APPLICATIONS OF A BEE-SWARM ALGORITHM BASED CRAWLER

[1]Mgbeafulike, I. J.
ike.mgbeafuike@gmail.com

[2]Ojiako, F.N.
lightmanfranklin1989@gmail.com

[1, 2]Department of Computer Science
Chukwuemeka Odumegwu Ojukwu University,
Uli Campus, Anambra State, Nigeria.

ABSTRACT
A web-crawler is a program that is used by search engines to build an index of web pages found by the crawler while it searches for relevant web pages on the worldwide web. The use of bee-swarm algorithm in implementing web crawlers has solved a lot of issues previously encountered in web crawling namely: (a) problem of bandwidth utilization in implementing web crawlers considering the fact that bandwidth is neither free nor infinite as it is restricted by the laws of Physics. The web can now be crawled in a scalable manner and also in a way that utilizes minimum system resources. (b) the relevance of crawled web pages has significantly increased too. The newly developed web crawling system has shown that the relevance of web pages returned by a web crawler that was implemented with the bee-swarm algorithm is very high, thereby improving the user's experience and reducing the wastage of system resources like time and memory. (c) The huge and ever increasing number of web pages and the rate at which the content of a web page changes is no longer an issue. This is due to the fact that the newly implemented system is a focused web crawling system that searches and indexes only the web pages that contain the keywords in the search query in order to gather documents on a specific topic. In other words, only a subset of the entire web is traversed. The methodology adopted was the Feature-Driven Development(FDD) methodology. FDD is a developer centric process which consists of modeling a feature into an overall shape and delivering the model as a build in short two weeks iterations. It focuses on specific units of work that go through a stringent process which proceeds through domain walkthrough, design, design inspection, code, code inspection and the promotion to build. This research findings show that the use of bee-swarm algorithm in implementing a web crawler shows a significant improvement in users search experience as it uses collective intelligence in returning only relevant web pages based on a specific topic and avoids irrelevant ones. This article aims at leveraging on the benefits of the above solved challenges to explore the various areas where this topic based crawling system can be very useful.

Keywords: Web crawler, bee-swarm, algorithm, web pages, internet

1.0 INTRODUCTION

Web-crawler is a computer software that enables the search engines to build an index of web pages found on the Worldwide Web, Cabot (2017). Searches are made in the search index in real time and not on the web. An index is analogous to the index found at the back of a textbook.

It consists of words and their corresponding pages on a website. The disparity is that a search index uses hyperlinks and is dynamic while a text book index uses page numbers and is static. A web-crawler can also be referred to as a web spider, spider bot, web bot, automatic indexers and robots. To get the result of a search query, for example, if you type "What is the difference between web design and web development" and press the enter button, the web crawler begins by crawling content on websites, after which it creates a search index for the search engine and lastly uses a search algorithm to order the web pages based on their relevance to the search query before presenting the relevant pages to the search engine. It is important to note that a user does not make his searches in real time due to the huge and constantly increasing number of web pages on the web which is estimated to be in tens of billions and is still increasing in number because people keep creating websites daily around the world. Consequently, the search engine delays a little before presenting results of a search query to the user. Masanès (2007) explains that a web-crawler begins with a given URL say U_1, it searches through the pages found on U_1 looking for the keywords in the search query. As it searches for the keywords on the first web page, it moves on to search other web pages using the hyperlinks to discover other URLs (U_2 U_3 U_4..... U_n) that contain the keywords. The crawler afterwards creates a data structure of all the URLs discovered. This data structure is usually a queue which is processed on Last In First Out(LIFO) or First In Last Out(FILO) basis and is called the frontier while the URL it began with is called the seed.

This article aims at exploring the possible areas of human endeavour where web crawling through the implementation of a web-based crawler using the Bee-Swarm Algorithm that will be able to search the web based on a particular topic can be useful.

