RESEARCH ARTICLE OPEN ACCESS

Investigation of Web Mining Optimization Using Microbial Genetic Algorithm

Dipali Tungar[#1], Prof. Amol D. Potgantwar[*2]
[#]Student, ME Computer, Sandip Institute of Tech. & Research Centre Nashik, University of Pune
[*]Faculty, Sandip Institute of Tech. & Research Centre Nashik, University of Pune

ABSTRACT
In today's modern internet era peopleneed searching on the web and finding relevant information on the web to be efficient and fast. But traditional search engines like Google suppose to be more intelligent, still use the traditional crawling algorithms to find data relevant to the search query. But most of the times it returns irrelevant data as well which becomes confusing for the user. In a normal XML data the user inputs the search query in terms of a keyword or a question and the answer to the search query should be more precise and more relevant. So, using the traditional crawling algorithms over XML data would lead to irrelevant results. Genetic algorithms are the modern algorithms which replicates the Darwinian theory of the natural evolution. The genetic algorithms are best suited for the traditional search problem as the genetic algorithms always tend to return quality as solution for any domain data. It would be a good approach to investigate how the genetic algorithms would be suitable for the search over the XML data of different domains. So, this system implements a steady state tournament selection Microbial Genetic Algorithm over the XML data of the different domains. This would be an investigation of how the genetic algorithm would return accurate results over XML data of different domains.

Keywords: XML data, Genetic Algorithm, Darwinian Theory, natural evolution.

I. INTRODUCTION

The keyword search model is popular today due to success of web search engines. Keyword search is proposed as an alternative means of querying the database. Keyword search is simple and familiar to most internet users as it only requires the input of some keywords. Keyword search in text documents take the documents that are more relevant with the input keywords as the answers.

XML is becoming a standard format of data representation, so it is desirable to support keyword search in XML database. XML is a user friendly and easy to understand. Traditional way to access XML databases is use of query languages. But this approach requires the knowledge of complex query languages and the database schema.

There are some challenges in Keyword searching over XML. First, the result of the keyword search query is not always an entire document, but it can be a nested tree of XML element. In general, XML keyword search results can be the “deepest” node containing the keywords. Second, the query results can ranked in different ways for XML and HTML keyword search. HTML search engines such as Google usually rank documents based on their hyperlinked structure. Since XML keyword search queries can return nested elements, ranking has to be done at the granularity of XML elements, as opposed to entire XML documents.

Traditional query processing approach on XML database is constrained by the query constructs imposed by the language such as XQuery and XPath. Firstly, the query language themselves are hard to comprehend for non-database users. For example, the XQuery is fairly complicated to grasp. Secondly, these query languages require the queries to be posed against the underlying structure and complex database schemas. These traditional querying methods are powerful but unfriendly for day-to-day users.

II. LITERATURE SURVEY

Although many research efforts has conducted in XML keyword search.

Fuzzy-Ahead [1] search approach over a XML data starts to guess the further part of the query that user may enter. It takes user query in terms of keyword to search and returns the data matching the search query approximately. This system also implements effective indexing and top-k algorithm to achieve higher accuracy. But the major drawback of this system would be that it may return very poor data as it searches the data by matching approximately.

XSEarch [2] is a semantic search engine for XML. It returns semantically related document fragments that satisfy the user□s query. Query answers are ranked using extended information retrieval techniques and are generated in an order similar to the ranking. Advanced indexing techniques were developed to facilitate efficient implementation of XSEarch [2].

XRANK [3] has a ranking mechanism which returns document fragments as answers. There is no distinction between keywords and labels and each keyword of an XRANK query is matched against every word of the document. XRANK ranks the elements of an XML document by generalizing the Page-Rank algorithm. It ranks the answers to a query by combining the ranking of elements with keyword proximity. An answer to an XSEarch [2] query is also an answer or some part of an answer to the XRANK query that consists of the same keywords and labels, but the converse is not necessarily true. Actually, XRANK may return answers with parts that are semantically unrelated. XRANK ranks the elements of an XML document by generalizing the Page-Rank algorithm of Google. It ranks the answers to a query by combining the ranking of elements with keyword proximity. The notion of proximity in XRANK means that the children of an element must be in the “right order” if that element should be ranked highly as an answer. In XSEarch [2], proximity is included in the ranking formula in terms of the size of the relationship tree and thus, it is not affected by the order of children. XSEarch [2] employs more information-retrieval techniques than XRANK [3], namely, tf-idf and similarity between the query and the document.

