World Journal of Environmental Biosciences
All Rights Reserved WJES © 2014
Available Online at: **www.environmentaljournals.org**

Volume 6, Supplementary: 46-51

ISSN 2277- 8047

Optimizing Page Ranking Algorithm by Using Collective Intelligence Techniquesin Data Mining

Hamed Ghazanfaripoor

Department of Engineering, Minab Branch, Islamic Azad University, Minab, Iran.

ABSTRACT

When users search on the web, public search engines provide available links to users of many pages as a result, which makes it difficult to assess and find the best information. In order to prevent such these problems and improve Web search operations, web crawling techniques and web page ranking algorithms can be used. In this regard, standard PageRank algorithm is one of the most popular and applicable algorithms, which it acts based on the web graph structure. In this research, we try to make a modified version by changing the standard algorithm that unlike the standard algorithm, the interest of users in web pages is involved in calculating the rank of the pages, which it results in better and more relevant outcomes. In this method, the interest of users is accessible through web server registers and Firefly algorithm is used to improve the algorithm of ranking. In the proposed algorithm, we consider web pages as firefly and the interest rate of users on pages as a factor indicating the amount of page attraction and the amount of cumulative Luciferin. The simulation results show that the proposed algorithm offers better ranks of web pages and generates more distinct ranks. In comparison, the proposed algorithm is improved 40% better than the standard PageRank algorithm and 59% relative to PageRank (VOL).

Keywords: *Web mining, ranking, PageRank algorithm, Firefly community.*

Corresponding author:Hamed Ghazanfaripoor

INTRODUCTION

The need for methods and techniques that can extract useful information is more than ever before by increasing the amount of information and web development. Nowadays, the ability of a site to respond quickly to visitors and their successful guidance to useful information is a key factor in the success of the sites. Therefore, web server activities have changed based on the interests of users in order to increase the efficiency of well-designed websites. The ability to know patterns of habits and interests of the users helps corporate operational strategies.

In order to provide better and more efficient results, numerous ranking algorithms have been used on web pages up to now, including: PageRank algorithms, Weighted PageRank, HITS, etc. In fact, Web page ranking algorithms are an essential component of search engines. Their goal is to provide a rank for every webpage, i.e., it is a measure that predicts how important the page is to be viewed by users. The above algorithms greatly reduce the search.

In fact, achieving these goals is possible by using web mining techniques. Based on the data collected on the World Wide Web, web mining is classified into three categories: web content mining, web structure mining, and web usage mining. Web content mining works by discovering information or knowledge of web server content. Web structure mining discovers the relationships between web pages by analyzing web structure. Based on topology of hyperlinks, the web structure mining groups WebPages and generates the relevant patterns such as similarity and communication between different websites. Web usage mining is the process of discovering what Internet users are looking for. The web usage mining focuses on techniques that can predict user behavior while they interact with the web. Considering the interest of users in ranking the pages and interfere with this interest in the web mining process can achieve better results in the field of sites' efficiency and effective policies in organizations. Since the standard rating algorithms generally do not consider this interest rate, this paper attempts to investigate the impact of this factor and improve the ranking standard algorithm.

Research objectives are: creating an index for users implicitly by tracking their pages viewed by them, improving the performance of web search engines (applicable purpose), selecting the appropriate usage-based web mining method (PageRank algorithm, HITS algorithm, etc.).

As stated by Yang in(Yang, Xin-She,2010), the firefly algorithm was first introduced by Krishnanand and Ghose in 2005 as an optimization algorithm and was presented in the category of Swarm Intelligence, and then was developed by Yang in 2007. Swarm intelligence examines the collective intelligence that comes from the community of intelligent and simple agents. Krishnanand and Ghose(Kolahka and etal,2013) presented the firefly algorithm of this algorithm with inspiration from the behavior of the firefly in nature, which indicates that firefly's worms live in colonies and cooperate in order to survive the colony.