2.0 LITERATURE REVIEW

Early crawlers attempted to download all the web pages on the web servers around the world. Consequently, irrelevant web pages were fetched and stored in the search engines' database after consuming resources like time, memory and bandwidth. The crawling of the entire web is also non-pragmatic due the enormous web size. This lead to the research into crawlers that search the web based on specific topic, thereby fetching only the paged that are relevant to the keywords of a search query and improving users search experience. Google was not the first search engine that emerged but it simply became popular because of the strategy with which the search engine was designed and their rankling algorithms were simply amazing.

There is need for continuous crawling of the web due to the high rate of changes that occur on websites even as a crawler is downloading a web page content, this will help make the crawler fetch web pages with current information. In other words, fresh rates of web pages in increased. The concept of topical or focused crawlers has been approached by some studies using many interesting strategies (Davidson, 2000). In the research into topical crawlers, the strategy of choosing web pages has inspired most of the research studies; another thing that inspired most research was features of the graph built from pages already seen. The process of web crawling can be perceived as searching for a problem on the web with several objectives and rules of engagement. Crawlers differ in their mechanisms for using the evidence available to them. In studying topical web spiders, its crawling nature is also considered. The important features of a crawl like the format of a search query input, the keywords, user-log and characteristics of

downloaded pages (e.g. authoritative pages) can make all the required difference. This can be achieved by specifying the number of web pages to be downloaded. So many objectives and inadequate awareness of the search space compounds the problem because some crawlers may have to perform a local vs global optimization (Pant et al, 2002). Comparisons must be fair and made with an eye towards drawing out statistically significant differences. Not only does this require enough crawl runs but also sound methodologies that consider the temporal nature of crawler outputs. Significant challenges in evaluation include the general unavailability of relevant sets for particular topics or queries. Thus evaluation typically relies on defining measures for estimating page importance. A study by Baeza-Yates *et al.* (2005) indicated that the Online Page Importance Computation(OPIC) technique and other techniques that use the length of each site queue outsmarts breadth first crawling. Their study also revealed that it is good to use the result of a previous crawl to guide the current one. They simulated 3million web pages divided into two groups. The web pages were from the .gr and .cl domain and they experimented on some web crawling strategies.

According to Shujaa and Bahaa (2013) Search engines are using web spiders to crawl the web in order to collect copies of the web sites for their databases, these spiders usually use the technique of breadth first search which is non-guided (blind) depends on visiting all links of any web site and one by one. Shujaa and Bahaa (2013) proposed a new algorithm for crawling web depending on swarm intelligence techniques, the adopted algorithm is bee swarm algorithm which takes the behavior of the bee for its work, the result in terms of speed and accuracy which means the relevancy of the collected sites.

2.1 WEB CRAWLERS

Web crawlers are programs that exploit the graphical structure of the Web to move from page to page. In their infancy such programs were also called wanderers, robots, spiders, and worms, words that are quite evocative of Web imagery. It may be observed that the noun "crawler" is not indicative of the speed of these programs, as they can be considerably fast. In our own experience, we have been able to crawl up to tens of thousands of pages within a few minutes (Henzinger et al, 1998) From the beginning, a key motivation for designing Web crawlers has been to retrieve Web pages and add them or their representations to a local repository. Such a repository may then serve particular application needs such as those of a Web search engine.

2.2 WEB CRAWLING METHODOLOGIES

Given the current size of the Web, even large search engines cover only a portion of the publicly available part. A 2005 study showed that large-scale search engines index no more than 40-70% of the indexable Web a previous study by Steve Lawrence and Lee Giles showed that no search engine indexed more than 16% of the Web in 1999 (Gulli et al, 2005). As a crawler always downloads just a fraction of the Web pages, it is highly desirable that the downloaded fraction contains the most relevant pages and not just a random sample of the Web. This requires a metric of importance for prioritizing Web pages. Different methodologies and types of web crawler has been defined and used by different people. These includes (i) Focused crawling: The importance of a page for a crawler can also be expressed as a function of the similarity of a page to a given query. Web crawlers that attempt to download pages that are

similar to each other are called focused crawler or topical crawlers. The concepts of topical and focused crawling were first introduced by Menczer (1997) and by Chakrabarti *et al* (1999)..