One widely adopted approach so far is to find the Smallest Lowest Common Ancestor (SLCA) of all keywords. Each SLCA [4] result of a keyword query contains all query keywords but has no subtree which also contains all the keywords. SLCA-based approaches only take the tree structure of XML data into consideration, without considering the semantics of the query and XML data. SLCA may introduce answers that are either irrelevant to user search intention, or answers that may not be meaningful or informative enough.

However, existing systems of keyword search over XML databases suffer from two problems: 1.Meaningfulness and completeness of answers, 2.The scope of the search. The existing approaches, such as SLCA [4] and XRank [3], return some irrelevant results and also miss some results from answers. Existing system return the answer of keyword search by taking only LCAs as the answer of keyword search which will be inaccurate. In addition, XML documents involve complicated structures, therefore it is difficult to find the meaningful desired data, which still preserves the structure relationship and conforms to the documents, for users through limited input keywords. Existing studies mainly focus on efficiency of keyword search on XML databases and usually leads to low effectiveness, and accordingly, how to discover the structure clue from the input keywords so as to improve the effectiveness.

A new information-access paradigm for XML data, called “Inks” [5] was proposed which searches on the underlying data “on the fly” as the user types in query keywords. Inks extend existing XML keyword search methods by interactively answering keyword queries. Here effective indices, early-termination techniques, and efficient search algorithms are proposed to achieve a high interactive speed.

Valuable Lowest Common Ancestor (VLCA) [6] was introduced to accurately and effectively answer keyword queries over XML documents. A new concept of Compact VLCA (CVLCA) [6] is developed which compute the meaningful compact connected trees rooted as CVLCAs as the answers of keyword queries.

So, this system investigates the performance of using an evolutionary approach of Genetic Algorithm on XML Search for various datasets and its accuracy will analyzed to justify whether it is good approach for XML Search.

III. METHODOLOGY

3.1 Microbial Genetic Algorithm –

Genetic Algorithms are powerful and widely applicable to search and optimization problems. Genetic Algorithms are based on the concepts of natural selection and evolution. One of the tried and tested genetic algorithms on search problems is Microbial Genetic Algorithm [8] which has optimum time complexity as well as data quality. However, no search system using Microbial Genetic Algorithm or any other Genetic Algorithm Search systems implement XML Search. Hence, it was decided to investigate the performance of Microbial Genetic Algorithm [8] on XML data search. This algorithm has several key components, which play an important role in the process.

3.1.1 Genotype –

Genotype is the full set of Genes that any individual in Population has. Each gene possesses a value, which will be a part of potential solution to given problem. In this system, the emphasis is given on keyword search and most of the times the title of the document contains all the crucial keywords of the document which convey the central idea. Therefore, taking this idea, the size of the genotype was kept to 10 so that each gene will hold a possible keyword of the central idea of the document.

Genotype Size N = 10
Fig. 1 Genotype

Fig. 1 Genotype

3.1.2 Phenotype –

Phenotype is individual solution to problem that Genotype encodes. For this system, it is necessary to encode each gene in terms of its Part of Speech type. Hence, each token of the title was passed through a simple English language parser, which returns the type of the token. If the token is Noun or Proper Noun then it is encoded as 1 else it is encoded as 0. Therefore, a possible solution will contain most of the nouns given in the search query.

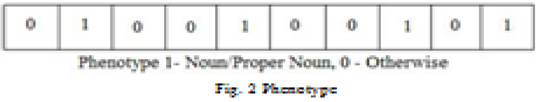

Fig. 2 Phenotype

3.1.3 Population –

The population size dictates the number of individuals in the population. Larger population sizes increase the amount of variation present in the initial population at the expense of requiring more fitness evaluations. It is found that the best population size is both applications dependent and related to the individual size. In this system, for the purpose of investigation the population size is kept varying.

3.1.4 Fitness Evaluation –

Fitness Evaluation is the most crucial part of the Microbial GA as it evaluates each member for its closeness to the optimum solution. In this system, the fitness function is carefully designed, as wrong evaluation would lead to poor and misleading data extraction. This system evaluates each member of the population based on how much the data in the member is close to the search query by user. Approximately the member 75% close to search query is considered as optimum solution.

3.1.5 Recombination Rate –

Crossover rate determines the probability that crossover will occur. The crossover will generate new individuals in the population by combining parts of existing individuals. The crossover rate is usually high and „application dependent□ . Many researchers suggest crossover rate to be between 0.6 and 0.95.

