International Journal of Engineering Research & Management Technology ISSN: 2348-4039
March-2016 Volume 3, Issue-2
Email: editor@ijermt.org www.ijermt.org

Survey on Improving Genetic Algorithm Using Searching Concepts of Data Structures for Query Optimization in Information Retrieval

Harpreet Kaur , Akshay Tahlani
Department of Information Technology,
SRM University, NCR Campus
Modinagar

ABSTRACT:
Information is an ultimate resource of every commercial and non commercial sector. With the advent of new technologies and popularize use of web, the information has been increasing at greater pace. The organization of this information into unstructured database repositories has made the Information Retrieval, a complex process. The use of Genetic Algorithm for retrieving the information from such large complex repositories has provided an improved approach to Information Retrieval Systems. However, these systems involve algorithms to improve the formulation for user query and provide ranking for representation of retrieved documents. The Genetic Algorithm matches the user queries against the database repositories to provide an optimized and relevant documents needed by the user. This paper represents the use of Sequential searching and Binary searching algorithms of Data Structures with an objective, to improve the efficiency and effectiveness of Genetic Algorithm for information retrieval from large databases.

KEYWORDS: Genetic Algorithm, Sequential Search, Binary Search, Information Retrieval

1. INTRODUCTION:

Retrieval of desired information over web from a large collection of documents requires an intelligible Information Retrieval Systems which serves the user requests by providing the information that the user needs. The general objective of an information Retrieval System is to minimize the overhead of a user locating needed information [3]. Overhead can be expressed as the time a user spends in all of the steps leading to reading an item containing the needed information (e.g., query generation, query execution, scanning results of query to select items to read, reading non-relevant items) [3], information overhead [8]. The user requests for information, in the form of queries, may sometimes, need guidance and requires reformulation. During the gradual increase in the amount of information over years, the storage systems provide complex ways to manage various large, unstructured segments of information. An Information Retrieval System is programmed software that deals with the user queries to provide relevant search solutions, containing the list of queries that may help the user to formulate its requirement, corresponding to the information that the user needs. The Information Retrieval Systems also provide the ways to calculate and assign the weights to the collection of documents in a search space. Research in IR includes modeling, document classification and categorization, systems architecture, user interfaces, data visualization, filtering, languages etc [3]. These researches have proved to be better solutions to reformulate the user queries, matching the documents with the required ones, retrieving the desired information and weighing the documents of the search space. Most of these researches involve the Genetic Evolution for the optimized search solutions. Genetic Algorithms started to be applied in information retrieval system in order to optimize query by genetic algorithm, a good query is a set of terms that express accurately the information need while being usable within collection corpus, the last part of this specification is critical to make the matching process efficient [9]. Genetic Algorithms provides evolutionary based search solutions over some

iterative generations. The user query is either modified or reformulated using the evolutionary search solutions provided, during the consecutive generations. In these generations, the documents are chosen that match best with their weights to the desired search query.
This paper provides a report to introduce the use of Data Structure searching concepts along with the initial stages of Genetic Algorithm to produce an array of result set that may use to formulate the user query. The proposed approach will improve the efficiency of the traditional Genetic Algorithm to provide the most relevant result for Information Retrieval.

1.1 INFORMATION RETRIEVAL (IR) PROCESS:

A document based IR system typically consists of three main subsystems: document representation, representation of users' requirements (queries) and the algorithms used to match user requirements (queries) with document representations [5]. An information retrieval process begins when a user enters a query into the system [8] [5]. Queries are formal statements of information needs, for example search strings in web search engines [8]. The representations of documents formulate the query for user requirements after matching and retrieving the document(s) from the database as per user needs. The architecture of Information Retrieval Process is shown in figure-1.1. Query formatting depends on the underlying model of retrieval used (Boolean models [Bookstein, 1985], vector space models [Salton & McGill, 1983], probabilistic models [Maron & Kuhns, 1960; Robertson, 1977], fuzzy retrieval models [Borgodna & Pasi, 1993], models based on artificial intelligence techniques [Maaeng, 1992; Evans 1993]) [5]. Precision is defined as fraction of retrieved documents which are relevant [1]. Recall is the fraction of documents that are relevant to the query that are successfully retrieved [8].