2. (ii) Restricting followed links: A crawler may only want to seek out HTML pages and avoid all other MIME types. In order to request only HTML resources, a crawler may make an HTTP HEAD request to determine a Web resource's MIME type before requesting the entire resource with a GET request. To avoid making numerous HEAD requests, a crawler may examine the URL and only request a resource if the URL ends with certain characters such as .html, .htm, .asp, .aspx, .php, .jsp, .jspx or a slash. This strategy may cause numerous HTML Web resources to be unintentionally skipped (Cho, 2001).
3. Some crawlers may also avoid requesting any resources that have a "?" in them (are dynamically produced) in order to avoid spider traps that may cause the crawler to download an infinite number of URLs from a Web site. This strategy is unreliable if the site uses a rewrite engine to simplify its URLs.
4. (iii) URL normalization: Crawlers usually perform some type of URL normalization in order to avoid crawling the same resource more than once. The term URL normalization, also called URL canonicalization, refers to the process of modifying and standardizing a URL in a consistent manner. There are several types of normalization that may be performed including conversion of URLs to lowercase, removal of "." and ".." segments, and adding trailing slashes to the non-empty path component (Pant et al, 2009).
5. (iv) Path-ascending crawling: Some crawlers intend to download as many resources as possible from a particular web site. So *path-ascending crawler* was introduced that would ascend to every path in each URL that it intends to crawl (Cothey, 2004) .
6. (v) Academic-focused crawler: An example of the focused crawlers are academic crawlers, which crawls free-access academic related documents, such as the *citeseerxbot*, which is the crawler of CiteSeer^{X} search engine. Other academic search engines are Google Scholar and Microsoft Academic Search etc. Because most academic papers are published in PDF formats, such kind of crawler is particularly interested in crawling PDF, postscript files, Microsoft Word including their zipped formats. (Wu et al, 2012).

2.3 BEE SWARM ALGORITHM

The bee-swarm algorithm is a swarm-intelligence algorithm that leverages on the collective foraging intelligence of honey bees. Its usefulness has been proven in solving optimization problems, combinatorial and continuous optimization problems. The algorithm consists of two stages: the initialization stage and the search cycle. The search cycle is repeated until a specified condition is satisfied or for a given number of times and the stages include: recruitment, local search, neighborhood shrinking, site abandonment, and global search (Pham et al, 2009). He explained further that each potential solution comprises of a food source which is analogous to a flower and a population which is seen as the bee colony. The algorithm uses these parameters to find the optimal solution. According to Baris et al (2013), in nature, honey bees have several complicated behaviors such as mating, breeding and foraging. These behaviors have been mimicked for several honey bee based optimization algorithms.

One of the famous mating and breeding behavior of honey bees inspired algorithm is Marriage in Honey Bees Optimization (MBO). The algorithm starts from a single queen without family and passes on to the development of a colony with family having one or more queens. In the

literature, several versions of MBO have been proposed such as Honey-Bees Mating Optimization (HBMO), Fast Marriage in Honey Bees Optimization (FMHBO) and The Honey-Bees Optimization (HBO).

The other type of bee-inspired algorithms mimics the foraging behavior of the honey bees. These algorithms use standard evolutionary or random explorative search to locate promising locations. Then the algorithms utilize the exploitative search on the most promising locations to find the global optimum. The following algorithms were inspired from foraging behavior of honey bees; Bee System (BS), Bee Colony Optimization (BCO), Artificial Bee Colony (ABC) and The Bees Algorithm (BA). Bee System is an improved version of the Genetic Algorithm (GA) . The main purpose of the algorithm is to improve local search while keeping the global search ability of GA (Baris et al, 2013).