Nasrovy and Pets(Rashidi and etal, 2012) displayed each profile which is a set of pages in a form of the weight of those pages after identifying and meetings and building user profiles. According to Salvan et al. (Selvan and etal, 2012), the PageRank algorithm, which is a set of non-queuing algorithms, is one of the most important factors used by Google in

computing Web page rankings. The PageRank value of a web page depends on the PageRank values of the pages pointing to it and the number of outbound links for those pages. In this algorithm, the page rank depends on the number of links it has. Richardson et al. [44] teach frank which is a nerve-centered RankNet. RankNet is one of the common techniques in learning the ranking functions. Kumar et al. (Kumar G., Duhan N., Sharma A.K,2011) proposed a standard page ranking algorithm based on viewing links (VOL) inspired by the standard PageRank algorithm, which calculates the number of visits to web page input links. Dinkar and Kumar (Dinkar S.K., Kumar H,2012) concluded time factor to determine the rank of each web page. The rank of each webpage is calculated based on each unit time in the improved page ranking algorithm. Thwe(Thwe, P,2013) presented a method for predicting web pages using a ranking algorithm based on similarity and popularity. In this way, a version of the PageRank algorithm is provided to predict access to the next page, which performs a ranking through the expansion of several motion features such as similarity, page size, page access time, etc., and can improve the page through prediction. Yang et al. (Yan and etal, 2011)presented a new algorithm for categorizing web pages based on the standard PageRank algorithm and using the genetic algorithm. The genetic algorithm is a search technique to find the most appropriate and precise solution for optimization and search and is categorized in the category of general search methods. Shomalinasab etal.(Shomalinasab and etal, 2014),used the firefly algorithm to determine the optimal similarity function for the purpose of creating an advisory system. In this research, the similarity of the users is measured in order to categorize them in separate groups by parametric measuring, and then the number K of the nearest neighbor to the active user is considered to be on the item i and the prediction of the rank is expressed based on it. Huang and Zhou(Huang, and Zhou, 2011) provided two new clustering methods based on the firefly algorithm. In the first method, the GSO algorithm is used to analyze data clustering automatically. Senthilnath et al. (Senthilnath and etal, 2011) used the firefly algorithm to cluster the problems.

RESEARCH METHOD

In the proposed method in this paper, the combination of web usage mining and web structure mining techniques was used to calculate web pages PageRank. Preprocessing data is essential and inevitable in the web usage mining. Given the fact that existing profiles are not just for a single user, and different information is maintained for each user, the marketing of information is accompanied by error in most cases. In web structure mining is also being explored the communication between web pages by web structure analysis. The proposed algorithm structure is shown in Fig. 1.

The steps involved in doing the proposed algorithm are as follows:

- Preprocess of the web server registries in order to extract user meetings (Preprocess of web server registries includes: data cleansing, user identification, and meeting detection.)
- Extracting user attributes using their meeting sets
- Construction of meeting vector
- Creating a user profile
- Inspiration of the firefly algorithm to update interest on each page
- Ranking of pages according to the results

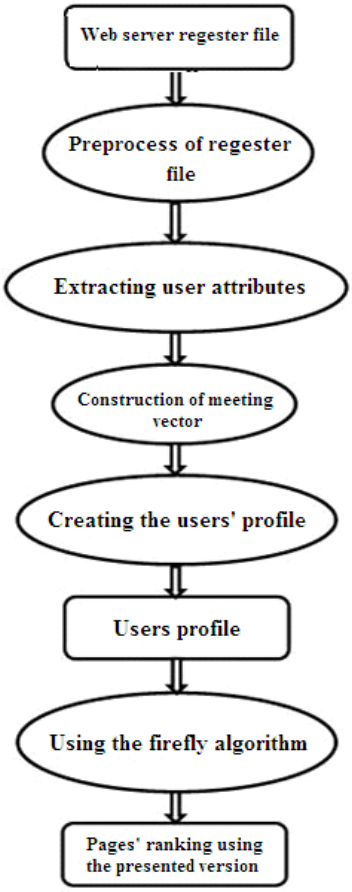

Fig. 1: Structure of the proposed algorithm

The details of each of the system components are explained below:

- **Preprocessing the web server records**

Web servers often store all site user access activities to web server registers. As stated, web servers registers data in a variety of formats according to their configuration parameters, but generally register's files include the same basic information such as: client IP address, requested, URL request time, HTTP status code, referral, etc. In order to perform web mining operations, it is necessary at first to prepare the data and then performs the following basic steps:

- **Data Cleansing**

Since in web-server registration, it is not suitable for all registered entries for use in web mining, it only needs to keep entries that contain relevant information. Therefore, the following data are deleted to remove the inappropriate entries from the registration file:

- Entries related to image files, audio files and other graphics files that are associated with requests for specific pages.
- Registered entries correspond to requests that have not been made, such as requests that are encountered with HTTP errors.
- Registered entries that have answers other than "GET" and "POST".

