Optimizing Search Engine Result using an Intelligent Model

Hamada M. Zahera
Menoufiya University
Cairo, EGYPT

Gamal F. El Haddy
Menoufiya University
Cairo, EGYPT

Arabi E. Keshk
Menoufiya University
Cairo, EGYPT

ABSTRACT
Many search engine users face problems while retrieving their required Information. For example, a user may find it is difficult to retrieve sufficient relevant information because he use too few keywords to search or the user is inexperienced and do not search using proper keywords and the search engine is not able to receive the user real meaning through his given keywords. Also, due to the recent improvements of search engines and the rapid growth of the web, the search engines return a huge number of web pages, and then the user may take long time to look at all of these pages to find his needed information. The problem of obtaining relevant results in web searching has been tackled by several approaches. Although very effective techniques are currently used by the most popular search engines, but no a priori knowledge on the user's desires beside the search keywords is available. In this paper, we present an approach for optimizing the search engine results using artificial intelligence techniques such as document clustering and genetic algorithm to provide the user with the most relevant pages to the search query. The proposed method uses the Meta-data that is coming from the user preferences or the search engine query log files. These data is important to find the most related information to the user while searching the web. Finally, the method implementation and some of the experimental results are presented with the conclusion of this research study.

Keywords
Search Engines; Information Retrieval; World Wide Web; Document Clustering; Genetic Algorithm

1. INTRODUCTION
Today, after the rapid growth of the Internet as more and more data are added to the World Wide Web (WWW), anyone can easily access information from the Internet. In the meantime, although it's now straightforward to obtain the information needed from the internet, the rapid growth of the WWW makes the problem of information overload [1, 2, and 8]. In the last few years, different types of search engines such as Google, Yahoo and Bing have been developed to help the users in finding their needed information easily. Search engines are useful tools to collect and index pages. After receiving the request that a user specified, the search engine uses an internal strategy to search for information on the Internet that match the user query and return the web-pages that includesuch user query [1]. The first problem is that the pages in the front may be very similar (or even equal) to each other , and the user may select one and ignore other pages however they have the same or have very close scoring values. The second one, when the pages that do not have a very low probability to be visited by the user. The proposed approach overcome these two problems by selecting a small subset from the search results which have high scores and semantically related to the user query which are different from each other and are chosen from different regions of some topics where the pages are represented. However, retrieving sufficient relevant information online is difficult for many people because they may not be familiar with the search context, they use too few keywords to search or use improper keywords, or search engines are not able to receive the users' real meaning through their given keywords [2].

Another problem faces many of search engine users is that due to the huge growth of WWW there are a lot of information available on the internet, when the search engine is searching for the required information, the search engine returns many of web pages related to the search query. The user does not have time to check all of the returned results, and then he can check a few numbers of these web pages and ignore the others. However his required information might be found in these pages. As a result, users cannot find the specific information they really want [3]. Accessing topical information through existing search engines requires the formulation of appropriate queries, which is highly challenging, and then the appropriate selection of query is an optimizing problem and the purpose is to obtain the best query to get the information through the web automatically [4]. Also such a way required in order to reduce this huge number of returned results so that the user can have the ability to check them for the required information and not neglects all of the returned pages. The idea behind this way comes from equivalence partitioning technique which is used to reduce the number of test cases.

In this paper, we discuss in details an approach to reduce the results in a short list by applying artificial intelligence techniques such as document clustering and genetic algorithm [5]. Document clustering is required to group all similar pages together into one partition (cluster) after that optimization of the results is applied using genetic algorithm to select from each cluster the best pages with high scores and other features like number of keywords. Finally the outcome of the genetic algorithm is the final shortlist of web pages that are chosen from different regions of information. Thus we have a reduced number of web pages that can be reviewed by the users in a short time.

The outline of the paper is as follows. Section 2, reviews the related work in the area of Information retrieval refinements and search engine results improvements using artificial intelligence techniques. Section 3, discusses in details the proposed method of optimizing search engine results. In Section 4, shows the experimental results of applying the proposed method. Finally the conclusion of this research study and future work are presented in Section 5.

