TELKOMNIKA, Vol.9, No.3, December 2011, pp. 403~410
ISSN: 1693-6930
accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010

Focused Crawler Optimization Using Genetic Algorithm

Banu Wirawan Yohanes[1], Handoko[2], Hartanto Kusuma Wardana[3]
Faculty of Electronic and Computer Engineering, Universitas Kristen Satya Wacana, Salatiga
Jl. Diponegoro 52-60 Salatiga 50711 Central Java, Phone/Fax.: +62 298 311884
E-mail: bona_yo@yahoo.co.id[1], handoko@staff.uksw.edu[2], hkwardana@yahoo.com[3]

Abstrak

Selama ukuran dari Web terus berkembang, pencarian informasi yang berguna pada Web telah menjadi semakin sulit. Focused crawler bertujuan untuk menelusuri Web dengan menyesuaikan kepada sebuah topik tertentu. Makalah ini mendiskusikan permasalahan yang disebabkan oleh algoritma pencarian lokal. Crawler dapat terjebak di dalam sejumlah komunitas Web yang terbatas dan mengabaikan halaman Web yang relevan di luar jalur penelusurannya. Sebuah algoritma genetik sebagai algoritma pencarian global dimodifikasi untuk mengatasi permasalahan tersebut. Algoritma genetik digunakan untuk mengoptimalkan penelusuran pada Web dan memilih halaman Web yang lebih sesuai untuk diunduh oleh crawler. Beberapa percobaan evaluasi diselenggarakan untuk memeriksa efektifitas dari pendekatan yang diajukan pada makalah. Peneluran crawler menghasilkan koleksi berisi 3396 halaman Web dari 5390 link yang ditelusuri, atau tingkat penyaringan seleksi Roda-Roulette sebesar 63% dan tingkat keakuratan 93% pada 5 kategori yang berbeda. Hasil tersebut menunjukkan bahwa penggunaan algoritma genetik telah memampukan focused crawler untuk menelusuri Web secara komprehensif, meskipun koleksinya relatif kecil. Selanjutnya, penelitian ini membawa potensi yang besar untuk membangun koleksi yang lebih baik dibandingkan dengan metode focused crawling tradisional.

Keywords*: focused crawler, algoritma genetik, pencarian pada Web.*

Abstract

As the size of the Web continues to grow, searching it for useful information has become more difficult. Focused crawler intends to explore the Web conform to a specific topic. This paper discusses the problems caused by local searching algorithms. Crawler can be trapped within a limited Web community and overlook suitable Web pages outside its track. A genetic algorithm as a global searching algorithm is modified to address the problems. The genetic algorithm is used to optimize Web crawling and to select more suitable Web pages to be fetched by the crawler. Several evaluation experiments are conducted to examine the effectiveness of the approach. The crawler delivers collections consist of 3396 Web pages from 5390 links which had been visited, or filtering rate of Roulette-Wheel selection at 63% and precision level at 93% in 5 different categories. The result showed that the utilization of genetic algorithm had empowered focused crawler to traverse the Web comprehensively, despite it relatively small collections. Furthermore, it brought up a great potential for building an exemplary collections compared to traditional focused crawling methods.

Keywords*: focused crawler, genetic algorithm, Web searching.*

1. Introduction

Nowadays the Web becomes a huge information source, which has attracted many people from all over the world. For a Web crawler, one of the most important parts of search engines, searching through so many documents to select the compatible ones is a tedious task. Moreover, the Web, which contains more than 11 million pages still keeps growing and changing rapidly.

Focused crawler [1] is used to selectively collect smaller Web pages collections according to a particular topic with high precision. A focused crawler will try to predict whether a target URL is pointing to a relevant Web page before actually fetching it. Focused crawlers rely on two kinds of algorithm to keep the crawling process on the track. First, Web analysis algorithm will evaluate the quality and relevance of Web pages pointed by target URLs. Second, Web searching algorithm will determine the optimal order in which the targets URLs are visited.

Later Web analysis algorithms can be categorized into two types: the content-based algorithms which analyze the actual HTML content of a Web page to obtain information about

Received April 18th, 2011; Revised June 11th, 2011; Accepted June 27th, 2011

the page itself and the link-based algorithms that represent a considerable amount of latent human annotation and offers some important information for analyzing the relevance and quality of Web pages [2]. For example, the content-based algorithms extract keywords or phrases from the body text using document-indexing techniques to determine a page's relevance. Web page can be considered as standard document which is already known as the specific domain using the vector space model [3]. The vector space model has been employed in many existing focused crawlers [4], [5]. Whereas in the link-based algorithms, Web pages consisting of more incoming links. They are considered to be more important than the other. This is similar to the citation analysis in which frequently cited articles are reputedly to be more significant. The most notoriously link-based Web analysis algorithms include Page Rank [6] and HITS [7].