- **Users' identification**

One way of identifying users is to pay attention to the IP addresses of users in a site. We can assume that each IP belongs to a distinct user. In this case, if some users use the same IPs (such as proxy servers), or if a user has accessed the Web from different systems, we use the referrer information and user agents to distinguish and identify users. This means that the operating system and browser that the user uses is checked. If the two IPs are the same and the user agent is the same, then the two can be considered as one user. In some cases, we may need to look at the structure of links between pages as well as the referrer URL. For example, the user may have requested a page that does not have direct access to any of the pages he has ever seen. In this case, the user can be identified as a new user at the same address.

- **Meetings' identification**

User meeting is a set of pages of a site visited by a user during a specified time. As meetings display the browsing behavior of users, they are very important for pattern discovery. There are different ways to diagnose user meetings, which are divided into two categories of time-based methods and subject-based methods. In this paper, a time-based method is used.

- **Construction of meetings' vector**

A user meeting vector is a set of his transactions which it contains weighted pages that have been viewed over a given time period. In other words, the user meeting can be expressed in terms of the weight of the pages.
Suppose that P is a set of pages accessible by users of a site such that p = {p1, p2, ... pm} and every pi is a page with a unique URL. S is a set of user access meetings defined as s = {s1, s2, ... sm}, in which each Si is a subset of P. Each Si meeting with the m-dimensional vector is represented as Si = {w (p1, Si), (p2, Si), ..., w (Pm, Si)}, which each W (Pj, Si) is the specified weight for the j-th web page viewed at the Si meeting. It should be noted that every webpage Pi can be repeated at the meeting.
The weight of W (Pj, Si) represents the interest of users in a web page. We use the page frequency parameters and the page viewing time to specify this(Kumar G., Duhan N., Sharma A.K,20011). The frequency of the page is defined by the number of times which a web page is met. A user may observe a page several times in a meeting. The larger the number of views a page has in a meeting, the more important the page is at that meeting. The page viewing time is the amount of time spent on a page, which can reflect the importance of the page, because if a page is appealing to the user, he will spend more time to view it and otherwise reject the page and goes to another page. According to(Kolahkaj, M., Harounabadi, A., and Sadeghzadeh, M,2013), equations (1) and (2) are used to calculate the two above-mentioned criteria.

Since the importance of the entire page depends on both of the mentioned parameters, and the interest of a page is high when both parameters are high, according to Rashidiet al.(Rashidi, S.F., Harounabadi, A., AbasiDezfouli, M,2012), the mentioned parameters are used in accordance with equation (3) in order to weigh the pages of the harmonic mean.

- **User indexing**
- This section of the system is about building user profiles and separating meeting vectors from different users. We assume that S1, S2, ... Sk is the set of meetings for the user i (Ui). In this case, the average vector Sui for the user Ui is calculated as the representative, which is actually a representation of the user's favorite pages, and the weight of each webpage in the average vector is obtained from the average of the weight of that web page in all meetings of the user (S1, S2, ..., Sk).

- **Using the firefly algorithm in line with the proposed method**

We first consider each page as a firefly worm and consider the amount of user interest on a page as the amount of luciferin secreted by each worm. Each user browses pages during their meetings, and leaves some interest on those pages, which, as previously described, comes from two screen frequency and page viewing time. As a result, the higher the average interests on a page, the more attractive the page. This amount of charm indicates the amount of Luciferin per worm. As a result, pages that are not much visited will have less Luciferin. Equation (4) shows the relationship of updated Luciferin.
Where and are the Luciferin values of the times of t-1 and t, respectively, p is the values of evaporation of Luciferin (0 <p <1), y is Luciferin replacement rate and J (Xi (t + 1)) is the value of the objective function in the i-th place at the moment of t + 1.