2. RELATED WORK

There are several research studies have been worked in the improvement of information retrieval results by employing clustering techniques and other intelligent approaches. In these studies the strategy was to build a clustering of the entire document collection and then match the query to the cluster centroids. More recently, clustering has been used for helping the user in browsing a collection of documents or in organizing the results returned by a search engine [6], or by a meta-search engine in response to a user query. As discussed in [6], the use of clustering in information retrieval (IR) is based mostly on the cluster hypothesis: "closely associated documents tend to be relevant to the same request". Several researchers have shown that the cluster hypothesis also holds in a retrieved set of documents, but they do not study how the clustering structure may help a user in finding relevant results more quickly. Metaheuristics, and more precisely, genetic algorithms, have been implemented in IR by several researchers and the results indicate that these algorithms could be efficient. In [7] the authors have discussed a new way of combining the clustering and genetic optimization in improving the retrieval of search engine results in different settings it is conceivable to design search methods that operate on a thematic database of web pages that refer to a common body of knowledge or to specific sets of users. They have considered such premises to design and develop a search method that deploys data mining and optimization techniques to provide a more significant and restricted set of pages as the final result of a user search. They adopt a vectorization method based on search context and user profile to apply clustering techniques that are then refined by a specially designed genetic algorithm.

In [8] the authors investigate the use of genetic algorithms in information retrieval. The method is shown to be applicable to three well-known documents collections, where more relevant documents are presented to users in the genetic modification. Gordon [9] presents a genetic algorithm based approach to improve document indexing. In that approach, the initial population is represented by a collection of documents judged relevant by a user, which is then evolved through generations and converges to an optimal population with a set of keywords which best describe the documents. In [10] the same author adopts a similar approach to document clustering, where a genetic algorithm is used to adapt subject descriptions so that documents become more effective in matching relevant queries. In [11] the authors apply genetic algorithm in information retrieval in order to improve search queries that produce better results according to users' preferences. In [12] Al-Dallal*et. al* proposed a text mining approach to web document retrieval that uses the tag information of HTML documents that Genetic algorithm is applied to find significant documents. In [13] Zhongzhi Shi *et. al* discussed the existing techniques for Web mining, which is moving the World Wide Web toward a more useful environment in which users can quickly and easily find the information they need. In [14] Eugene Agichtein*et. al* showed that incorporating user behavior data can significantly improve ordering of top results in real web search setting. They examined the alternatives for incorporating feedback into the ranking process and explore the contributions of user feedback compared to other common web search features.

In [15] YaJun Du *et. al* proposed an intelligent model and it's implementation of search engine that the process of searching information on Internet is similar as book search. And so, they proposed that Search Engines take on the five intelligence behaviors corresponding five parts intelligence of humankind: the apperceiving behavior, the memory behavior, the learning behavior, the thought behavior, the comprehension behavior. They divided the process of information searching of searchengine into four stages: classifying Web page, confirming a scope of information searching, crawling Web pages in internet, and filtrating the result Web pages. In [11] Fatemeh*et. al* have presented a method using genetic algorithm in a distributed way according to users' favorites to optimize query sent to search engine and finally to optimize quality of result pages.

3. METHOD DESCRIPTION

The standard results of the search engine and single web page are represented as P and p respectively while each page $p \in P$ is associated with a score based on the search query that generated P .The page score is used to order the search results decreasingly before displaying them to the user. The order of the page plays an important role in finding the required information by search engine, the probability that the user considers a page p strongly decreases as the position of p in the order increases. According to the current ranking mechanism of search engines, it will lead to two major problems that have been discussed above in the previous section. The proposed approach must be provided with Meta information from the user about the search context, in order to have promising results. This required information could be available by one of the following two ways:

• A search context which is a common topic to which the search query will be related, that is not necessarily linked with the search keywords that generated the set P; it may be viewed as an item of a topic catalog or directory as the ones that are frequently provided by the last generation of search engines (i.e., catalogs)

• A user profile of his choice which will be a subjective identification of the user trends , likeness and preferences, or extracted from the pages that have been visited more recently by that the user.

Basically, the idea of this method is to use the additional information comes from the search context or the user profile to have more analysed structure of set *P* and get a small subset of search results which are more precise and related to the search query. This is done into three additional steps to the standard search engine structure to have an intelligent search engine which are shown blew in Figure.1

First, both of the search context and the user profile are used to specify a finite list of significant words or page attributes from all pages in P, this list is used to create a vector of characteristics for each page which is called page vectorization, second, the vectorized pages are clustered into similar groups of similar pages called clusters, finally in the third step, the objective is to provide the user with small subset of the search results, considering the structure identified by the clusters and the score of pages, therefore genetic algorithm is applied for optimization and selecting the best set of pages from each cluster to obtain the best of the best search results as a final results displayed to the user.