Many different Web searching algorithms had been examined in focused crawling. Among them are the breadth-first search [8] and the best-first search [1], [4-5], [9], the two most popular searches. The other more advanced searching algorithms such as spreading activation [2] and genetic algorithm (GA) [10] had been proposed as well.

Several problems emerged from traditional focused crawler design, notably the ones caused by using local Web searching algorithms. The local searching algorithms traversed the search space by visiting neighbors of previously visited nodes. Hence, they could find only suitable pages within a limited sub-graph of the Web nearby the starting URLs. This problem is usually assumed as being trapped in local optimal. It became more obvious after the previous Web structural studies revealed the existence of Web communities [7], [11], [12].

Researchers found three structural properties of Web communities that made local searching algorithms were not suitable for focused crawling. First, instead of directly linked to each other, many pages connected to each other through co-citation relationships [11], [13]. Those Web pages could be missed by focused crawlers, as illustrated in Figure 1a. Second, relevant pages within the same domain could be separated into different Web communities by using irrelevant pages [14], as described in Figure 1b. Third, sometimes links could be laid between two pages of different compatible Web communities, but these links usually only pointed from one community to the other with none of them pointing back in reverse direction [13]. This is shown in Figure 1b.

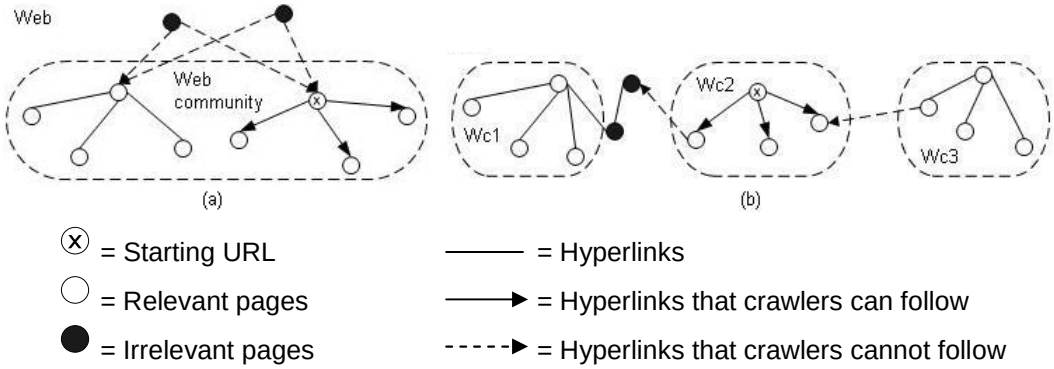

Figure 1. Problems caused by local searching algorithms:
(a) Crawler could miss pages which connected to each other through co-citation relationship.
(b) Crawler was trapped within the initial community.

To alleviate the problems of local searching algorithms, researchers have suggested several strategies. One of them is by using more starting URLs. However, composing a list of high-quality starting URLs is an expensive and time-consuming task. Bergmark [14] proposed to use tunneling method to address the problems. Even though it can find more suitable pages than those without tunneling, it does not change the local searching nature of focused crawling. Overlapping between searching indices of major search engine is not significant. Furthermore, according to Lawrence and Giles [15], the combined top results from multiple search engines had high coverage over the Web. As a potential solution, the meta-searching through multiple search engines is integrated into crawling process.

GA is going to across some vast search spaces efficiently and it can discover the approximate global optimal solutions instead of the local ones. Chen et al. [10] did an experiment using GA to build a personal search agent. Their results showed that GA could effectively prevent the search agent from being trapped in local optimal, and then it would significantly improve the quality of search results. Because of close resemblance features between a personal search agent and a focused crawler, GA is proposed to optimize Web searching in focused crawler.

2. Proposed Genetic Algorithm

In this paper, GA is used to improve the quality of searching results in focused crawling. GA is an adaptive and heuristic method for solving optimization and searching for problems. GA exploits several techniques inspired by biological evolution such as inheritance, selection, cross-over, and mutation. GA is a member of evolutionary algorithms which is included to the rapidly growing area of Artificial Intelligence.