- Extended version of PageRank algorithm

The amount of Luciferin in any given date can be considered as the rank of each page on that date, which directly correlates with the amount of the average interest of users to that page at any given date, and also depends on the rank of that page in the previous history. Finally, the improved equation presented in this article will be in the form of equation (5):
In the above equation, PR (u) (t-1) and PR (u) (t) are respectively the U-rank value at the date of t-1 and the current date t .P is the evaporation rate of Luciferin (0 <P <1) and y is the Luciferin replacement rate. Avr (interest) (t) represents the average of the interests of users who viewed the page at date of t on page u. d displays the probability that a continuous user clicks on the links and (1-d) is a probability that the user jumps to a random page. In fact, d is a moderation factor and is usually set to a value between 0 and 1, (0.85) for the web graph. PR (v) and B (u) are the set of pages that have an input link to the page u. Nv is the page output level v

- **Preprocessing the registration file and users' indexing**

In this article, the set of web server register file data CTI(www.facweb.cs.depaul.edu. Access Time: winter,2016) has been used for two weeks. Using the Microsoft SQL Server software, preprocessing was dealt with on this file. Then we communicated between the required tables and the user meetings were identified with a threshold of 30 minutes(Srivastava and etal, 2000).
After identifying the meetings, the page frequency and the viewing time of the page were computed using the equations (1) and (2), and finally we calculated the importance of the page (user interest to that page) using equation (3). Then we got the weight of each web page in the mean vector, from the average weight of that web page in all user meetings on that date. User profiles were created in this way.

- **Meetings' cleansing**

After identifying user meetings, we need to delete the inappropriate pages. Inspired by Castellano et al. (Castellano and etal, 2011), we removed pages that appeared in less than 10% and more than 80% of the total number of accesses in meetings (such as homepages). Also, all user meetings were deleted with the length of less than three, and 20946 meetings

were remained. Eventually, the number of pages on which the ranking was executed was 359 pages.

3. Data analysis

We will have two database tables for ranking pages. The first table, which obtains after the preprocessing stages, the construction of meeting vectors and creating user profiles, contains pages and meetings information and the percentage of users' interest in the pages.

The second table, expressed as a matrix, indicates the relationship between the pages, all of which are on both the rows and the columns. The values of this table have two values of zero or one, if the two pages are linked together, the value of the field of the crosses in these two pages is 1 and otherwise it is 0. In this way, the set of elements of each column represents the number of output links for that page and the set of elements of each row indicates the number of input links for that page.

In this step, according to the details of the implementation, we will compare the standard PageRank algorithm with the proposed algorithm. The values of y, p, and d are shown in Table 2 inspired by(Huang, Z., Zhou, Y,2011).

Table 2: Default values for performing ranking operations [4]

	y	p	d
Values	1	0.98	0.85

Let's take 1 also for the initial rank rate in repeat. In this way, the results of the simulation of the proposed algorithm and its comparison with the standard PageRank algorithm are as follows:

The Fig. 1 and Fig. 2 show the rankings of 100 pages and all web server pages, respectively; as shown in the figures, the proposed algorithm yields better scores for the ranking.

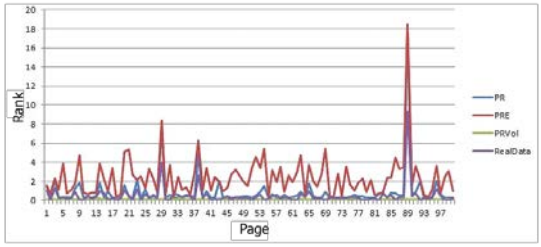

Fig. 1. Comparing 100 pages rankings in the proposed method, PageRank Algorithm and PageRank Algorithm (VOLE)

Fig. 2. Comparing the ranking of all pages in the proposed method, the PageRank Algorithm and the PageRank Algorithm (VOLE)

Figures 3 and 4 show the proposed method error, the PageRank algorithm and the PageRank algorithm (VOLE), respectively. We use the absolute difference between the actual page rank and the rank value in each of the methods to calculate the page ranking error according to equation (6).

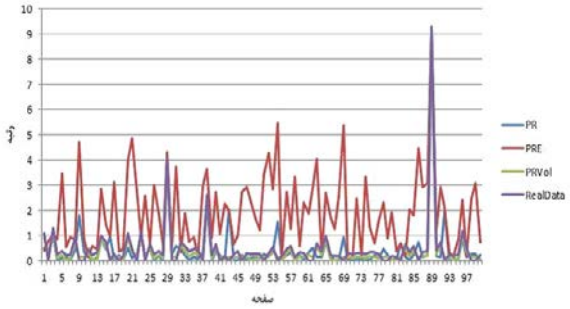

Fig. 3. Comparison of 100 pages' error in proposed algorithm method, PageRank Algorithm and PageRank Algorithm (VOLE)