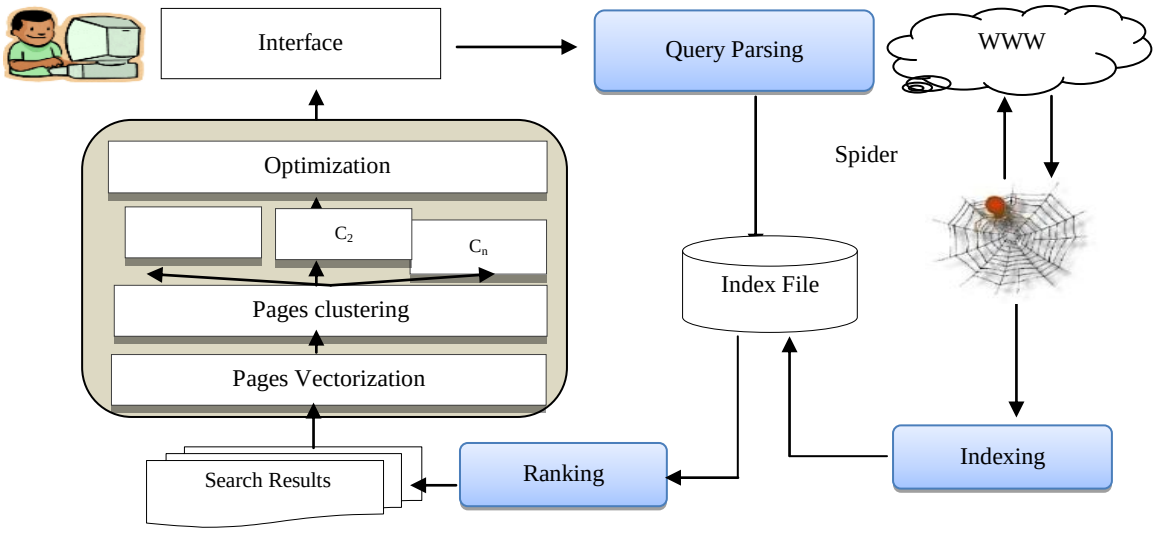

Fig 1: Block Diagram of Intelligent Search Engine

The final results quality depends on the way of how the genetic algorithm is implemented; that is, how the fitness of each chromosome is evaluated and how chromosomes are selected and combined in each iteration. The main idea is that, when properly designed, the genetic algorithm can determine chromosomes that are heterogeneous enough and whose pages have good values for the original score, the details of genetic algorithm are discussed in details in section .3.3

3.1 Page Vectorization

At first, each page is viewed as a vector in m-dimensional document space (where m is the number of distinguishing terms used to describe contents of the pages in a collection) and each term represents one dimension in the vector space model [16]. We can also taking into account other measurable characteristics that are not specifically linked with the words that are contained in the page, such as the presence of pictures, tables, banners and so on. As mentioned above, the vectorization is based on search context, or user profile which is chosen by the user. One may then assume that, for each of the contexts/profiles in the search engine, a list of words that are relevant to that context/profile is available and a related vectorization of the page is stored. Obviously, many enhancements to this simple approach may be considered. . First, it should not just consider simple "words", but also sets of words (synonyms, singular/plurals, etc.) which all contribute to the occurrence count stored in one component of the vector. The vector dimension is not theoretically restricted to be particularly small, but in order to apply the above method over a significant number of pages is it reasonable to consider m =100.Various methods have been used to identify the list of m words that are associated with a context/profile that can be divided into two main groups:

1. The list could be created according to the user knowledge as a user-defined list;

2. The list could be automatically extracted of given sets of pages

In the first case, the predefined words are determined in a configuration phase of the intelligent search engine, while the managers determine which are the contexts/profiles supported and what are the words that are representative of that context/profile. The application of this case may be done together with the users when this search engine is dedicated to a specific environment (such as an association of companies, a large corporation, a community of users).In the second case, words are specified starting from an initial set of pages that are used as training sample for a context/profile. From the given set of pages the words of high occurrence are extracted skipping articles, verbs, etc.

3.2 Clustering the vectorized pages

Clustering has become an increasingly important task in modern application domains. Clustering techniques have been applied to categorize documents on Web and extracting knowledge from the Web. Web page clustering methods categorize and organize search results into semantically meaningful clusters that assist users with search refinement [6]. But finding clusters that are semantically meaningful to users is difficult. The Document clustering in information retrieval is usually processed by agglomerative hierarchical clustering algorithms or k-means algorithm. K- Mean is the algorithm used in the clustering phase because it is simple and more appreciate for fast document clustering compared with other algorithms for document clustering [17]. The procedure of k-mean algorithm follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. Finally , all vectorized pages from the previous phase have been partitioned into similar clusters according to their similarity in order to apply the last phase of the proposed method that optimize these results using genetic algorithm which is discussed in details in the preceding section

3.3 Optimization of the Search Results

Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination and mutation operators to these structures so as to preserve critical information. An implementation of a genetic algorithm begins with a population of chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to reproduce than those chromosomes which are poorer solutions [18]. The goodness of a solution is typically defined with respect to the current population. Usually there are only two main components of genetic algorithms that are problem dependent:

the problem description and the fitness function (objective function / evaluation function) which will be described in a later section in details.