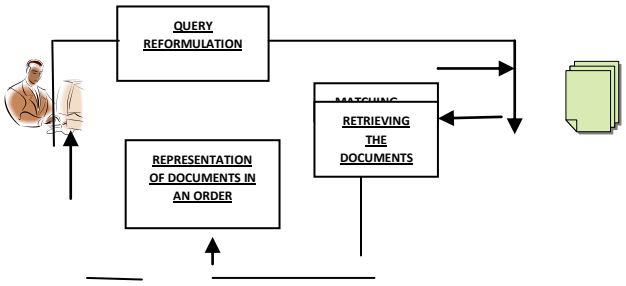


FIGURE – 1.1 INFORMATION RETRIEVAL PROCESS:

1.2 DATA STRUCTURES (DS) SEARCHING ALGORITHMS:

Data Structures (DS) is an important field of computer science that basically, deals with the storage, organization, management and representation of data or information [10]. The paper represents the use of two searching algorithms as described below.

(A) BINARY SEARCH:

Binary Search Algorithm performs searching from the mid of the provided array. It again divides the upper half; if the element to be search is greater than mid element else divide the lower half, for search element to be smaller. This process continues till search is reached [10].

(B) SEQUENTIAL SEARCH:

Sequential Search is performed by visiting each element of a sorted list of elements. Every element of the list is compared with the key element till the search is found [10].

1.3 CONCEPTS OF GENETIC ALGORITHM:

Genetic Algorithms are the most popular evolutionary based mechanisms that evolve out with search solutions over consecutive generations. Genetic Algorithms' applied natural selection and natural genetics in

artificial intelligence to find the globally optimal solution to the optimization problem from the feasible solutions [7]. These algorithms work on some initial set of sample space called initial population to evolve out with various combinational subsets over each generation until a relevant solution is obtained. The whole process involves the use of genetic operators such as selection, crossover and mutation. Genetic evolution process involves the some stages as shown in Figure-1.3.

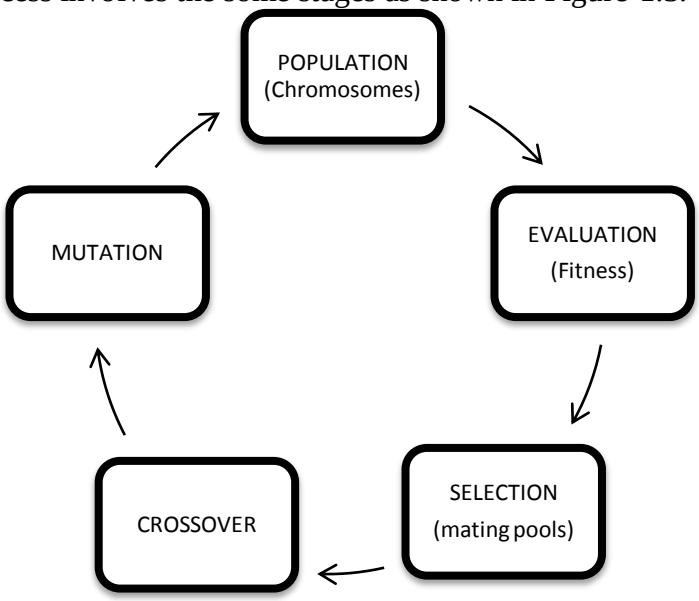


FIGURE-1.3 GENETIC ALGORITHM LIFE CYCLE

4. LITERATURE REVIEW:

Many authors have published their researches on Query Reformulation [1] [4] [2], Matching Documents [3] [2] [5] and Improving Genetic Algorithm [1] [4] [2] [9] for relevant Information Retrieval.

In paper [1] authors have provided an approach for the user query reformulation, specifically, by obtaining the candidate terms to reformulate the query from a morphological thesaurus, after applying stemming that a word can adopt and three cosine functions ($\sqrt{\cos\theta}$, $\cos\theta$, $\cos^2\theta$) to improve the performance of Genetic Algorithm [2]. José and Santesmases has experimented with a set of Spanish words to evaluate for a Query which has shown the precision improvement (.55 - .68, 19.11% of improvement) [1].