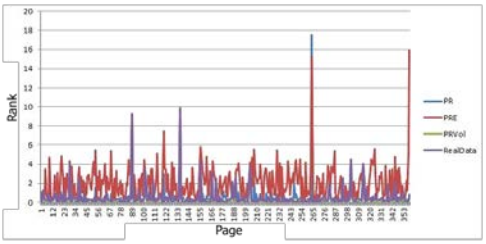

Fig. 4. Comparing the error of all pages in the proposed method, the PageRank Algorithm and the PageRank Algorithm (VOLE)

Figure 5 shows the comparison of 20 pages' ranking in the proposed method, the PageRank algorithm and the PageRank algorithm (VOL) for ease of observing the changes.

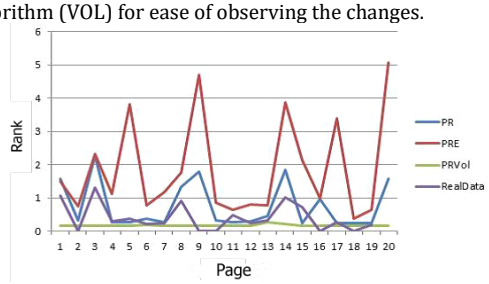

Fig. 5. Comparison of 20 pages' ranking in the proposed method, PageRank Algorithm and PageRank Algorithm (VOL)

In Table 3, the results of the ranking of 10 first pages are presented in all three methods. As you can see, since the users' interest is involved in calculating page rankings in the proposed method, the rank values are higher.

Table 3: Ranking results of 10 pages in the proposed method, PageRank Algorithm and PageRank (VOL) Algorithm

	Real data	PageRank	PageRank(VOL)	Proposed method (PRE)
1	1.071584	1.5668	0.15135	1.494
2	0.214022	0.32635	0.15815	0.75825
3	0.00544	2.2591	0.15036	2.3331
4	0.280337	0.25321	0.15012	1.1252
5	0.00866	0.26235	0.15032	3.8268
6	0.216796	0.38096	0.17489	0.77176

7	0.238818	0.26084	0.15028	1.1655
8	0.909611	1.3233	0.15266	1.7617
9	1.051715	1.7987	0.151125	4.6921
10	0.283714	0.32759	0.15205	0.85413

Table 4 shows the rankings in 20 random pages, the proposed algorithm has produced more distinct ranks for several distinct pages than two other algorithms. In fact, as it is observed, the proposed algorithm produced 19 distinct ranks among the 20 pages, the standard PageRank algorithm produced 2 distinct ranks and the PageRank (VOL) algorithm only produced 6 distinct ranks. In Table 5, the results of the ranking of 359 pages are observed. In the proposed method, 357 distinct rankings were generated for pages, and 277 and 213 unique rankings, respectively, were generated in the standard PageRank and PageRank (VOL) algorithms.

Table 4: Ranking results in 20 random pages

Pages	PR	PR (VOL)	PRE
15	0.24226	0.15006	2.1441
32	0.24226	0.15012	0.36812
42	0.24226	0.1505	204689
79	0.24226	0.15025	0.86859
116	0.24226	0.15001	0.36812
126	0.24226	0.1505	1.5226
129	0.24226	0.1505	0.6658
137	0.24226	0.15007	1.1281
141	0.24226	0.15025	1.0641
150	0.24226	0.1505	0.68741
161	0.24226	0.1505	2.4129
215	0.24226	0.1505	2.3071
253	0.24226	0.1505	4.9619
256	0.24226	0.1505	0.37269
259	0.24226	0.1505	0.80013
272	0.24226	0.1505	0.46658
273	0.24226	0.1505	2.4014
282	0.24226	0.1505	0.53091
19	0.24248	0.1505	0.64404
170	0.24248	0.1505	2.8533

According to [17], when several pages have the same rank, none have any superiority over other pages. For example, among 15 pages, if an algorithm can find 15 distinct ranks for these pages, it is more efficient than an algorithm that considers 10 distinct pages. In fact, this means that the second algorithm does not differentiate between different pages and identifies less important and relevant pages. Therefore, there are unique and distinct results from useful features for ranking.

Table 5: Results of ranking of all pages

Algorithm	The number of unique rank
PageRank	277
PageRank (VOL)	213
Proposed method (PRE)	357

Based on the above, the proposed algorithm has improved in comparison with the standard PageRank algorithm by 40% and compared to the PageRank algorithm (VOL) by 59%.

