A Study on Competent Crawling Algorithm (CCA) for Web Search to Enhance Efficiency of Information Retrieval

S. Saranya, B.S.E. Zoraida and P. Victor Paul

Abstract Today's Web is very huge and evolving continually in dynamic nature. Search engines are the interface to retrieve information from huge repository of the World Wide Web. Due to the difficulty in accessing the information from massive storage of Web, search engines depend on the crawlers for locating and retrieving relevant Web pages. A Web crawler is a software system, which systematically finds and retrieves Web pages from the Web documents. Crawlers use many Web search algorithms for retrieving Web pages. This paper proposes a competent Web search crawling algorithm, which is derived from page rank and BFS Web search algorithm to enhance the efficiency of the relevant information search. In this paper, an attempt has been made to study and examine the work nature of crawlers and crawling algorithms in search engines for efficient information retrieval.

Keywords Web search crawling algorithm · BFS Web search algorithm · Web crawler · URL address · CCA

1 Introduction

Search engines are the software tools widely used by people to retrieve information from Internet [1]. Web crawling algorithm is the process by which a search engine gathers pages from the Web crawler to index them and support a search engine.

S. Saranya (✉) · B.S.E. Zoraida
Department of Computer Science and Engineering, Bharathidasan University, Trichy 620023, India
e-mail: saran.aamec@gmail.com

B.S.E. Zoraida
e-mail: chayav88@gmail.com

P. Victor Paul
Teknuance Info Solutions, Chennai 600099, India
e-mail: victorpaul.ind@gamil.com

© Springer India 2015

L.P. Suresh et al. (eds.), *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, Advances in Intelligent Systems and Computing 325,
DOI 10.1007/978-81-322-2135-7_2

The search engines use Web crawlers to provide up-to-date data. The crawler is an important component of a Web search engine [2]. The quality of a Web crawler may directly affect the information searching quality. A Web crawler [3] algorithm is used to traverse a web page by extracting inlinks and outlinks. Extracted links are seeded in URL list for future use. The Web documents are linked by multiple hypertext links which connect diverse resources. Web crawler implements various Web search algorithms for retrieving and locating Web documents, such as page rank algorithm, best first, breadth first, shark search, and hypertext-induced topic selection (HITS). The implementation of different Web search crawling algorithms provides different search efficiencies. By analyzing various Web search algorithms, this paper proposes a competent crawling algorithm (CCA), which is derived from best crawling algorithms.

2 Crawler

A Web crawler [3, 4] is a relatively simple automated program or script that methodically scans or "crawls" through Internet pages to retrieve information from Web data. Alternative names for a Web crawler include Web spider, Web robot, bot, crawler, and automatic indexer. The documents in Web contain graph structure, i.e., connected through hyperlinks (Fig. 1).

Crawl manager starts [2] crawling from a given set of URLs, progressively fetches and scans them for new URLs (*out-links*), and then fetches these pages in turn, in an endless cycle. The newly founded URLs are taken for next cycle process to extract *in-links* and *out-links* of the respective Web pages. The traversed or visited pages are stored in buffer for future use, and the *out-links* of Web pages are stored to visit list in frontier. These URLs are classified using ontology editing tools

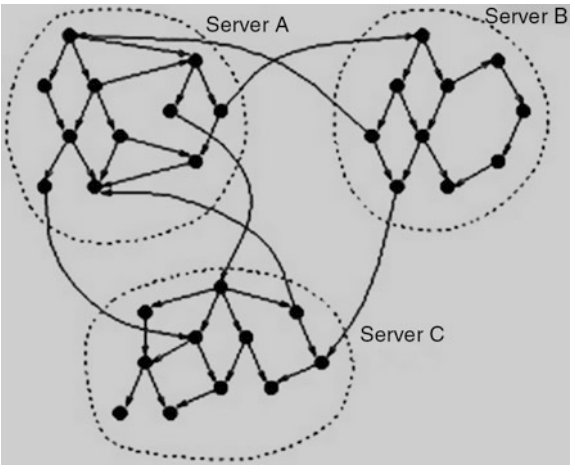

Fig. 1 Hyperlinked graph structure of Web documents

and indexed by indexer mechanism to enhance the Web search efficiency. The Web page (HTML/XML) contents are parsed using parser mechanisms, and the interpreted data are used for constructing inverted matrix system. The inverted matrix index contains the number of occurrences of words and location of the text in particular document. The indexer constructs keyword/search key from the inverted index using ontology classification to enhance the information retrieval.

2.1 Web Crawler Architecture

The architectural design of crawler (Fig. 2) describes the sequential flow of URL extraction and parsing of Web documents for efficient information retrieval. The in-links and out-links present in traversed Web pages are extracted and listed in frontier to visit.

