference on Management of Data,". 2005, pp. 551-562.

[12] X. Wang, T. Tao, J. T. Sun, A. Shakery, and C. Zhai, "DirichletRank: Solving the Zero-One Gap Problem of PageRank," ACM Transaction on Information Systems, Vol. 26, Issue 2, 2008.

[13] A. K. Singh and P. Ravi Kumar. "A Comparative Study of Page Ranking Algorithms for Information Retrieval," International Journal of Electrical and Computer Engineering 4, no. 7 (2009), pp. 469-480.

[14] Z. Gyongyi and H. Garcia-Molina, "Web Spam Taxonomy," First International Workshop on Adversarial Information Retrieval on the Web *(AIRWeb 2005)*, 2005.

[15] Z..Gyongyi and H. Garcia-Molina, "Link Spam Alliances," Proc. of the 31st International Conference on Very Large DataBases (VLDB), 2005, pp. 517-528.

[16] A. Rajaraman, J. Leskovec, and J. D. Ullman, "Mining of Massive Datasets," 2013, pp.161-198.

[17] S. Brin and L. Page, "Anatomy of a large-scale hypertextual web search engine," Proc. 7th Intl. World-Wide-Web Conference, 1998, pp. 107–117.

[18] A. Broder, et al. "Graph structure in the web," *Computer networks* 33.1 , 2000, pp. 309-320.

[19] T.H. Haveliwala, "Topic-sensitive PageRank," Proc. 11th Intl. World-Wide-Web Conference, 2002, pp. 517–526.

[20] J. Qiu and Z. Lin, "A framework for exploring organizational structure in dynamic social networks," Decision Support Systems, 51, 2011, pp.760–771.

+