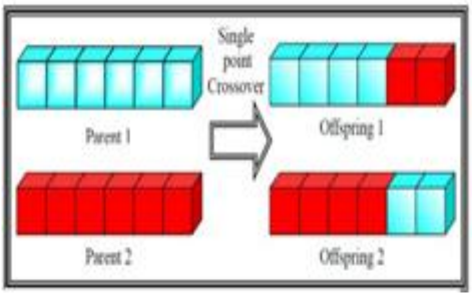

Fig. 3 Single Point Crossover

3.1.6 Mutation Rate –

Mutation rate determines the probability that a mutation will occur. Mutation is employed to give new information to the population and prevents the population from becoming saturated with similar chromosomes, simply said to avoid premature convergence. The best mutation rate is „application dependent□ . For most applications, mutation rate is between 0.001 and 0.1, while for automated circuit design problems, it is usually between 0.3 and 0.8.

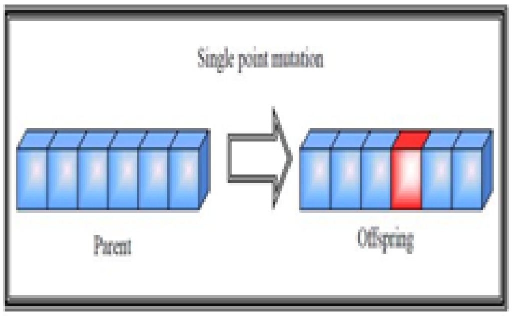

Fig. 4 Single Point Mutation

3.1.7 Algorithm –

```
Step 1 - Initialize the Recombination & Mutation
        Rates.

Step 2 - Repeat until the solution is not achieved
Step 2.1 Randomly select two members from the
        population.

Step 2.2 Compute the fitness of the selected
        members

Step 2.3 Check whether Fitness (Member1) >
        Fitness (Member2)

        Begin

        If yes then

                Mark member1 as Winner
                Mark Member2 as Loser
        end
        if no then
```

```
            Mark member2 as Winner
            Mark Member1 as Loser

        end End

Step 2.4 Recombine the Winner Genes with Loser
        Genes if the random Recombination value
        is less than Recombination Rate.

Step 2.5 Mutate the Loser Genes by variation
        Based on the Gene Values.

Step 2.6 Check the fitness of new generated off-
        spring.

            if Fitness(off-spring) == 100

            break the generation process

        end.
        else

            Overwrite off-spring with loser go
            to Step 2.1

        end.

Step 3 - Display the Solution Member of the
        Population

Step 4 - Terminate.
```

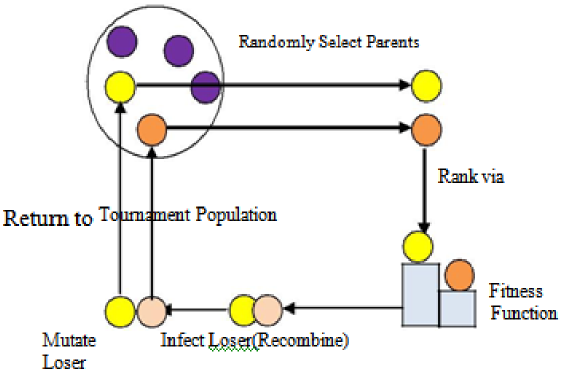

Fig. 5 Microbial Genetic Algorithm

3.2 Proposed System –

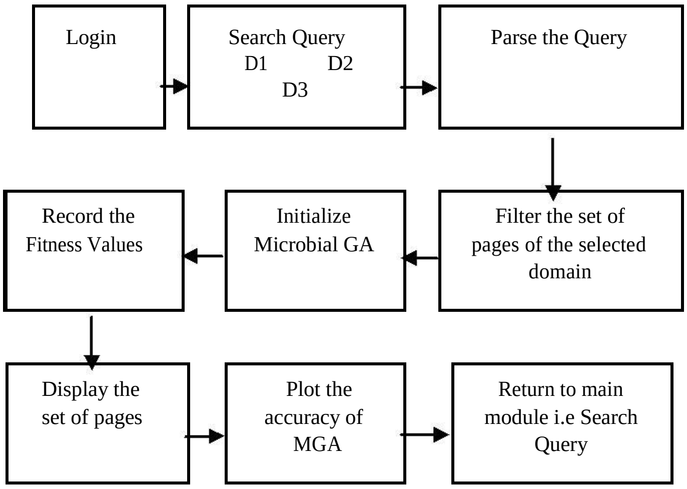
Login
Search Query
D1
D2
D3
Parse the Query
Filter the set of pages of the selected domain
Initialize Microbial GA
Record the Fitness Values
Display the set of pages
Plot the accuracy of MGA
Return to main module i.e Search Query