Because it is hard to represent Web pages in bit strings and other conventional genetic operators cannot directly be applied in the Web context, a focused crawler is designed based on the previous study by Chen et al. [10]. The flowchart of the GA-crawler is shown in Figure 2. Although GA-crawler does not add new terms like the Gcrawler [16] and the MultiCrawler Agent (MCA) [17] do, it is expected to maintain a good tracking throughout Web links. In different field, a multi-objective GA for generator contribution based congestion management was proposed by Sen et al. [18]. The algorithm optimizes both real and reactive losses using optimal power flow model. In the many applications GA have successfully implemented [19-20]. The GAs are often modified to solve some specific problems.

Step 1. Initialization

The first phase is to set up several parameters of GA such as population size, generation size, cross-over rate or probability of cross-over, and mutation rate or probability of mutation. Starting URLs and Web pages as the lexicon also becomes an input for the crawler. After all initial parameters are determined Web pages which are pointed by the starting URLs are fetched back by the crawler and saved in a set called generation.

Step 2. Selection based on content-based Web analysis

Jaccard's similarity function used as the fitness function of the GA-crawler is utilized to calculate fitness value of a Web page, which represents similarity between a page and the lexicon on specific domain. The higher the fitness value the more similar a page to domain lexicon. Consecutively, it becomes more compatible to the target domain. The Jaccard's score based on both links and keywords analysis is combined.

Jaccard's function based on links is a ratio of the number of intersection links and union links between two Web pages. The more number of common links that both Web pages have, the higher Jaccard's score of a Web page compared to the domain lexicon will be.

$$J_{link}(A,B) = \frac{\#(X \cap Y)}{\#(X \cup Y)} \quad (1)$$

A represents a domain lexicon and B represents every Web page that has been visited by crawler. X is a set of links within page A and Y is a set of links within page B. $\#(S)$ denotes cardinality of set S and $J(A,B)$ represents Jaccard's score based on links of a Web page compared to the domain lexicon.

Jaccard's function based on keywords is calculated using Term frequency-Inverse document frequency method (Tf-Idf). The weighted term of keywords j in a Web page i, called d_{ij} is evaluated as follows:

$$d_{ij} = tf_{ij} \times log\left(\frac{N}{df_j} \times w_j\right) \quad (2)$$

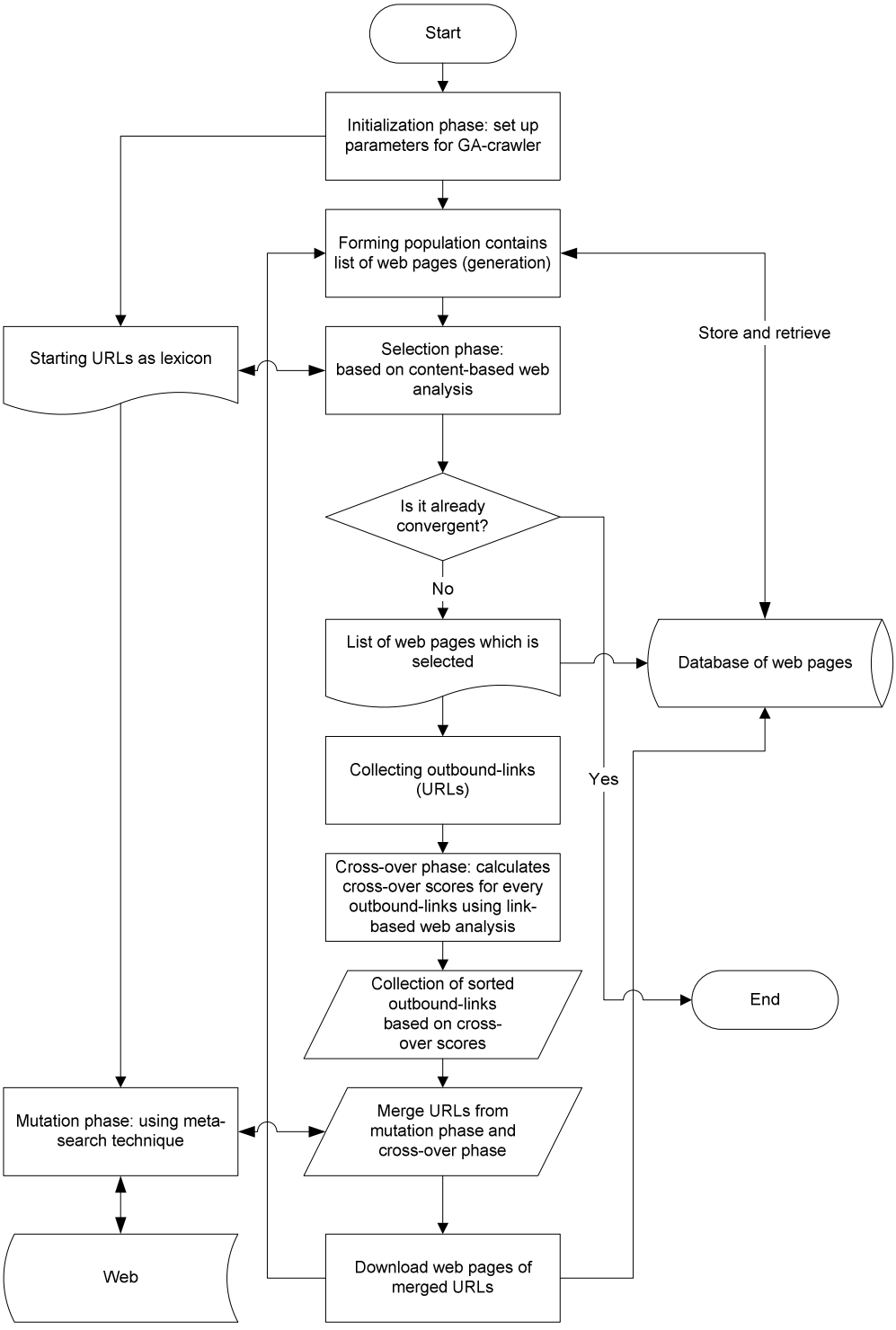

Figure 2. Flowchart of the genetic algorithm crawler

Where tf_{ij} is the number of keywords' appearance, j, in a Web page i. df_j is the number of Web pages in a collection N, where keywords j is found. w_j is a number of words from keywords j. N is a number of Web pages that have been visited by the crawler. A similar Tf-Idf method had been used by Ghozia et al. [21] to estimate the textual similarity of Web pages.

The Jaccard's score based on keywords for each Web page compared to the domain lexicon is calculated as follows:

$$J_{keyword}(A,B) = \frac{(d_{Aj} \times d_{Bj})}{d^2{}_{Aj} + d^2{}_{Bj} - (d_{Aj} \times d_{Bj})} \quad (3)$$

The more often keywords appear in a document and the rarer documents contain the keywords. Therefore, the Jaccard's score of a Web page based on keywords will be better.

Finally, Jaccard's score, $J(A,B)$, for each page that has been visited by the crawler is an average between $J_{link}(A,B)$ and $J_{keyword}(A,B)$. It is presented as follows:

$$J(A,B) = \left(0.5 \times J_{link}(A,B)\right) + \left(0.5 \times J_{keyword}(A,B)\right) \quad (4)$$

After the fitness values denoted by Jaccard's scores in the current generation are calculated, pages with better fitness values are stochastically selected by a random number generator. A higher fitness value will give a page more likelihood to survive in this selection phase. The survived pages are stored locally and the rest of pages are discarded as they are irrelevant. The survived pages are eligible to form a new population for the next generation.

Afterwards, the crawler checks whether the search has reached a converging point or not. If it has, then the crawler will stop searching. On the other hand, the crawler will continue to examine the Web until it reaches any convergent point, which has been specified at the beginning of the crawling process. The criteria of a converged point are the number of pages in local repository, which has reached a pre-set limit or the improvement of fitness values of all pages in a generation below a threshold value or the iteration through generation has reached a pre-set counter.

Step 3. Cross-over based on link-based Web analysis

All outbound-links (out-going URLs) in the survived pages are extracted, and then a cross-over operation is performed to select the most promising URLs. For each outbound-link, μ, the cross-over score is calculated as follows;

$$Cross - over(\mu) = \sum_{W} fitness(P) \quad (5)$$

W is every page P which contains URL μ.

The URLs are sorted according to their cross-over scores and put into the crawling queue. Similar to the Page Rank [6], the cross-over operation favors URLs that have been cited by more high-quality pages with much less computationally cost. In general, cross-over operator supports exploitation of promising local links and it is similar to the best-first search process.