Many authors has described the use of web crawler to extract the keyword from the database to count the frequency of these keyword which then used for the combinations of the keywords to retrieve the desired documents [3]. Jing and Robert have a vector space model in which Queries are represented as vectors and both weighted and un-weighted can be used [2] [4]. But the weighted and un-weighted keywords have profound effects while retrieval of information [3] [4]. The studies are needed for having a dynamic Query window size which increases with the Query length [4]. In paper [5] authors have used Genetic adaption of matching functions to rank documents for weights and improve retrieval performance. Bangorn and Ouen [7] have presented an approach to represent chromosomes in a document vector with n keywords for online information retrieval. The study needs more advancement while working with large collection of documents (high Recall and high Precision) [5] [7].

The authors of paper [9] have described Adaptive Genetic Algorithm (AGA) with different fitness functions and variable operator rates on vector space model, extended model and language model for improved information retrieval. The study proved that the vector model with cosine as fitness function provides the higher performance [1] [9].

5. PROPOSED MODEL:

The work has been done on various Genetic adaption functions to provide different approaches that has successfully solved for query reformulation and optimization to match the documents in the provided database. They have appreciably, provided improved information retrieval results. The algorithms that have worked over large databases were prone to more revisions. Moreover, query reformulation, matching the documents using Genetic Algorithms, document representation were most focused concepts, even for the further extended improvements or revisions.In this proposed model, the searching concepts of Data Structures (Sequential search and Binary search) are introduced with the initial stages of Genetic Algorithm as shown in flowchart figure – 5. By using the resultant arrays obtained from these simultaneous searching threads along with the fitness of each array document will account for query formulation. It provides an array of most probable indexed documents as array elements that could participate, to evolve an optimized and most relevant document in few generations of Genetic Algorithm. Hence this improves the efficiency of a traditional Genetic Algorithm and provides an approach to be use for Information Retrieval from large collection of database.

5.1 GENETIC ALGORITHM USING DS SEARCHING MECHANISMS:

PROPOSED APPROACH:

The proposed approach has used searching concepts of Data Structure to evaluate the initial population and evaluation function stages, and also provided an idea for calculating the fitness function based on the user query. The model will search for the most relevant array of documents that have the tendency to participate and evolve out with an optimum and desired solution, provided that the array list of named index database in some sorted order. The use of this mechanism filters out, the best initial population of participants chromosomes from the week ones to evolve out with the required solution after processing in few generations. Hence, the proposed approach will improve the efficiency of Genetic Algorithm, even for larger search space. The detailed proposed algorithm has described in figure-5.1.

5.1.1 THREADING - THE POPULATION GENERATION AND EVALUATION FUNCTION:

Threading is the most common and important concept of any advanced computer programming language which has made many tasks to be accomplish, simultaneously. In this approach, using the two searching threads namely sequential searching thread and binary searching thread can bind the initial two stages to work simultaneously in order to improve the effectiveness and efficiency of Genetic Algorithm.

(A) BINARY SEARCHING THREAD:

In this approach, each visited element (document) will be stored in an array named as **arr[b]** – array of documents obtained during

B_Search (binary search) arr[b] lists the documents in an ascending order of their fitness values to formulate the user query.

PROPOSED ALGORITHM

Generate the initial population (array of documents) and evaluate each array element (document) by analyzing fitness function (F = Eq - Ei) for all (0<F<1)

- **Sequential Searching thread**
  - **Full search over array**
  - **Evaluating Fitness values for each array element (document)**
  - **Searched array arr[s]**
- **Binary Searching thread**
  - **Generation of initial population**
  - **Searched array arr[b]**

(If search results does not matched)

Selection of two parent strings P1 and P2

Generate Offspring using One - Point Crossover

Mutate each generated String

FIGURE-5.1:

(B) SEQUENTIAL SEARCHING THREAD:

In this approach, sequential search performs evaluation function by visiting each element (document) at least once and thereby calculating the corresponding fitness of each element (document) with respect to the query element. Hence, the sequential searching provides an array named as,

arr[s] – array of documents obtained during L_Search (Linear or Sequential Search)