GA algorithm is selected in the application of the proposed method because of the following reasons: first the use of Meta-heuristics techniques is well established in optimization problems where objective function and the constraints do not have a simple mathematical formulation [8]. Second, we have to determine a good solution in a small computing time where the dimension of the problem may be large. Third, the structure of our problem is straightforward representable by the data structure used by genetic algorithm commonly [7].

3.3.1 Clustering the vectorized pages

Selection: Individual strings are copied according to their objective function (Fitness Function) value. This represents a measure of the utility or goodness related to what we want to maximize. Copying strings according to their fitness function values means that strings with a high value have a higher probability of contribution to one or more offspring in the next generation.

Crossover: The second operator is the genetic operator that combines (mixes) two chromosomes (parents) together to produce a new chromosomes (offspring).The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. GA constructs a better solution by mixture good characteristic of chromosome together. Higher fitness chromosome has an opportunity to be selected more than lower ones, so good solution always alive to the next generation. We use a single point crossover, exchanges the weights of sub-vector between two chromosomes, which are candidate for this process. In order to do this, an integer position (cutting point) is selected uniformly at a random along the chromosomes. Splitting the two selected strings at this point generates left and right parts. An offspring can then be produced for example by joining the left part of the first chromosome with the right part of the second chromosome, thus we have two generated offsprings as shown in figure.2

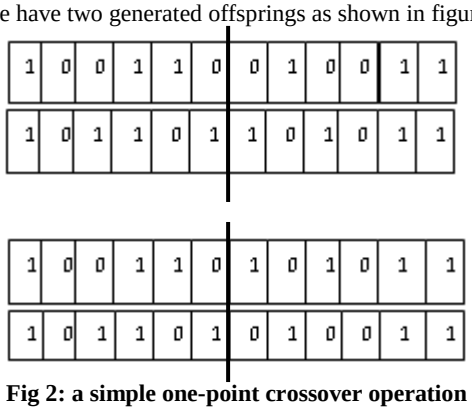

Fig 2: a simple one-point crossover operation

3.3.2 Chromosome Encoding

Each chromosome represents a solution to the problem and is composed of a string of genes in which each gene represents a page. The binary alphabet {0, 1} is often used to represent these genes but sometimes, depending on the application, integers or real numbers are used. In our application of genetic algorithm, each page represent as 0 or 1 in the chromosome solution representation which 0, 1 mean the absence/ presence of the pages in that solution. We indicate with *dc* the number of pages included in each chromosome in the initial population, and with *nc* the number of chromosomes. The population dimension will thus contain $np = dc \cdot nc$ pages. Here follows a simple example, We have a set of ten pages {*P1,P2, P3, P4, P5, P6, P7, P8, P9, P10*} and these pages were clustered into two clusters: the first cluster has *P1, P3, P5, P8, P9*, while the second cluster has P2, *P6, P4, P7, P10* are in the second and third cluster respectively. Assuming the ranking or the pages according to their scores only is {*P4, P7, P3, P2, P5, P8, P10, P6, P9, P1*}.

The initial population of the genetic algorithm is constructed by selecting the pages with high scores from each cluster. Each chromosome is created by picking up a page from , starting with the pages with higher score .Thus , the first chromosome created will the pages with highest scores in each cluster , the second chromosome will have the pages with the best score in each cluster and so on as shown in figure.3. At some cases, the number of pages in each clusters will be different, therefore these clusters will not be represented in each chromosome while other clusters with large number of pages may have more than one page representing them in some chromosomes.

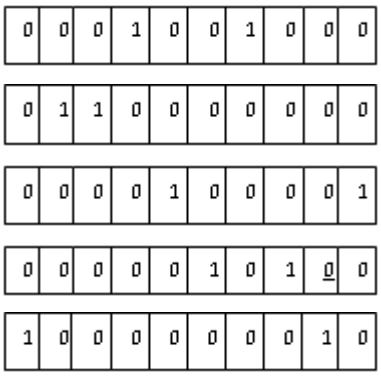

Fig 3: Initial Population of the five chromosomes

3.3.3 Fitness Function

The fitness function is used to evaluate the goodness (validness) of each chromosome, how it will be valid to be used for creating the new generations of offspring. For each chromosome fitness value is computed as a positive value that the higher fitness value for the better chromosome, and is thus to be maximized [20, 21]. It is composed of four terms. The first term $T_1(C)$ is the sum of the score of the pages in chromosome C