3.0 THE NEW SYSTEM

The new system (bee crawling system) was built using VB.NET and a relational database management system to store web pages= Microsoft Access. The system is designed so as enable for the searching of the web via certain specific parameters so as to enhance and enrich the inter user's search experience. It makes use of the bee swarm algorithm to scan through any selected webpage to look for items or topics relating to the search query that the user seeks information on. The new web crawler application will be used for searching information related to any topic of interest from the Internet. It identifies the most promising links that lead to target documents, and avoid off topic searches. In addition, it does not need to collect all web pages, but selects and retrieves relevant pages only. It starts with a topic vector, and for each URL, the relevance is computed for the contribution of web page in the selected domain. If it is found to be important, it gets added to the URL list else, gets discarded.

3.1 Crawling algorithm

The bee swarm algorithm is employed for the web crawler and it is as presented below:
Get initial web search space as n.
Get Fitness Value
Set i=0
While i<= Fitness Value
1. Increment i
2. Select the best closest links nodes
3. Recruit new "bee" nodes
4. Evaluate the fitness value of each new node
5. Sort the results based on their fitness value
6. Allocate the rest of the "bee" nodes for global search for other locations on the web page
7. Evaluate the fitness value for the other locations
8. Sort the overall results based on their fitness values
9. Rerun code until Fitness value is reached

3.2 System Architecture

The architecture of the new web crawler system is displayed in the figure 3.1.

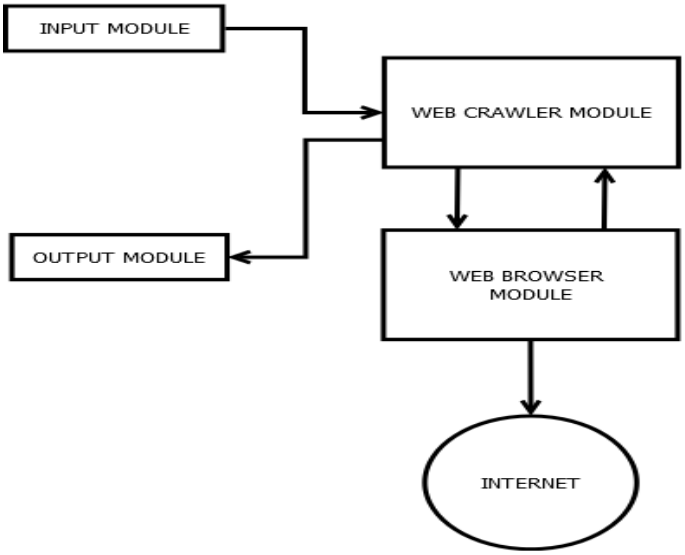

Figure 1: Architecture of the New System

The following makes up the system:
Input module: This is the module that collects input from the user
Output module: this is the module that retrieves information and display to the user.
Web Crawler Module: Makes use of the bee swarm algorithm to search the web for data
Web Browser Module: Used to display the web page that is crawled
Internet: The access to the World Wide Web.

3.3 SYSTEM IMPLEMENTATION

The system was implemented using the following interfaces
System Login Form: This form lets the system admin submits login details to gain access into the system.

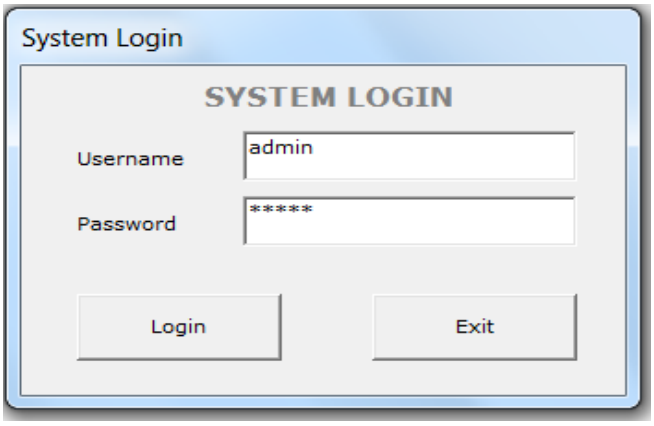

Figure 2: System Login

Web Crawler Input Form: This interface is used to enter the desired parameters that will be used for crawling the web via the internet.