Step 4. Mutation based on meta-search

It is aimed at giving the crawler ability to explore multiple suitable Web communities comprehensively. Random keywords are extracted from the lexicon, which describes starting URLs. The selected keywords run as query for three well-known search engines, Google online at http://www.google.com, MSN online at http://www.bing.com, and Yahoo online at http://search.yahoo.com. GA-crawler did not expand initial keywords [16-17], but it could only change the keywords' composition based on the probability of mutation. The crawler is prevented to explore broader search spaces for improving the crawling rate. Top results from those search engines are combined to build a crawler's queue alongside with URLs from cross-over phase. Given a fact that search indexes of different major search engines have little overlap and their combination covers a very large portion of the Web, it is likely that mutation operator adds diverse URLs from many different and relevant Web communities. Furthermore, as major search engines often include highly co-cited URLs in their search results, mutation phase can make the exploration of individual relevant communities more extensive by adding those co-cited URLs into the collection.

Compared to previously suggested approach as using more starting URLs, the proposed GA approach has numerous advantages. For instance, a list of domain-specific queries is required, that is much easier to compose than a list of high-quality starting URLs. Moreover, the list of queries can be updated by adding frequently used queries found in search engines' search log. This will not only make the collection building process easier, but also allow the final collection to address users' information needs more effectively.

The proposed approach also shows advantages over tunneling method. As a global searching algorithm, GA allows crawler to find new compatible Web communities without any distance limit and it does not introduce noise into the pages collections.

3. Research Method

The research is specialized to investigate how different crawling algorithms and heuristics can be applied so that crawler can retrieve suitable information from the Web more effectively. In order to reach the objective, an application called GA-crawler was written in C# using MS VS C# 8.0 compiler, to search and collect Web pages from the Internet. Hence, the software must be connected to the Internet through any Internet Service Provider (ISP).

GA-Crawler automatically searches for Web pages with relatively high Jaccard's scores which means more similar to the lexicon. Web pages that had been visited during exploration of the Web will build a population for GA's selection. They are kept in a queue called frontiers. Later Web pages which had been chosen on roulette-wheel selection would be downloaded to local repository and saved in database. Finally the GA-crawler would generate an HTML reporting page when it has finished crawling the Web.

Interesting feature that has been added to the crawler is that it would be able to download single distinct Web pages resources such as images, scripts, and cascading style sheets (CSSs) for each domain. GA-Crawler differentiates resources using its type and name. It will deliver a fairly large local resources saving.

To evaluate the effectiveness of the proposed collection building method, some benchmarking experiments were conducted. A conventional best-first search (BFS) crawler could be made by disabling the mutation phase in the GA-crawler. Two small collections consist of about 3,300 Web pages in each collection were built by two crawlers using identical settings. Later the performance of those crawlers were compared each other.

4. Results and Analysis

The numbers of collections in different categories were compared. The collections built from the traditional focused crawler contained about 3,000 nodes (Web page). They were divided into five different categories. While the collections built from GA-crawler contained about 3,300 nodes were also divided into five different categories. Those additional Web pages had been derived by mutation phase in GA. The GA-crawler was expected to visit more compatible Web communities than the best-first search crawler and traditional focused crawler.

Some indicators of the crawler's performance are Web searching precision or the Web pages' relevance which is collected by the crawlers, Web crawling's scope, speed and robustness, and also the total number of resources used in crawling the Web. Several experiments were conducted using different starting URL and keywords for each category. They were taken place on an Intel Dual Core CPU T4200 running at 2.0 GHz, 1 GB RAM and an enhanced 3G network or 3.5G network called High Speed Download Packet Access (HSDPA) which supports down-link speeds up to 3.6 Mbps for the Internet connection. The first 100 Web pages within each category are examined to calculate the precision of the crawling process. The crawler's precision was measured by checking the Web page's relevance compared to the starting URL and category or keyword which was presented.

Table 1 depicts the precision of Web crawling between those two crawlers in five different categories using some keywords. Although the GA-crawler could achieve higher precision of Web crawling than the BFS one, it must be reckoned that GA-crawler's filtering rate to select Web pages is quite big, at about 63%. In other words, GA-crawler retrieved only 63 Web pages for every 100 links it founded. It could have missed some suitable Web pages on the road. Even the size of the collections was only about 3,300 Web pages. It was too small to make decent analysis and conclusion.

Table 1. The precision of the Web crawlers

Category	BFS crawler's precision	GA-crawler's precision
Education	90%	97%
Computer	85%	97%
Digital	80%	82%
Analog	63%	93%
Sport	90%	95%

The speed of the two crawlers to process a Web page was also evaluated. GA-crawler's average crawling rate at 19-103 seconds per page was less than the BFS crawler at 10-40 seconds on the same Internet connection. It was because GA-crawler needs more time to accomplish the selection, cross-over, and mutation methods in order to look for better links.