Fig. 6 Overall System Diagram

As shown in the Fig. 6 the system provides user authentication, which helps to maintain search history for each registered user. Once, the user has logged in successfully, it is provided with an interface, which asks to choose the data set on which the search should be performed as well as the search query to be used for searching. Once, the dataset is selected, all the data in the domain is parsed and transformed into the process required structure. After the data is pre-processed, the population of the Microbial GA is initialized and the process of selection and evolution starts. The necessary members close to solution are extracted and their fitness values are recorded. The data contained by these members is returned to the user and the plot of the fitness for current search query is presented to user. The same process can be repeated for other datasets as well.

IV. RESULT

4.1 Data Set –

4.1.1 DBLP Data Set –

DBLP dataset maintains a collection of computer conference journals, papers and proceedings. It has a collection of more than 2.3 million articles with their information like author, title of the paper, link to author□ s home page, etc.

4.1.2 Monodial Data Set –

Monodial dataset has a collection of geographical and political information like political changes in several countries after 1998, small dependent and independent island countries and geographical information of the countries.

4.2 Result Set -

Since the system is in design and implementation phase it is assumed that the system will perform better than the traditional tree based approaches for the XML Search. Any user within the system can monitor the performance of the search on any dataset.

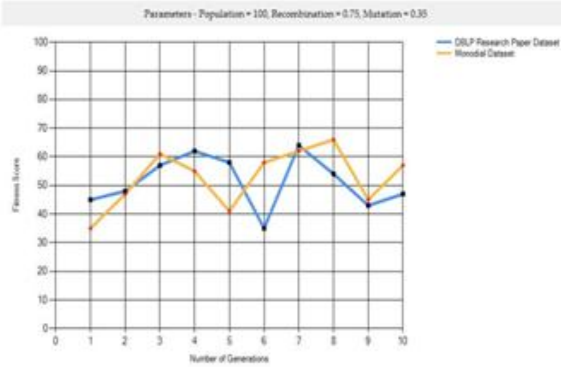

Fig. 7 Fitness value in each Generation

Fig 7 depicts the graph showing the fitness value in each generation of the search evolution. Average 65% of the accuracy is assumed to get when XML search is performed using Microbial Genetic Algorithm.

V. CONCLUSION

Genetic Algorithms (GAs) implement optimization strategies based on simulation of the natural law of evolution of a species by natural selection. GAs have been applied to a variety of function optimization problems, and have been shown to be highly effective in searching a large, poorly defined search space even in the presence of difficulties such as high-dimensionality, multi-modality,
discontinuity and noise. Therefore, Microbial Genetic algorithm may give optimum solution to user query for XML data retrieval.

This system implements a steady state tournament selection Microbial Genetic Algorithm over XML data set. This would be an investigation of how the genetic algorithm would return accurate results over XML data of different domains.

VI. ACKNOWLEDGMENT

We thank anonymous referees whose comments improved this paper.

REFERENCES

[1] J. Feng, And Li, G., “Efficient Fuzzy Type-Ahead Search in XML Data,” *Proc. of IEEETransactions on Knowledge And Data Engineering, Vol. 24 No. 5,* pp. 882-895, 2012.

[2] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, “XSEarch: A semantic search engine for XML,”*Proc. Int’l Conf. Very Large Data Bases (VLDB),* pp. 45-56, 2003.

[3] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, “XRANK:ranked keyword search over XML documents,” *Proc. ACM SIGMOD, Int’l Conf. Management of Data,* pp. 16-27, 2003.

[4] Y. Xu and Y. Papakonstantinou, “Efficient keyword search for smallest LCAs in XML databases,” *in SIGMOD, 2005,* pp. 537–538.

[5] G. Li, J. Feng, and L. Zhou, “Interactive Search in Xml Data,” *Proc. Int’l Conf. World Wide Web (WWW),* pp. 1063-1064, 2009.

[6] G. Li, J. Feng, J. Wang, and L. Zhou, “Effective Keyword Search for Valuable LCAS over XML Documents,” *Proc. Conf. Information andKnowledge Management (CIKM),* pp. 31-40, 2007.

[7] G. Li, B. Ooi, et al (2008) “Ease: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-Structured and Structured Data”, *In Proceedings of ACM SIGMOD International Conference on Management of Data,* pp. 903-914.

[8] I. Harvey, ”The Microbial Genetic Algorithm,” Unpublished report (1996).

[9] D.E. Goldberg (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts. J.H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).