$$T_1(C) = \sum_{p_i \in C} score(p_i) \qquad (1)$$

where $score(p_i)$ is the original score given to page pi which considers the positive effect of obtaining any pages with high score as possible in a chromosome, but would also reward chromosomes with many pages regardless of their score (a chromosome with many pages with low score could produce a higher fitness of another with few good pages). This drawback is fixed by adding the second term $T_2(C)$ that penalizes the distance of the dimension of a chromosome from an ideal dimension. Let ID be such ideal dimension;

$$T_2(c) = \frac{np}{abs\,(|C| - ID) + 1} \qquad (2)$$

Wherenp is the maximum value of ratio that will be reached when the dimension of C is exactly equal to the ideal dimensionID, and decreases when the number of pages contained in chromosome C is smaller or greater than ID. The chromosomes that are present in the initial population are

characterized by the highest possible variability as far as the Although, the evolution of the population may alter this characteristic, creating chromosomes with high fitness where the pages belong to the same cluster and are very similar to each other. Moreover, the fact that pages belonging to different clusters are different in the vectorized space may not be guaranteed, as it depends both on the nature of the data and on the quality of the initial clustering process. For this reason, the fitness function has a third term measures directly the overall dissimilarity (distance) of the pages in the chromosome. The distance between pages in the chromosomes is computed by Euclidian distance that is more appreciate for document retrieval [22]. Let$ID(\vec{p}_i,\vec{p}_j)$be the distance of the vectors representing pages and as described above in the page vectorization section.

$$T_3(C) = \sum_{p_i,p_j \in C, p_i \neq p_j} D(\vec{p}_i,\vec{p}_j) \quad (3)$$

$$ID(\vec{p}_i,\vec{p}_j) = \sqrt{\sum_i^m (p_i - p_j)^2} \quad (4)$$

A fourth term $T_4(C)$ should be added to the fitness function in order to have more precision fitness function that has a better evaluation of the chromosomes. The fourth term calculates the total number of keywords in the chromosomes. The more keywords are existed in the chromosome, thus mean this chromosome is more precision and relative to the search context / search profile

$$T_4(C) = \sum_{p_i \in C}^{n} kw(p_i) \quad (5)$$

Where $kw(p_i)$ means the number of keywords of page i in the chromosome C, n is the number of pages in the chromosome. Finally, compute the fitness value of the chromosome by summing all of these terms together. The higher fitness value of the chromosome, the better chromosome existed. Our goal is to find the chromosome with the maximum fitness value after various numbers of iterations to generate new chromosomes. The final form of the fitness function for chromosome C is:

$$ff(C) = \alpha.T_1(C) + \beta.T_2(C) + \gamma.T_3(C) + \delta.T_4(C) \quad (6)$$

Whereα,β ,$\gamma$and δ are constant parameters which depend on the size of the initial score and of the vectors that are represented the pages. Particularlyα,β ,$\gamma$and δ are chosen so as so as the contributions given by $T1(C), T2(C), T3(C)$ and $T4(C)$ are balanced.

4. EXPERIMENTAL RESULTS

The proposed model of optimization the search engine results as shortlist results is developed in the Java language .The choice of java implementation is for many reasons like the model will be easily integrated with open source search engines which use Lucene or Solr API, the code will be open source of the other developers and researchers so they can extend and add their own ideas and the code will be free and under GPL software license. We have run number of experiments with the method described above under Windows platform. The process of the solution described below took at most few seconds the instances using a PC with 2.20 GHz processor and 2GB RAM

clusters to which the pages belong are concerned.

The overall behaviour of the method has been tested with two different page sets: The first set of pages has (500 web pages, 20 Keywords and 5 clusters) and the second has (1000 web pages, 40 keywords, 6 clusters).The proposed method is applied by these two sets as described above in the method description section: the pages are vectorized according to the keywords then we have a vector of keyword frequency for every page then the clustering process is performed by k-mean clustering algorithm.

In case of page set 1: the pages are clustered into 5 clusters, thus we at least 100 pages in each cluster. The clustered pages are passed to the genetic algorithm in order to select the pages with higher scores to be displayed into the final list with max 40 pages as the chromosome length. In case of page set 2: the pages are clustered into 6 clusters, thus we at least 80 pages in each cluster. The clustered pages are passed to the genetic algorithm in order to select the pages with higher scores and displayed into the final list with max 40 pages as the chromosome length.

The table.1 shows the behaviour of genetic algorithm in different number of keywords and different size of page sets. The fitness function value increased gradually reaches its highest at the last iterations. The first column represents the number of pages, the second column represents the number of keywords, and the initial value of fitness function and the best value are presented in the third and fourth column respectively. The parameters of GA fitness function have been set to these values $\alpha = 1$, $\beta = 50$, γ =1 and $\delta = 1$

# Exp	# Pages	# Keywords	Fitness Value at Iteration 0	Best Fitness Value
1	500	20	1416.73	6286.2
2	500	20	1614.04	6379.1
3	500	20	1552.83	6613.6
4	500	40	1444.0	5426.0
5	500	40	1392.0	5652.0
6	500	40	1617.0	6072.0
7	1000	20	1398.0	6422.0
8	1000	20	1558.0	8855.0
9	1000	20	1576.0	6171.0
10	1000	40	1567.0	7011.0
11	1000	40	1480.0	8113.0
12	1000	40	1479.0	7369.0

.