To avoid duplication in URL extraction, the URL manager implements exclusion function, so that already extracted URLs will automatically get deleted from the frontier list. The extracted URLs are stored in hash table based on URL ranking

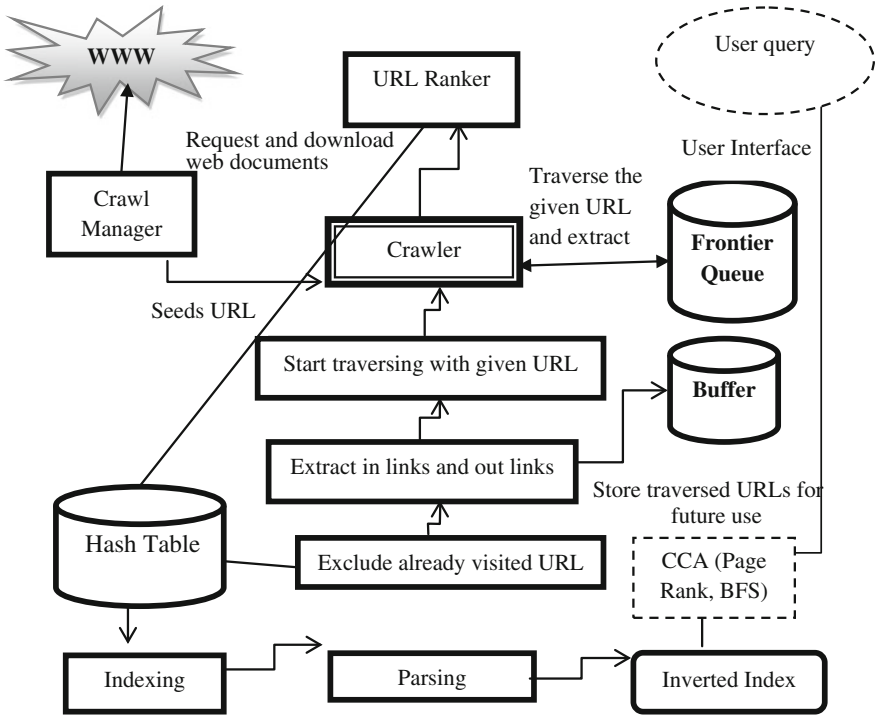

Fig. 2 Architecture of Web crawler

without any duplication. The indexer generates index from hash table. The parser interprets the data from Web pages and classifies the data based on ontology mechanisms to construct inverted index. The page ranker prioritizes the URLs and displays the result to user interface based on the user queries.

2.2 Work Flow of Crawlers

In this section, the work flow of crawlers illustrated using a Win Web Crawler tool [5].

Step 1: The crawler gets URL as an input (Fig. 3).

In step 1, URL www.csbdu.in is given as input to the Win Web Crawler. From the input, URL crawler starts the process by extracting the in-links and out-links.

Step 2: Retrieve and download the Web page. In step 2, the URLs associated with www.csbdu.in are extracted and processed by Win Web Crawler (Fig. 4).