5.1.2 CHROMOSOMES FORMATION:

arr[b] is mapped to a[s] to evolve out with an array of most common set of documents to encode as chromosomes. Assume arr[s] = {A, S, N, T, L, F, U, R, K, M} arr[b] = {A, S, T, L, U, R}, then chromosomes can be encoded as

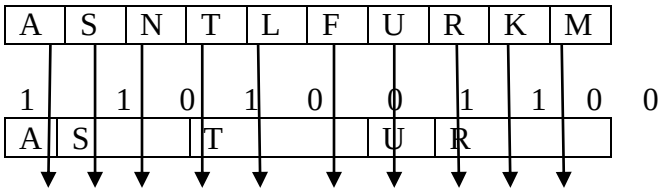


FIGURE-5.1.2

Hence, a chromosome string 1101001100 is obtained.

5.1.3 CALCULATION OF FITESS FUNCTION (F):

(a) Evaluation of each document

Analyzing each document from the indexed database and evaluating the corresponding function E, defined as E = {numeric value (Alphabets)} E is the set of numeric values of a string from the series of alphabets (1 - 26).

Words	Alphabets	Numeric codes
Data	DATA	4-1-20-1
Info	INFO	9-14-6-15
From	FROM	6-18-15-14
Dataset	DATASET	4-1-20-1-19-5-20

Table-5.1.3(a)

(b) FITNESS CALCULATION:

Fitness of every document is calculated by analyzing the distance of document index keyword to that of Query Keyword. The fitness of each document is calculated with respect to user query, provides a score to each document.

Hence, it can be defined as the Euclidean [4] distance between the evaluation function of query keyword [Eq] and the evaluation function of each indexed document keyword say [Ei]. Hence Fitness F of any document can be analyzed as:

$F = Eq - Ei \quad [0 < F < 1]$

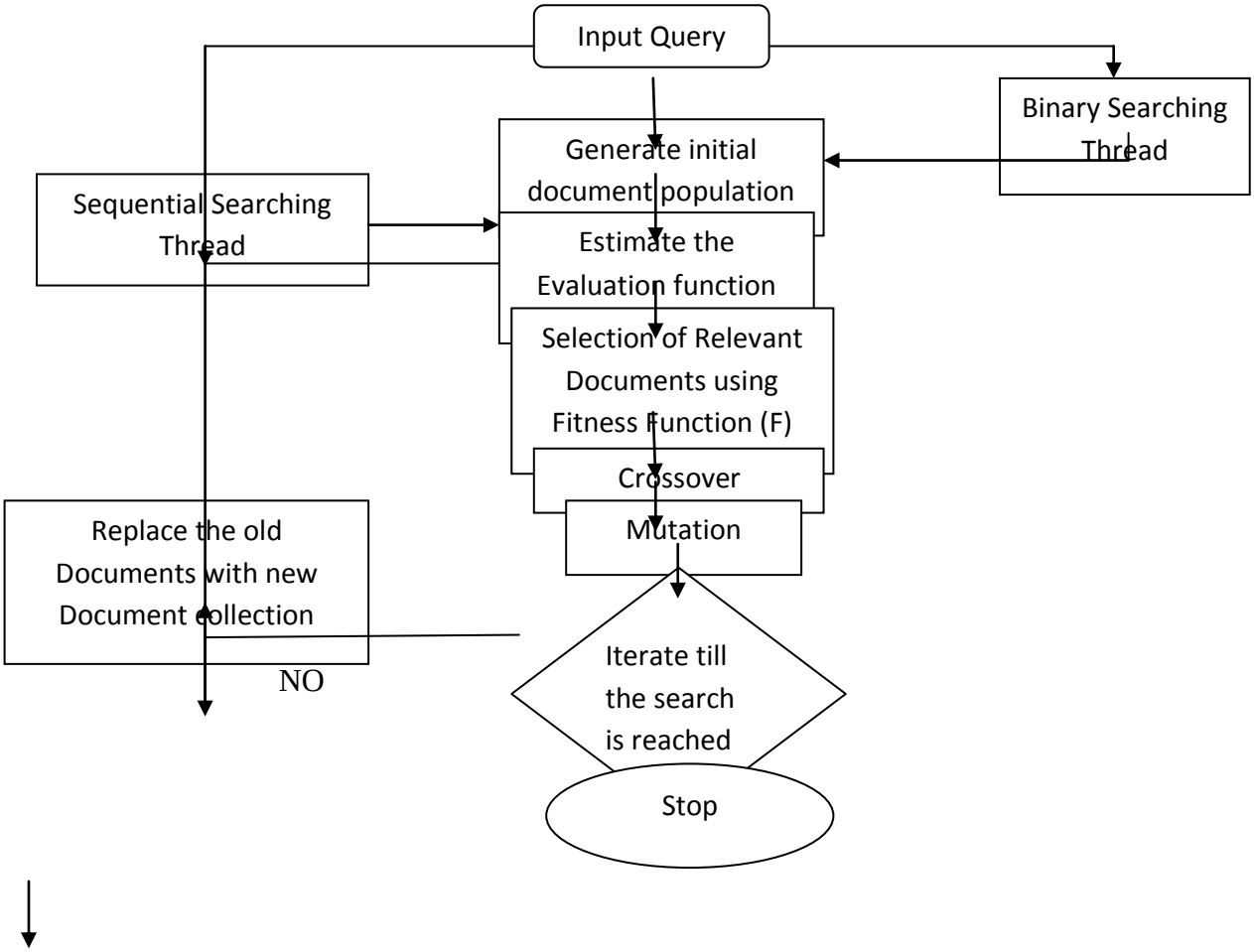
Let us take an example of Query Keyword as INDEX and hence finding the corresponding the fitness of each keyword enlisted in Table-5.1.3(a). The most relevant keyword will have the least fitness values.

