*Journal of Engineering Sciences, Assiut University, Vol. 36, No.5, pp.1213 -1223, September 2008*

# AN ENHANCEMENT OF PAGERANK ALGORITHM COMPUTATION TIME (PRCT)

***Mahmoud Z. Abdu,***
*Electronics & Communication Dept., Faculty of Engineering,*
*Helwan University, Egypt*
*E-mail: Eng_MahmouZaki@hotmail.com*

***Manal A. Ismail, and***
*Electronics & Communication Dept., Faculty of Engineering,*
*Helwan University, Egypt*
*E-mail: manal_shoman@helwan.edu.eg*

***Mohamed E. El-Adawy***
*Electronics & Communication Dept., Faculty of Engineering,*
*Helwan University, Egypt*

*(Received July 19, 2008 Accepted August 25, 2008)*

*The World Wide Web is growing at phenomenal rates. Millions of result returned from search engines. The rank of pages in the search engines is very important. One of the basic rank algorithms is PageRank algorithm. This paper proposed an enhancement of PageRank algorithm to speed up the computation. The results of the proposed algorithm is speed up the computation time by nearly 22%.*

***KEYWORDS:*** *search engine, ranking algorithms, PageRank algorithm.*

## 1. INTRODUCTION

The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the art of web search engine. World Wide Web search engines have become the most heavily-used online services, with millions of searches performed each day [2]. The web search engines use the structure present in hypertext to provide much higher quality search results [6].

The rank of pages is an integral component of any search engine. In the context of the web search engines the role of ranking becomes even more important. The most important researches proposed **Link Analysis Ranking** [1,3] are Kleinberg [4], and Brin and Page [6].

Link Analysis Ranking can be described as the use of hyperlink structure for the purpose of ranking web documents. Link Analysis Ranking operates on the **graph** representation of hyperlinked web documents. The hyperlink graph is based on representation of a web page as a **node** and hyperlink between pages as a **directed edge**. The goal of Link Analysis Ranking is to extract this information, and use this information to determine weight for every page, and use these weights to rank the web documents. This paper proposes an enhancement to speed up the computation of **PageRank algorithm** [6] by eliminating the redundant computation.

The rest of the paper is organized as follows. Section two presents overview about the Link Analysis Ranking Algorithms. Section three explains the Depth Paths

for PageRank Algorithm (DPPR). Section four presents the experimental data preparation. Section five presents experimental results. Finally, Section six concludes the paper.

## 2. LINK ANALYSIS RANKING ALGORITHMS

All the Link Analysis Ranking algorithms start with a collection of Web pages to be ranked. These algorithms proceed as follows:

- **Extracting**: extracting the hyperlinks between the pages
- **Constructing**: constructing the underlying hyperlink graph. The hyperlink graph is constructed by creating a node for every Web page, and a directed edge for every hyperlink between two pages.
- **Calculated the weight**: The graph is given as input to the Link Analysis Ranking algorithms. The algorithms operate on the graph, and produce a weight for each Web page. This weight is used to rank the pages.

In the following two sections; the **In-degree algorithm [5]** and **PageRank algorithm [6]**, will be briefly described as an example of Link Analysis Ranking algorithms.

### 2.1- The In-degree algorithm

The In-degree algorithm is ranking the pages according to their popularity. The popularity of a page is measured by the number of pages that link to this page. It ranks pages according to their in-degree pages (the number of pages that link to the page). This algorithm was applied by several search engines in the early days of Web search [1].

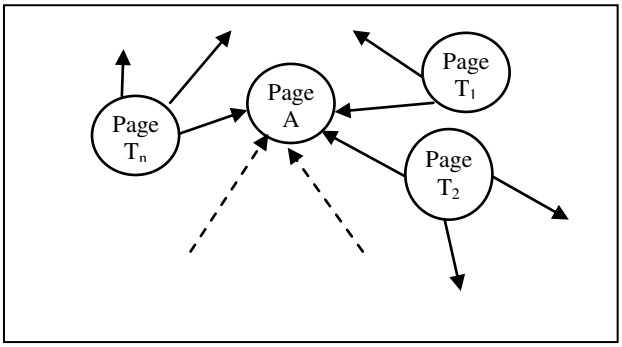


Figure 1: The PageRank graph

### 2.2- The PageRank Algorithm

The algorithm depends on the computation of the PageRank weight of all pages in the graph by equation (1).

The PageRank weight of a page A is given as follows:

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right) \qquad (1)$$

PageRank weight is defined as follows:

- Assume page A has pages $T_1$, T2 ... $T_n$ which point to it (Figure 1).
- The parameter d is a damping factor which can be set between (0, 1) usually set d to 0.85.
- C (T) is defined as the number of links going out of page T.
- PR (A) is the PageRank weight of the page A.