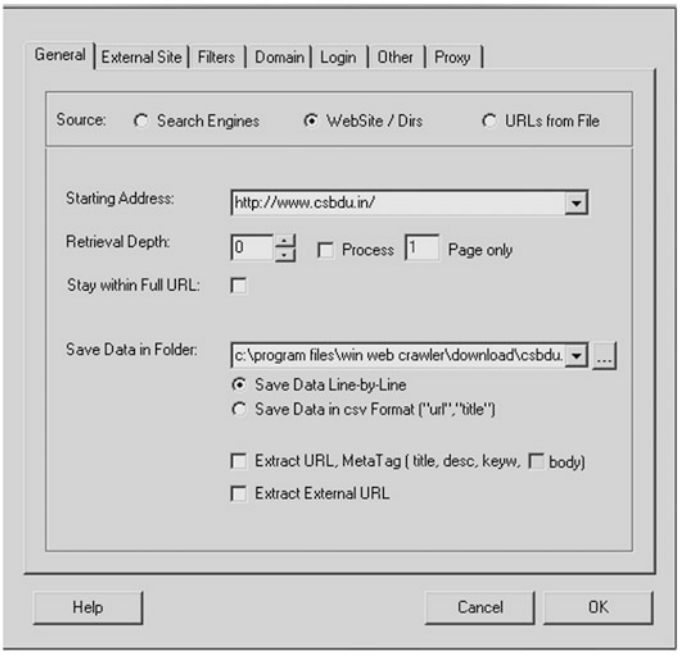

Fig. 3 Cycle 1—input

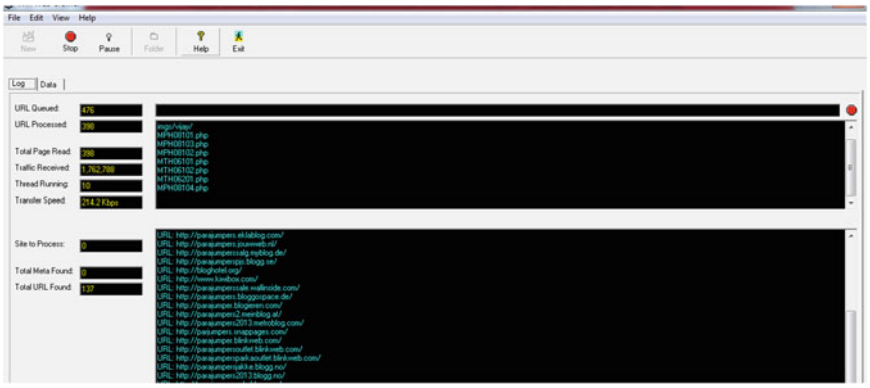

Fig. 4 Extract URLs

The process provides the count of processed URLs, queued URLs, total page read, traffic received, and total URLs found in the particular URL.

Step 3: Parse and extract the in-links and out-links from the retrieved Web pages to traverse (Fig. 5).

After the completion of step 3, the crawler produced the count 1,758 of total URLs found in www.csbdu.in. The extracted 1,758 URLs are stored in frontier to visit it. The crawler will traverse 1,758 URLs to extract and download the in-links and out-links of these URLs. The cycles will get repeated, until it reaches the first URL.

Step 4: The process is repeated, until it reaches the starting node.

Fig. 5 Completion of first cycle

2.3 Crawling Algorithm's Performance Comparisons

Generally, five different crawling algorithms are used for Web search. They are breadth first, best first, fish search, shark search, and page rank. In this section, the efficiency of different crawling algorithms is examined to find out the best one. The efficiency of the crawling algorithms is measured by the following important factors: precision, recall and F-score values [6].

$$\text{Precision}(P) = \text{No. of Relevant Pages Retrieved}/\text{Total Number of Retrieved Pages}$$
$$\text{Recall}(R) = \text{No. of Relevant Pages Retrieved}/\text{Total number of Relevant Pages}$$
$$\text{F-Score} = 2(P * R)/(P + R)$$

By considering the above factors, the efficiency of crawling algorithms is analyzed and stated below. The widely used popular crawling Web search algorithm is page rank algorithm. It was proposed by Tripathy and Patra [2] as a possible algorithm for monitoring and analyzing the user behaviors. The page rank algorithm works on the basis of number of in-links and out-links to a Web page. Initially, the page rank is set to 1 for all [7]. A page's score is calculated recursively upon the scores of the pages that point to it, called *in-links*. A Web page will get more score and priority, if it has more number of in-links. Best-first crawlers have been studied by Cho et al. [2] and Hersovici et al. [2]. The phenomenon of the best-first search is based on the factors such as precision, recall, accuracy, and F-score. The URL is selected for crawling, using the frontier as a priority queue based on maximum visit or by its popularity. In this algorithm, the URL selection process is guided by the lexical similarity between the topic's keywords and the source page of the URL [8] (Fig. 6).

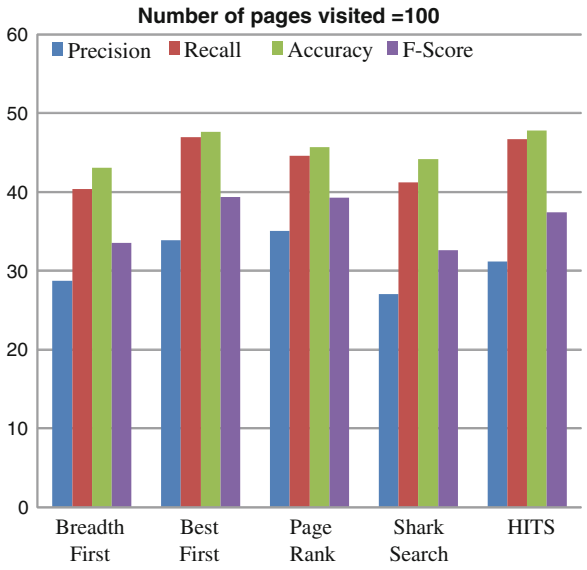

Fig. 6 Graphical view of crawling algorithms: performance measures

Table 1 Performance measures

Algorithms	Precision	Recall	Accuracy	F-score
Breadth first	28.8	40.4	43.12	33.61
Best first	33.93	47	47.7	39.41
Page rank	35.09	44.63	45.78	39.29
Shark search	27.05	41.23	44.25	32.66
HITS	31.23	46.77	47.85	37.45

Breadth-first algorithm [7] is the simplest and easy technique to traverse in the order of first in first out. BFS algorithm uses the frontier URL list of traversing the Web pages. Frontier is used as FIFO queue and crawls the links in the order in which they are encountered. The HITS algorithm works on the basis of relevant topic search. The algorithm classifies Web pages into two types in order to calculate page scores. If a page provides trustworthy information for a given topic, then it is classified as authority pages. The Web pages that give links to authority pages are called hub. Weights are assigned to every hub, and authority pages and page score will be calculated accordingly.