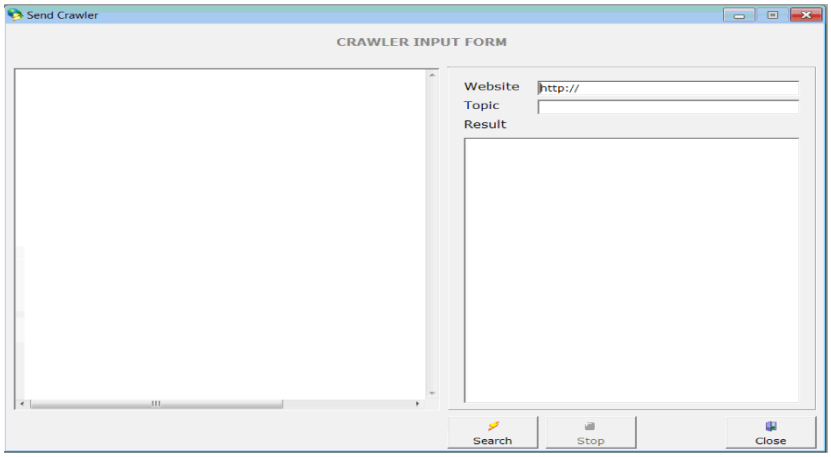

Figure 3: Web Crawler Input Form

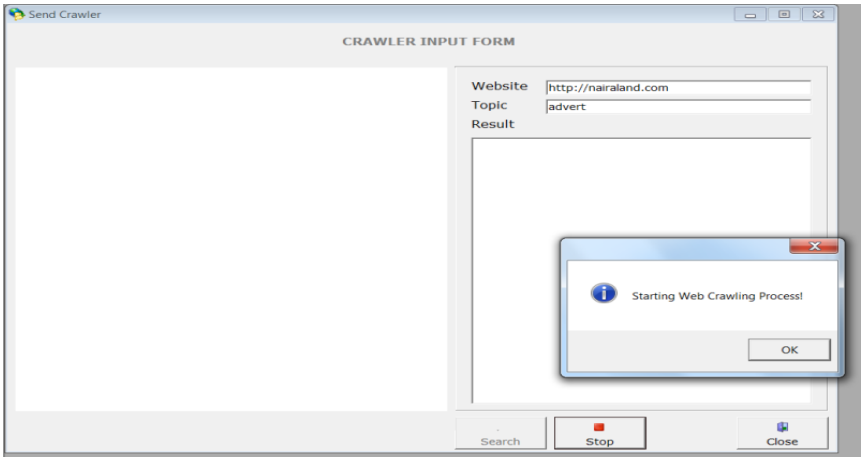

Figure 4: Web Crawling Commencement Process

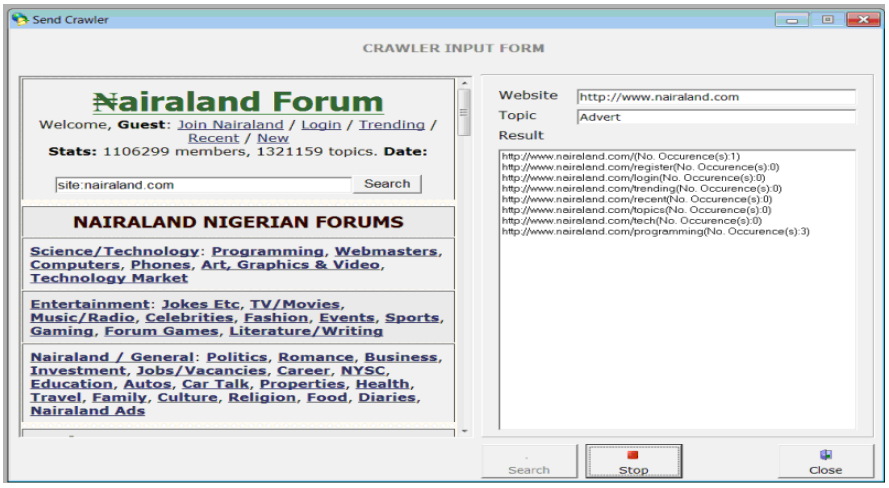

Figure 5: Web Crawling Process

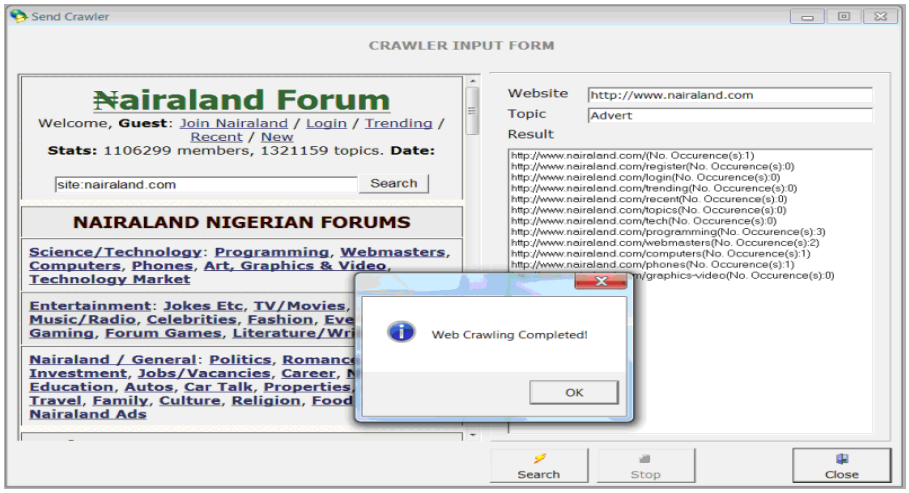

Figure 6: Web Crawling Termination

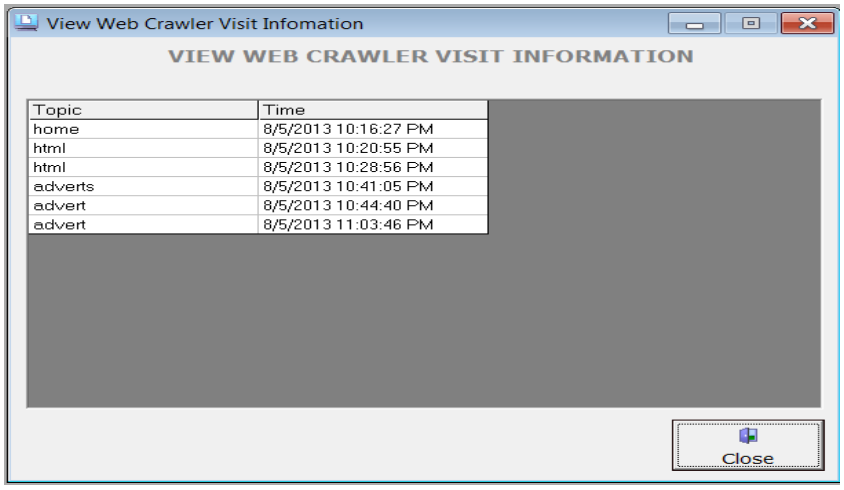

Figure 7: Crawler Visit Info Form

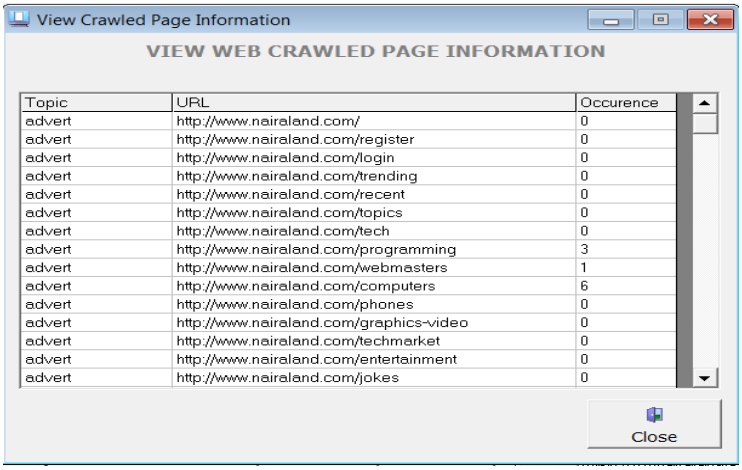

Figure 8: Crawled Page Info Form

3.4 APPLICATION AREAS

Bee-swarm algorithm based crawlers can be very useful in the following areas:

i. Powering price comparison site: E-Commerce business companies can automate this process in order to monitor prices of goods on different websites. This helps them in competing with other companies who may be selling likely products by sending them cold emails and sales pitch. It can be very useful for buyers too, if someone wants to buy some electronics gadgets, he can crawl information from different websites in order to determine the company that sells better and cheaper product. In this 21st century, if your business is not on the internet, you