The overall performance of the proposed model is evaluated using the two metric of information retrieval [16, 22] performance evaluation: Recall and precision. Precision measures the retrieval accuracy while recall measures the ability of retrieving relevant items from the whole data sets

$$Recall = \frac{\text{number of relevant retrieved pages}}{\text{total number of relevant pages}} \quad (7)$$

$$Precision = \frac{\text{number of relevant retrieved pages}}{\text{total number of retrieved pages}} \quad (8)$$

Figure.4 shows the recall evaluation of chromosomes A, B, C and D which represents the final solution .it can be clearly seen that these chromosomes have big ratio of relevant pages to the search context. The relevant ratio of the page is the average between the original score and the ratio of keywords inside this page. For example, chromosome A has 56 % relevant pages and 44 % irrelevant pages. Both of pages are important to be returned to the user because at some cases the needed information could be found in the pages with low scores, therefore they should not be ignored or neglected at all

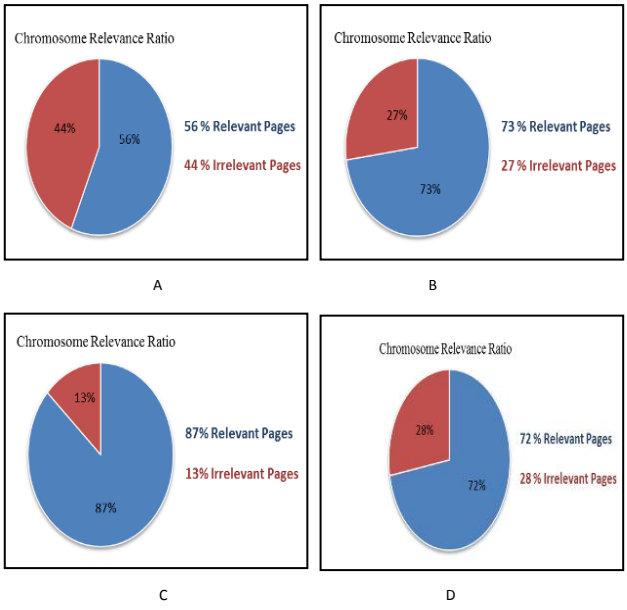

Fig4: Recall Evaluation of Chromosomes

The precision between the two fitness functions are computed according to the equation (8) and compared as it shown in figure (6), (7) .it's clearly seen that precision of *ff2* is higher than in (ff_1) in most cases that is because the forth term T_4(C).This Term plays an important role in evaluating the chromosome fitness with more precision because it considers the keywords inside the page beside its original score

In the other side, it's very important to compare the running time of the proposed method with the refinements and the running time of the original method. The comparison between the two methods are shown in figure.7, as it shown the time of the two methods are very close to each other and in some cases they are the same. Therefore, we have a more precision method for optimizing the search engine results and better information retrieval with the same cost of time.

The fitness function ($ff2$) which is used in the proposed method is extended from the original fitness function ($ff1$) proposed by Caramia [7] to evaluate the chromosome fitness. If the page has a low score and a good number of keywords of search context, this page will not beconsidered in the final solution however this page is relevant. Therefore the precision of (ff_2) is greater than (*ff1)* because it considers more relevant pages in the final solution. The precision evaluation has been tested on larger sets of pages (page set 2) as it shown in figure.5 to assure that the precision retrieval did not depend on the size of pages, but it depends on the approach used to evaluate the chromosome fitness.

The precision between the two fitness functions are computed according to the equation (8) and compared as it shown in figure (6), (7) .it's clearly seen that precision of *ff2* is higher than in (ff_1) in most cases that is because the forth term T_4(C).This Term plays an important role in evaluating the chromosome fitness with more precision because it considers the keywords inside the page beside its original score

In the other side, it's very important to compare the running time of the proposed method with the refinements and the running time of the original method. The comparison between the two methods are shown in figure.7, as it shown the time of the two methods are very close to each other and in some cases they are the same. Therefore, we have a more precision method for optimizing the search engine results and better information retrieval with the same cost of time.