The experimental dataset for comparing these Web search algorithms is taken from [7]. Experiments are conducted using the Lund dataset containing 100 attributes. The precision, recall, accuracy, and F-score factor are examined using breadth-first, best-first, fish search, shark search, and page rank algorithms. The results and finding of the experiment are shown in Table 1.

Based on the dataset experiments [2, 7] with the following performance measures such as precision, recall, accuracy, and F-score, which are taken into account for their assessment, it is observed that page rank algorithm outperforms other algorithms by various performance measures.

2.3.1 Competent Crawling Algorithm

To enhance the performance of the crawler, in this paper, competent crawling algorithm (CCA) is proposed from the existing Web search crawling algorithms, to increase the efficiency of information retrieved. From the conclusion of the last section, page rank algorithm performs better than other algorithms. The proposed algorithm CCA combines the functionality of both page rank and BFS algorithms to enhance the efficiency of Web search. Compared to existing crawling algorithms, CCA has several advantages to increase the time efficiency by dequeuing the visited URLs from buffer before the crawler encounters it. The major advantages of the CCA are scalability and robustness. In CCA, dynamic hash tables are used for scalability and the system is reliable to crawler crashes. The complexity of the searching problem will overcome by CCA.

Pseudo code **:**

```
CCA (Starting Url){ For ( P=0; p <=Starting
Url; p++){ Enqueue (Frontier, url,max_trip_sc)
RetrivedUrl=Dequeue
(Fortier);Do{RetrivedPage=Fetch(RetrivedUrl);If (
multiples(trip)){ Max_trip_sc;}     Trip+=1;
            }While(visited<maxPages);ENQUEUE(BufferedPages
,max_trip_sc,Frontier) If(max_trip_sc>VisitedPages){
            DEQUEUE (max_trip_sc);
}If(Frontier>max_buffer){     DEQUEUE
(Frontier);}Merge(Frontier,max_trip,BufferedPages);}
```

3 Conclusion

In this paper, the best features of page rank algorithm and best-first algorithm are combined together to provide the best output. The functions of Web crawler components are described using the architectural design. Win Web Crawler tool clearly examines the working process of a crawler using an example. The comparative study on various crawling algorithms gives clear idea for better information retrieval. An attempt has been made to implement CCA, which provides the efficiency, scalability, and robustness. The newly proposed algorithm is time-efficient by dequeuing the repeated pages. Implementation of CCA in C# has taken as future work of this paper.

References

1. K.S. Shetty, S. Bhat, S. Singh, Symbolic verification of web crawler functionality and its properties, in *International Conference on Computer Communication and Informatics* (*ICCCI*, 2012)
2. A. Tripathy, P.K. Patra, A web mining architectural model of distributed crawler for internet searches using PageRank algorithm, in *IEEE Asia-Pacific Services Computing Conference* (2008)
3. A. Guerriero, F. Ragni, C. Martines, A dynamic URL assignment method for parallel web crawler. IEEE
4. A. Vadivel, S.G. Shaila, R.D. Mahalakshmi, J. Karthika, Component based effective web crawler and indexer using web services, in *IEEE-International Conference on Advances in Engineering, Science and Management* (*ICAESM*, 2012)
5. R.R. Trujillo, Simulation tool to study focused web crawling strategies (2006)
6. Accuracy, Precision, Recall and F-Score. Wikipedia, the free encyclopedia
7. S. Jaiganesh, P. Babu, K. Nimmati Satheesh, Comparative study of various web search algorithms for the improvement of web crawler. Int. J. Eng. Res. Technol. (IJERT) **2**(4) (2013)
8. Y. Yang, Y. Du, Y. Hai, Z. Gao, A topic-specific web crawler with web page hierarchy based on HTML Dom-Tree, in *Asia-Pacific Conference on Information Processing* (2009)

http://www.springer.com/978-81-322-2134-0

Artificial Intelligence and Evolutionary Algorithms in Engineering Systems
Proceedings of ICAEES 2014, Volume 2
Suresh, L.P.; Dash, S.S.; Panigrahi, B.K. (Eds.)
2015, XVII, 873 p. 484 illus., Softcover
ISBN: 978-81-322-2134-0