The intuition underlying the *In-degree* algorithm is that a good weight is a page that is pointed to by many nodes in the graph. Brin and Page [6] extended this idea further by observing that not all links carry the same weight. Links from pages of high quality should confer more weight. It is not only important how many pages point to a page, but also what the quality of these pages is. Therefore, they propose a one level weight propagation scheme, where a good weight is one that is pointed to by many good weights. Figure 2 shows the PageRank algorithm.

PageRank Algorithm

Initialize all PageRank weights to 1

Repeat until the PageRank weights are convergence

For every node i

$$PR(i) = (1 - d) + d \sum_{j \in F(i)} \frac{PR(j)}{C(j)}$$

Figure 2: PageRank Algorithm

Due to the huge size of actual web, an approximate iterative computation is usually applied to calculate the PageRank weight. This means that each page is assigned an initial starting value and the PageRanks of all pages are then calculated in several computation circles based on Eq. (1). The minimum PageRank of a page is given by (1 - d); while the maximum PageRank is determined as dN + (1 - d). This maximum can theoretically achieve, only if all Web pages solely link to one page, and this page also solely links to itself [9].

## 3 PAGERANK ALGORITHM COMPUTATION TIME (PRCT)

In the PageRank algorithm many pages converge quickly, while relatively few pages take much longer time to converge [7]. The drawback of the PageRank algorithm is that the PageRank weight of pages that have converged is recomputed at each iteration.

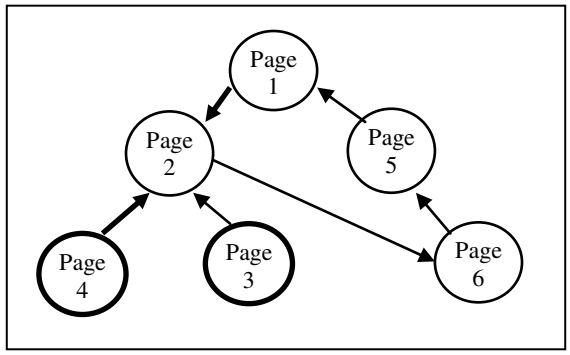


Figure 3: Example of graph G

The computation of the PageRank algorithm can be significantly reduced by eliminating redundant computation. In particular, the algorithm does not need to recompute the PageRank weight of the pages that have already converged. It does not need to recompute the contribution of PageRank weight from pages that have converged to other pages.

The proposed algorithm reduces the number of iteration on the graph by computing the PageRank weight of the most pages at the first iteration. It iterates on the pages that needed only to converge.

The proposed algorithm gives the same top ten pages of the PageRank algorithm. This paper aim to reduce the computation time of the PageRank algorithm by found the reason of much iteration. As described in section 2-2, the number of iterations of the PageRank algorithm depends on the complexity of the graph, and the complexity of the closed loop in this graph. An example of closed loops is shown in Fig. 3.

```
ComputeByIteration (Page)
    In = In_degree [Page]                          // In is an array of in-
    degree pages of page P
    For I = 1 To length (In)
          IP=In [I]                                          // IP is the in-
    degree page number I of page P
          If   visited [IP] = False   then          // if page I not visited
    before
                    ComputeByIteration (In [I])
          End If
    End For
    Calculate the PR [Page]                        //from equation (1)
```

```
PRCT ( )
    N: Number of pages.
    For Page=1 To N
        If   In-degree Number (Page) = 0 then    // if number of in-degree pages of
    Page equal 0
                Calculate the PR [Page]          // from equation (1)
                Computed [Page] = True           // the Page is computed.
        Else
                PR [Page] =1                     // Initialize PageRank weights of
    Page to 1
                Computed [Page] = False          // the Page is not computed yet.
        End If
    End For
    For Page = 1 To N.
        If (computed [Page] =False)
                GetDepthPath (Page)
        End If
    End For
    Sort the PageRank weight in the graph to get top ten pages.
    Print the top ten pages.
End PRCT

GetDepthPath (Page)
    In = In_degree [Page]                    // In is an array of in-degree pages of Page
    For I = 1 To length (In)
        IP=In [I]                            // IP is the in-degree page number I of Page
        If visited [IP] = False then         // if page I not visited before
                GetDepthPath (IP)
        Else if    visited [IP] = True   And    computed [IP] =False    then
            Detect any outer closed loop under page IP
        End If
    End For
    If    all in-degree pages of Page are computed        then
        Calculate the PR [Page]              //from equation (1).
    End If
    If   detect closed loop     then
        Repeat until the PageRank weight of the pages in the closed loop converge at 0.01
        ComputeByIteration (Page)
    End If
End GetDepthPath
```

The proposed algorithm can solve the drawback of the classical PageRank algorithm by:

1- Computing the PageRank weight of the pages out of any closed loop at the first iteration.
2- Isolating the closed loops and iterates on the not converged pages only in this closed loops.
3- In turn, it reduces number of iteration on the pages.