Eq	Ei	$\|F = Eq - Ei\|$
Index	Data	0.87
Index	Info	0.19
Index	From	0.53
Index	Dataset	0.41

Table-5.1.3(

Hence, in a particular generation for matching INDEX keyword among document set {Data, Info, From, Dataset}, info will be the most fitted document with least fitness value as shown in Table-5.1.3(b).

FIGURE-5 FLOWCHART OF PROPOSED GENETIC ALGORITHM:



6. EXPERIMENTAL ANALYSIS:

The performance evaluation has used the Java API for Genetic Algorithm to provide a programming environment.

For a population size – 1000, using one – point crossover and mutation, best fitness is evaluated as:

Traditional GA	PROPOSED GA
GENERATION : 47	GENERATION : 11
Best Fitness : 0.20	Best Fitness : 0.14
Runtime : 00:28:28	Runtime - 00.04:44
Standard Deviation : 155.886	Standard Deviation : 47.809

Table – 6

The Table – 6 shows the efficiency of the proposed model with respect to the best for the relevant document over few generations, from a large collection of database in smaller time than a traditional Genetic Algorithm

for relevant document access time. Hence, binding the first two steps of a simple Genetic Algorithm saves the time and account for providing results in lesser generations.

Below two figures 6(a) and 6(b) provides a view containing the solutions for same number of chromosomes. The first figure 6(a) uses a traditional Genetic Algorithm to evolve out with the document of best fitness in a particular generation.

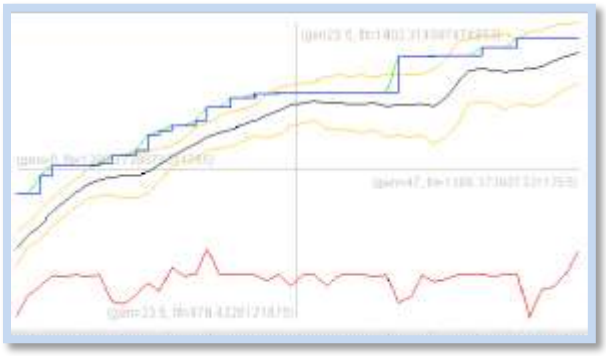
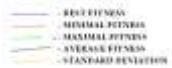


FIGURE – 6(a) PERFORMANCE ANALYSIS USING TRADITIONAL GENETIC ALGORITHM



GEN : 47, FITNESS : 0.20

While, the second figure 6(b) uses the proposed approach to calculate the best fitness in few generations.