will soon be out of business. People now leverage on various online trading platforms like 1688, Alibaba, Aliexpress, taobao, Amazon, Ebay etc. for price comparison and purchase of goods from the international market.
ii. Search engine optimization: it is the process of getting traffic from the free, organic, editorial or natural search results on search engines. This helps to improve search presence e.g. Semrush uses it.
iii. Analyzing and monitoring competitors
iv. Product cataloging
v. Analytics and market research across all industries
vi. Fueling job boards(e.g. JobPiks, Jobberman): This is the curation and aggregation of available jobs and qualified job seekers.
vii. Media monitoring: To be continuously updated on what is making news around around the world, you can program a tool to help you store all those information, keeping you abreast with the latest happenings.
viii. Social media analysis(Twitter and Pinterest): You can crawl a lot of user's information here. If you want to avoid your information from being crawled, you can use the settings to hide some information you want to keep secret and also turning off your device location.
ix. Content production(Blogging, autoblogging, content curation and aggregation)

4.0 CONCLUSION AND DISCUSSION

This work provides invaluable material for researchers interested in the field of Web Crawling in any system. Inept study of the performance capabilities of various web crawling algorithms will give more insight into how to best apply them depending on the focus of the search. This will improve the overall overhead costs, increase the efficiency and work output of the system and reduce wastage of time in system resources management. Swarm intelligence has proven to be a good improvement in web crawling using keywords for a group of sites with speed and relevancy of the crawled sites to desired topics.

REFERENCES

1. Abiteboul, S., Mihai, P., & Gregory, C.(2003). Adaptive on-line page importance computation. *Proceedings of the 12th international conference on World Wide Web.* Budapest, Hungary: ACM. pp. 280–290.
2. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A. and Raghavan, S. (2001). Searching the Web. *ACM Transactions on Internet Technology*, 1(1), 2001.