Figure 4 summarize the PRCT algorithm. The algorithm calls a recursive procedure "***GetDepthPath "***. This procedure gets the in-degree pages of every page in the graph to determine the PageRank weight at the first iteration. If any closed loop detected, the algorithm iterates on this closed loop only by using the method "***ComputeByIteration "***.

The advantage of the proposed algorithm:

1- It computes the PageRank weight of most pages at first iteration.
2- It reduces the number of iteration on the pages to compute the PageRank weight.
3- In turn, it reduces the computation time of the algorithm, compared to the PageRank algorithm.

Figure 4: PRCT Algorithm

## 4 EXPERIMENTAL DATA PREPARATION

This section presents a brief description of experimental results of the proposed algorithm. The experimental results data are text files from [8]. These text files contain information about queries used in the experimental phase. This text files represent 33 queries:

"abortion", "affirmative action", "alcohol", "amusement parks", "architecture", "armstrong", "automobile industries", "basketball", "blues", "cheese", "classical guitar", "complexity", "computational complexity", "computational geometry", "death penalty", "genetic", "geometry", "globalization", "gun control", "iraq war", "jaguar", "Jordan", "movies", "national parks", "net censorship", "randomized algorithms", "recipes", "roswell", "search engines", "shakespeare", "table tennis", "vintage cars", "weather".

Each query represented by three text files named: "nodes", "adj_list", and ***"inv_adj_list".***

***The nodes.txt file*** is formatted as follows:

- Number of pages
- Information of each page, each page described as follow :
  - Page ID
  - http address of the page
  - page title
  - number of in-degree pages
  - Number of out-degree pages

***The Adjacency List file:*** Contains list of out-degree pages IDs for each page. An example of a page entry is the following

***1:20 500 6***

This means that the page with ID ***1***, points to the pages with IDs ***20, 500, 6.***

***The Inverted Adjacency List file:*** Contains list of in-degree pages IDs for each page. An example of a page entry is the following

***20:1 5 80 -1***

This means that the page with ID ***20***, is pointed to by the pages with IDs ***1, 5, 80***.
The following are steps form implementation:

- Data preprocessing step.
- Run proposed algorithm.
- Compare the results.

The preprocessing step includes creating array of structure to represent the data. The structure is as follows:

- ***NodeID:*** a unique identifier for the Page
- ***NodeName:*** http address of the page
- ***NodeNodes:*** the title of the page
- ***NodeIndegreeNumber:*** Number of in-degree Pages
- ***NodeOutdegreeNumber:*** Number of out-degree Pages
- ***[ ] NodeIndegree:*** Array of NodeID of in-degree pages
- ***[ ] NodeOutdegree:*** Array of NodeID of out-degree pages

Table 2: Total number of iterations for classical PageRank vs. PRCT in all queries

| **Query** | **Number of iteration** | |
|---|---|---|
| | **PageRank alg.** | **Proposed alg.** |
| abortion | 163660 | 4773 |
| affirmative action | 131196 | 3690 |
| alcohol | 220512 | 5360 |
| amusement parks | 153450 | 4798 |
| architecture | 369950 | 10524 |
| automobile industries | 52624 | 3713 |
| armstrong | 148350 | 3983 |
| basketball | 229862 | 8617 |
| blues | 256992 | 8807 |
| cheese | 150236 | 6654 |
| classical guitar | 113400 | 4287 |
| complexity | 128304 | 6274 |
| computational complexity | 38700 | 1530 |
| computational geometry | 38964 | 2921 |
| death penalty | 171920 | 13508 |
| Genetic | 227814 | 6177 |
| Geometry | 211974 | 5010 |
| globalization | 216700 | 5951 |

| | | |
|---|---|---|
| gun control | 153660 | 4485 |
| iraq war | 158844 | 6749 |
| Jaguar | 141000 | 3892 |
| Jordan | 200450 | 4729 |
| Movies | 294779 | 12494 |
| national parks | 228336 | 6133 |
| net censorship | 124704 | 3571 |
| randomized algorithms | 6678 | 890 |
| Recipes | 272636 | 7453 |
| Roswell | 150660 | 4059 |
| search engines | 524655 | 14670 |
| shakespeare | 219150 | 5905 |
| table tennis | 54544 | 1828 |
| vintage cars | 148780 | 5834 |
| Weather | 376517 | 18329 |