Due to lack of statistical analysis and the small size of the collections, the hypothesis was not fully supported by the experiments' results. GA-crawler relatively has a better chance to visit more suitable Web communities than the BFS crawler and traditional focused crawler. In general, the research obtained some promising results from the benchmark study.

5. Conclusion

It is important to build high-quality domain specific search engines, as the size of the Web keeps growing. This research had proposed a crawling technique to form domain-specific collections, which serve search engines that incorporate GA as a global searching algorithm into the crawling process. With the effective combination of content-based and link-based Web analysis, together with the ability to perform global searching, the proposed technique has a considerable potential to address many problems that had plagued previous focused crawling methods.

The result showed that the GA-crawler could traverse the Web search space more comprehensively than traditional focused crawler. More experiments on larger scales are required for further study the performance of different Web searching algorithms.

References

[1] Chakrabarti S, van den Berg M, Dom B. *Focused Crawling: a New Approach to Topic-Specific Web Resource Discovery*. Proceedings of the 8th International WWW Conference. Toronto, Canada. 1999: 545-562.
[2] Chau M, Chen H. Comparison of Three Vertical Search Spiders. *IEEE Computer*. 2003; 36(5): 56-62.
[3] Salton G. Another Look at Automatic Text-retrieval Systems. *Communications of the ACM*. 1986; 29(7): 648-656.
[4] Bergmark D. Collection Synthesis. *Proceedings of JCDL 2002*. Portland, Oregon, USA. 2002.
[5] Kitsuregawa M, Toyoda M, Pramudiono I. Web Community Mining and Web Log Mining: Comodity Cluster Based Execution. *Proceedings of the 13th Australasian Database Conference*. Melbourne, Australia. 2002.
[6] Brin S, Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*. 1998; 30: 1-7.
[7] Kleinberg JM. *Authoritative Sources in a Hyperlinked Environment*. Proceedings of the ACM-SIAM Symposium on Discrete Algorithms. San Francisco, California, USA. 1998: 668-677.
[8] Flake GW, Lawrence S, lee Giles C. Efficient Identification of Web Communities. *Proceedings of the 6th ACM SIGKDD*. Boston, Massachusetts, USA. 2000.
[9] McCallum A, Nigam K, Rennie J, Seymore K. A Machine Learning Approach to Building Domain-Specific Search Engines. *Proceedings the International Joint Conference on Artificial Intelligence (IJCAI-99)*. 1999: 662-667.
[10] Chen H, Chung Y, Ramsey M, Yang C. A Smart Itsy-Bitsy Spider for the Web. *JASIS*. 1998; 49(7):604-618.
[11] Dean J, Henzinger MR. Finding Related Pages in the World Wide Web. *Proceedings of the 8th International WWW Conference*. Toronto, Canada. 1999.
[12] Gibson D, Kleinberg J, Raghavan P. Inferring Web Communities from Link Topology. *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*. Pittsburgh, Pennsylvania, USA. 1998.
[13] Toyoda M, Kitsuregawa M. Creating a Web Community Chart for Navigating Related Communities. *Proceedings of ACM Conference on Hypertext and Hypermedia*. Århus, Denmark. 2001.

[14] Bergmark D, Lagoze C, Sbityakov A. Focused Crawls, Tunneling, and Digital Libraries. *Proceedings of the 6th ECDL*. Rome, Italy. 2002.
[15] Lawrence S, lee Giles C. Searching the World Wide Web. *Science*. 1998; 280(5360): 98.
[16] Shokouhi M, Chubak P, Raeesy Z. Enhancing Focused Crawling with Genetic Algorithms. *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*. 2005; 2: 503-508.
[17] Ibrahim SNA, Selamat A, Selamat MdH. Scalable E-business Social Network Using MultiCrawler Agent. *Proceedings of the International Conference on Computer and Communication Engineering*. Kuala Lumpur, Malaysia. 2008.
[18] Sen S, Roy P, Chakrabarti A, Sengupta S. Generator Contribution Based Congestion Management Using Multiobjective Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(1): 1-8.
[19] Bhaskar MM, Benerji M, Sydulu M. A Hybrid Genetic Algorithm Approach for Optimal Power Flow. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(2): 211-216.
[20] Tahami M, Nademi H, Rezaei M. Maximum Torque per Ampere Control of PMSM using Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(2): 237-244.
[21] Ghozia A, Sorour H, Aboshosha A. Improved Focused Crawling Using Bayesian Object Based Approach. *25th National Radio Science Conference (NRSC 2008)*. Egypt. 2008.