CONCLUSIONS AND SUGGESTIONS

In this paper, inspired by the Firefly algorithm, a extended version of the PageRank algorithm is presented that combines the usage mining and web structure mining. Initially, we carried out preprocessing operations as recordable (including clearing of data, identifying users, and identifying meetings), and then designed meetings and created profiles for users. We then used Luciferin update section inspired by the Firefly algorithm to update web page ranking. In the next step, we ranked pages using our extended version of PageRank. As it can be seen, the relationship provided in this paper gives better outcomes from ranking by affecting the users' average interest rate to a page. It also shows that in addition to the effective structural parameters on page rank, the rank of each page per date depends on its attractiveness for users on that date.

The results show that the proposed method generates more unique ranks than standard PageRank and PageRank (VOL) algorithms, which leads to the proposal of more essential and relevant pages and easier access for users to pages. For future research, other features, which are extracted from the register file, can be used such as page viewing history, page access sequences to determine the interest of users.

REFERENCES

1. Castellano, G., Fanelli, A.M., Torsello, M.A. 2011. "Newer: a system for neuro-fuzzy web recommendation". Applied Soft Computing vol. 11, Issue 1, pp 793-806.
2. Dinkar S.K., Kumar H. 2012. "Interaction information retrieval and improved PageRank algorithm based on access duration of page". International Journal of Engineering Research & Technology (IJERT vol.1, Issue 6, pp 1-5.
3. Huang, Z., Zhou, Y. 2011. "Using Glowworm Swarm Optimization Algorithm for Clustering Analysis". Journal of Convergence Information Technology, Vol.6, No.2. pp 78-85.
4. Kolahkaj, M., Harounabadi, A., and Sadeghzadeh, M. 2013. "Providing a method to personalize the web using the neural network." The First National Conference on Modern Approaches to Computer Engineering and Information Marketing in Iran. 1-5. Islamic Azad University of Rudsar and Amlash.
5. Krishnanand, K, N., Ghose, D. 2009. "Glowworm Swarm Optimization for Searching Higher Dimensional Spaces". Innovations in Swarm Intelligence, SCI 248, pp 61-75.
6. Kumar G., Duhan N., Sharma A.K. 2011. "Page ranking based on number of visits of links of web page". International Conference on Computer & Communication Technology (ICCCT), IEEE. Allahabad, India, Sep 15-17, pp 11-14.
7. Rashidi, S.F., Harounabadi, A., AbasiDezfouli, M. 2012. "Prediction of users future requests using neural network". Management Science Letters. vol.2, Issue 6, pp.2119-2124.
8. Richardson M., Prakash A., Brill E. 2006. " Beyond PageRank: Machine learning for static ranking". Proceedings of the 15th International Conference on World Wide Web. New York, USA, pp 707-715.
9. Selvan, M, P., Sekar, A, C., Dharshin, A, P. 2012. "Survey on Web Page Ranking Algorithms". International Journal of computer Applications. Vol.41, No.19. pp 1-7.
10. Senthilnath,J., Omkar,S,N., Mani,V. 2011. "Clustering using firefly algorithm:

Performance study".Swarm and Evolutionary Computation 1. Elsevier. pp.164-171.
11. Shomalinasab.F., Sadeghzadeh, M., Esmaeilpour,M. 2014."An Optimal Similarity Measure for Collaborative Filtering Using Firefly Algorithm". Journal of Advances in Computer Research, Vol.5, No.3. pp 101-111.
12. Srivastava,J., Cooley,R., Deshpande, M., Tan,P,N. 2000."Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data ". University of Minnesota. vol.1, Issue 2, pp 12-23.
13. Thwe, P. 2013. " Proposed approach for web page access prediction using popularity and similarity based PageRank algorithm". International Journal of Scientific & Technology Research(IJSTR), vol.2, no.3, pp 240-246.
14. www.facweb.cs.depaul.edu. Access Time: winter 2016.
15. Yan ,L., Gui,Z., Du,W., Guo,Q. 2011." An Improved PageRank Method based on Genetic Algorithm for Web Search". Advanced in Control Engineering and Information Science(CEIS). pp 2983-2987.
16. Yang, Xin-She. 2010. Engineering Optimization: An Introduction with Metaheuristic Applications. Chapter 17. John Wiley & Song Publishing.