## 5 EXPERIMENTAL RESULTS

The following results of implementation of the proposed algorithm comparing with PageRank algorithm. The comparison based on the following factors:

Table 3: Computation time in seconds in classical PageRank vs. PRCT

| **Query name** | **Computation time (Sec)** | |
|---|---|---|
| | **PageRank alg.** | **Proposed alg.** |
| Abortion | 0.703125 | 0.5625 |
| affirmative action | 0.40625 | 0.25 |
| alcohol | 1.015625 | 0.78125 |
| amusement parks | 0.65625 | 0.625 |
| architecture | 2.578125 | 3.15625 |
| automobile industries | 0.109375 | 0.09375 |
| armstrong | 0.546875 | 0.390625 |
| basketball | 2.0625 | 1.46875 |
| blues | 1.46875 | 1.203125 |
| cheese | 0.5625 | 0.484375 |
| classical guitar | 0.5625 | 0.484375 |
| complexity | 0.796875 | 0.625 |
| computational complexity | 0.09375 | 0.0625 |
| computational geometry | 0.359375 | 0.28125 |
| death penalty | 1.359375 | 0.796875 |
| genetic | 1.484375 | 1.046875 |

| | | |
|---|---|---|
| geometry | 1.140625 | 0.78125 |
| globalization | 1.140625 | 0.890625 |
| gun control | 0.609375 | 0.421875 |
| iraq war | 0.84375 | 0.625 |
| jaguar | 0.484375 | 0.3125 |
| jordan | 0.734375 | 0.5625 |
| movies | 3.453125 | 2.953125 |
| national parks | 1.1875 | 0.78125 |
| net censorship | 0.46875 | 0.375 |
| randomized algorithms | 0.0625 | 0.03125 |
| recipes | 1.3125 | 1.109375 |
| roswell | 0.5 | 0.390625 |
| search engines | 9.734375 | 7.296875 |
| shakespeare | 0.984375 | 0.65625 |
| table tennis | 0.25 | 0.234375 |
| vintage cars | 0.640625 | 0.65625 |
| weather | 3.59375 | 2.84375 |

- Number of iteration for each query.
- Total computation time.
- Number of pages computed at the first iteration.
- Number of iterations of the top ten pages

Table 2 shows the total number of iterations for each query for both proposed algorithm and PageRank algorithm. The average percentage of the number of iteration in all pages in the proposed algorithm vs. PageRank algorithm is 4 %.

Table 3 shows computation time in seconds in the proposed algorithm vs. PageRank algorithm. The average computation time of the proposed algorithm is 0.77 of the computation time of PageRank algorithm. The computation time depends on the complexity of the graph in the query.

Figure 5 shows the percentage of the number of pages calculated at the first iteration to the total number of Pages in each query. The percentage of the number of pages calculated at the first iteration in all queries is over 91 %. The proposed algorithm reduces the number of iterations on the graph.

The most important feature of the proposed algorithm is the reduction of computation time. The main important factor reducing computation time based on isolating of closed loops in the graph. All nodes not in any closed loop calculated at first iteration. The iteration on closed loop to converge is mostly less than iterations on all graph nodes.

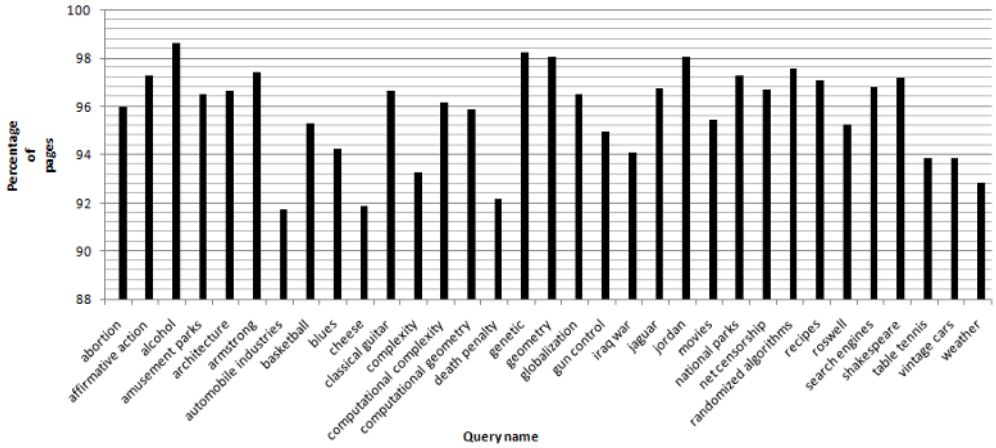


Figure 5: The percentage of the number of pages calculated at the first iteration.