3. Baeza-Yates, R., Castillo, C., Marin, M. and Rodriguez, A. (2005). Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering. *In Proceedings of the Industrial and Practical Experience track of the 14th conference on World Wide Web*. ACM Press, Chiba, Japan. Pp. 864–872.
4. Boldi, P., Bruno C., Massimo S., Sebastiano V. (2004). UbiCrawler: a scalable fully distributed Web crawler. *Software: Practice and Experience* 34 (8): 711–726. doi:10.1002/spe.587. Retrieved 2009-03-23.
5. Cabot, T. (2017). Web Crawlers, Everything You Need to Know. Medium. [online] *Available From <https://medium.com/@cabot_solutions/web-crawlers-everything-you-need-to-know-6dce26ee8ad8> (Feb 28, 2017).*
6. Chakrabarti, S., VandenBerg, M., and Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11–16):1623–1640.
7. Cho, J. (2001). Crawling the Web: Discovery and Maintenance of a Large-Scale Web Data, *Ph.D. dissertation, Department of Computer Science, Stanford University.*
8. Cothey, V., (2004). Web-crawling reliability. *Journal of the American Society for Information Science and Technology*. 55 (14): 1228–1238.
9. Davison B.D. (2000). Topical locality in the web. *In Proc. 23rd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2000.*
10. Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., and Gori, M. (2000). Focused crawling using context graphs. *In Proceedings of 26th International Conference on Very Large Databases (VLDB)*, Cairo, Egypt. Pp. 527-534.
11. Jian, W., Pradeep, T., Madian, K., Stephen, C., Douglas, J., Jose, S., Wandelmer, Xin, L., Prasenjit, M., Lee, C., (2012). Web crawler middleware for search engine digital libraries: a case study for citeseerX. *In proceedings of the twelfth international workshop on Web information and data management*, Maui Hawaii, USA. Pp 57-64.
12. Karaboga, D. (2005). An Idea Based on Honey Bee Swarm for Numerical Optimization. *Technical Report for Erciyes University; Kayseri, Turkey*.
13. Lawrence, S. and Lee, G. (2000). Accessibility of information on the web. Nature 400 (6740): 107. Bibcode:1999Natur.400..107L. doi:10.1038/21987. PMID 10428673
14. Lieberman, H., Christopher, F., and Weitzman, L.(2001). Exploring the Web with Reconnaissance Agents. *Communications of the ACM. Vol.*, 44(8):69.
15. Masanès, J. (2007). Web Archiving. *Springer*. Pg. 1. ISBN 978-3-54046332-0. Retrieved April 24, 2014.
16. Najork, M. and Wiener, L. (2001). Breadth-first crawling yields high-quality pages. *In Proceedings of the Tenth Conference on World Wide Web*, Elsevier Science. Hong Kong. Pp 114–118.
17. Pant, G., Srinivasan, P., and Menczer, F.(2002). Exploration versus exploitation in topic driven crawlers. *In WWW02 Workshop on Web Dynamics.*
18. Pant, G., Srinivasan, P., Menczer, F., Levene, M., Poulovassilis, A. (2004). Web Dynamics: Adapting to Change in Content, Size, Topology and Use. *Springer*. Pp. 153–178. ISBN 978-3-540-40676-1.
19. Patil, Y., Patil, S. (2016). Review of Web Crawlers with Specification and Working. *International Journal of Advanced Research in Computer and Communication Engineering.* 5 (1): 4.

20. Pham, D.T., Castellani, M. (2009). The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimisation Problems. *Proc. ImechE, Part C,* 223(12), 2919-2938.
21. Shujaa, M. I., Bahaa, U. A. (2013). Building web crawler based on bee swarm intelligent algorithm. *International Journal of Computer Science Issues (IJCSI).* Vol. 10 issue 5, p.134. Available from<http://connection.ebscohost.com/c/articles/91274174/building-web-crawler-based-bee-swarm-intelligent-algorithm.
22. Shkapenyuk, V. and Suel, T. (2002). Design and implementation of a high performance distributed web crawler. *In Proceedings of the 18th International Conference on Data Engineering (ICDE)*. IEEE CS Press, San Jose, California. Pp. 357-368.
23. Teodorovic, D. and Dell'Orco, M. (2005). Bee Colony Optimization-A Cooperative Learning Approach to Complex Transportation Problems. *Proceedings of the 10th EWGT Meeting and 16th Mini-EURO Conference;* Poznan, Poland. Pp. 51–60.