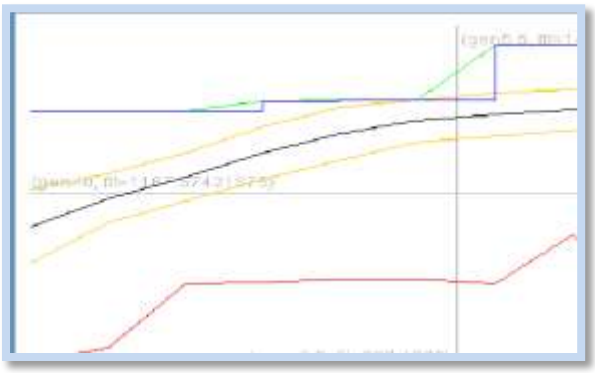
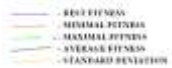


FIGURE – 6(b) PERFORMANCE ANALYSIS USING PROPOSED APPROACH



GEN : 11, FITNESS : 0.14

7. RESULTS AND DISCUSSIONS:

The paper has described a model to improve the efficiency of Genetic Algorithms for an effective Information Retrieval from large database repositories. Recent works were more focused on improving the information retrieval using adaptive methods for Genetic Algorithms has needed more study for Information Retrieval from large collection of database.

The proposed approach has shown the improvement in the fitness values resulted in few generations for large database. The further study still requires more calculations based on the evaluation measurements for Information Retrieval performance.

The idea of using Data Structure searching concepts to improve the performance of Genetic Algorithms, has particularly applied for Information Retrieval, can be extended for other research applications that uses Genetic Algorithms.

The use of binary search provides a list of documents to formulate the query as per user needs. The fitness value of every document, based on the alphabetic organization of keywords is matched against the user query. The distance between the fitness values of every document in the database from the user query provides a score to each document that will be counted as the rank or weight as per user requirements (queries).

The future work on this paper needs a method for having dynamic window for addition or insertion of new terms to the user query. The work also requires more revisions on different sized database repositories and recorded evaluation of the IR system performance.

8. CONCLUSION:

This paper provides a survey based approach to use the Data Structures searching techniques to improve the search optimization of a traditional Genetic Algorithm. The proposed algorithm can work on large collection of stored database repositories to evaluate out, with a subset collection of most probable documents, followed by working out with Genetic Algorithm.

REFERENCES:

1. José R. Pérez – Agüera and Jose García Santesmases, "Using Genetic Algorithms for Query Reformulation", BCS IRSG Symposium, Future Directions in Information Access (FDIA 2007).
2. Anubha Jain, Swati V. Chande and Preeti Tiwari, "Relevance of Genetic Algorithm Strategies in Query Optimization in Information Retrieval," International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5(4), 2014, pp. 5921 – 5927.
3. S.Siva Sathya and Philomina Simon, "A Document Retrieval System with Combination Terms Using Genetic Algorithm," International Journal of Computer and Electrical Engineering, Vol. 2, No. 1, Feb. 2010, pp. 1793-8163.
4. Jing-Jye Yang, Robert R. Korfhage and Edie Rasmussen, "Query Improvement In Information Retrieval Using Genetic Algorithms-A Report on the Experiments of the TREC Project," pp. 31-58.
5. Praveen Pathak, Michael Gordon and Weiguo Fan, "Effective Information Retrieval using Genetic Algorithms based Matching Function Adaption," Proceedings of the 33rd Hawai International Conference of System Sciences-2000, pp. 1-8.
6. http://en.m.wikipedia.org/wiki/Web_Search_Engine
7. Bangorn Klabbankoh and Ouen Pinngern, "Applied Genetic Algorithms In Information Retrieval".
8. http://en.m.wikipedia.org/wiki/Information_Retrieval
9. Wafa Maitah, Mamoun Al-Rababaa and Ghasan Kannan, "Improving the Effectiveness of Information Retrieval System Using Adaptive Genetic Algorithm," International Journal of Computer Science and Information Technology (IJCSIT), Vol. 5, No. 5, Oct 2013 pp. 91-105.
10. A Seventh Edition of Textbook for class XII "Computer Science with C++"by Sumita Arora based on CBSE, January 2010.