## 6 CONCLUSIONS

This paper presents an algorithm (PRCT) to speed up the computation of the PageRank algorithm. The contributions of the proposed algorithm can be summarized as follows:

1- Compute the PageRank weight of most pages at the first iteration.
2- Reduce the number of iteration for each query.
3- Reduce the total computation time.

The PageRank weights of most pages are computed at first iteration in the proposed algorithm. Any page not in closed loop is computed at the first iteration.

The numbers of iterations for each page to converge are reduced in the proposed algorithm. It iterates on the pages in the closed loop only. The number of iteration on the closed loop should be less than the total number of iterations in the all graph.

The total computation time is reduced in the proposed algorithm. This reduces overhead time to compute the PageRank weights. The average computation time of the proposed algorithm is 77.66723% from the classical PageRank.

## REFERENCES

[1] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link Analysis Ranking Algorithms Theory and Experiments". ACM Transactions on Internet Technologies, ,Vol. 5, No. 1, Pages 231–297, February 2005

[2] B. Davison, "Recognizing nepotistic links on the web". In AAAI-2000 Workshop on Artificial Intelligence for Web Search, Austin Texas, 2000. AAAI Press.

[3] Islam A.Elgawad " Development in web search engine optimization ", Msc thesis, Computers and Information - Helwan university, 2005

[4] J. Kleinberg. "Authoritative sources in a hyperlinked environment". Journal of ACM (JASM), vol. 46, pages 604-632, 1999.

[5] P. Tsaparas (2004), "Link Analysis Ranking", PhD Thesis, University of Toronto, March 2004.
[6] S. Brin and L. Page, "The anatomy of a large-scale hyper textual Web search engine". In Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 1998.
[7] Sepandar Kamvar, Taher Haveliwala, and Gene Golub, "Adaptive Methods for the Computation of PageRank." Technical Report, Stanford University, 2003.
[8] "http://www.cs.toronto.edu/~tsap/experiments/download/download.html"
[9] Nan Ma,Jiancheng Guan,Yi Zhao. "Bringing PageRank to the citation analysis", Information Processing & Management, Volume 44, Issue 2, 2008, Pages 800-810.

## ملخص عربى للورقة العلمية

## تحسين زمن التنفيذ فى خوارزم PageRank

**مقدمة :**

النمو كبير فى مصادر المعلومات على الشبكة الدولية . محركات البحث من أهم المصادر على الشبكة الدولية, لزيادة المعلومات وأعداد نتائج الطلبات التى تصل إلى آلاف بل الملايين . وفى نفس الوقت المستخدم يرى أوائل النتائج فقط وهو ما يهتم به ولذلك ترتيب النتائج هى أهم ما يجعل محرك البحث ناجحا . من أهم الخوارزميات المستخدمة لترتيب النتائج PageRank . تقدم هذه الورقة تحسين فى خوارزم PageRank لتحسين زمن التنفيذ.

**ترتيب وتحليل الوصلات:**

ربط وتحليل الترتيب يعمل على افتراض انه ، بالنظر الى جميع المواقع على الشبكة الدولية يتضمن معلومات مفيدة عن الوصلات المشار إليها . الهدف من الربط والتحليل هوالترتيب . وتستخدم هذه الوصلات المشار إليها القيم لترتيب شبكة الربط وتحليل المواقع .

الترتيب يبدأ بمجموعة من الصفحات على الشبكه لنصل إالى الترتيب في الخوارزم المقترح ، ثم تنتقل الى استخلاص وصلات بين الصفحات ، وبناء عليها توصيله الرسم.يكون شكل الشبكة هو عمل نقطة لكل موقع ونقطة اتصال بين هذا الموقع والمواقع التى تشير إلى هذه الصفحة

**ربط و تحليل خوارزميات الترتيب**

صلة تحليل الترتيب يبدأ بمجموعة من صفحات الشبكه لتكون مرتبة.في الخوارزم ننتقل الى استخلاص وصلات بين الصفحات ، وبناء عليها توصيله الرسم. توصيله الى الشكل لنبنى نقطة لكل صفحة ، ووصلة لكل توصيله بين صفحتين.

الرسم البياني يعطي كمدخل لربط تحليل خوارزميه الترتيب. الخوارزميات تعمل على الرسم البياني ، ووضع وزن لكل صفحة. هذه الاوزان تستخدم لترتيب الصفحات