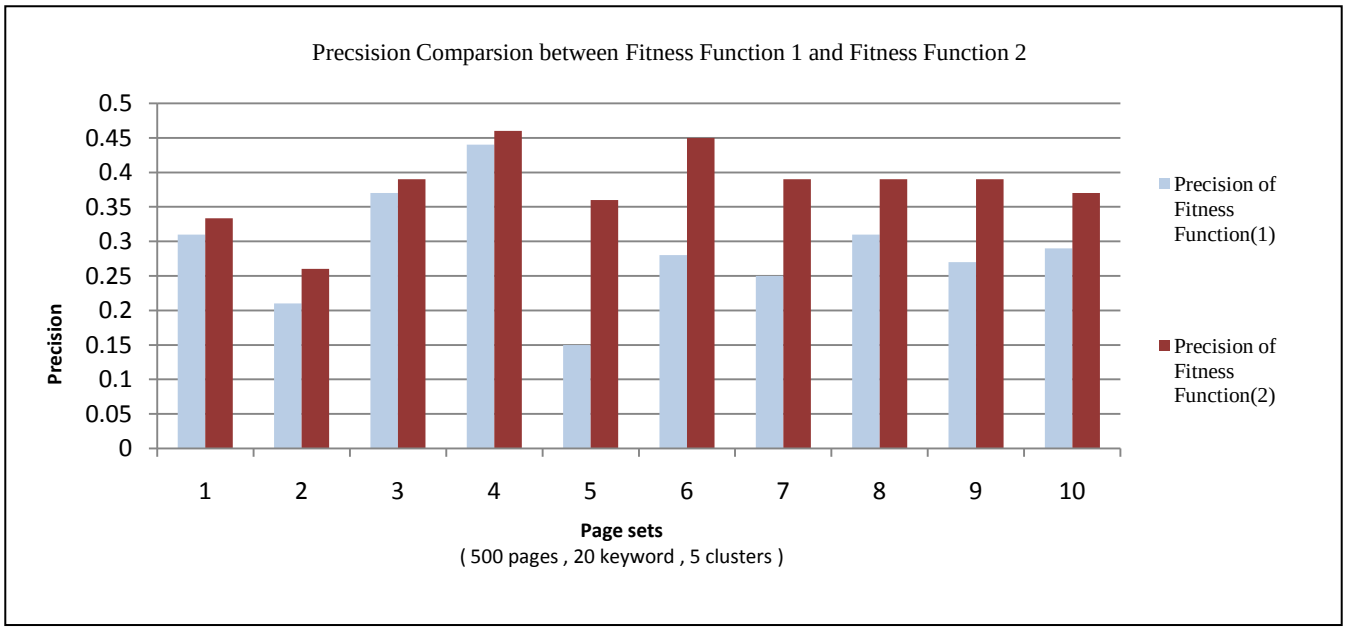

Fig.5 Comparison between Fitness Function1 and Fitness Function2 in Page set1

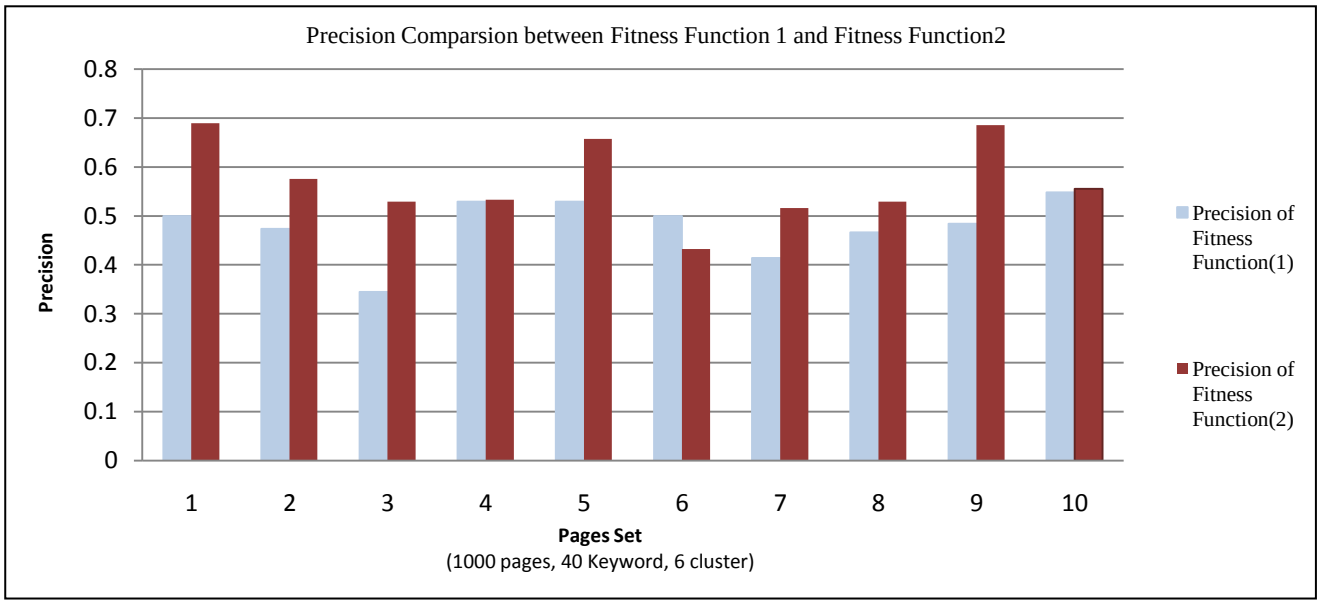

Fig.6: Comparison between Fitness Function1 and Fitness Function2 in Page set2

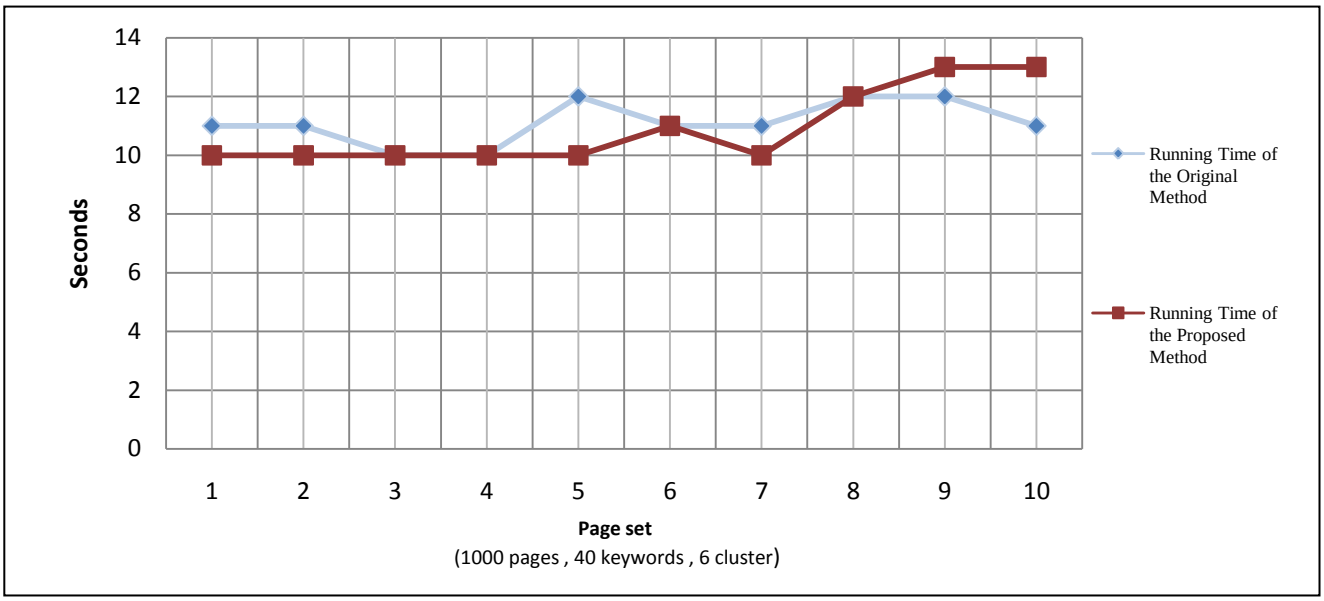

Fig5: Comparison between running time of proposed and original methods

5. CONCLUSION AND FUTURE WORK

The experimental results presented in this paper show that the proposed method with the recent modifications can be effective in the selection of small subsets of pages of good quality, where quality is not considered as a simple sum of the quality of each page but as a global characteristic of the subset. The implementation of the GA and of the clustering algorithm has allowed us to obtain convergence to solutions in reasonably short computational times on a standard personal computer (a few seconds). One may question whether the described method can be run on-line in a search engine as the standard execution of a user's query. We believe that with proper tuning on the parameters and a proper engineering of the algorithms, the computational effort can be dealt with and that only few seconds may be added to the overall search process.

The Future work will use Fuzzy C-Mean algorithm in document clustering as some pages can't be classified in one cluster only, they have the probability to join more than one clusters. Also it will replace the genetic algorithm by quantum genetic algorithm to overcome the problem of random evolutionary in genetic algorithm.

6. REFERENCES

[1] W.Lee and T.Tsai, "An interactive agent-based system for concept-based web search"," Expert Systems with Applications, vol. 24, pp. 365-373, 2003.

[2] R.Yates, "Information retrieval in the Web: beyond current search engines," International Journal of Approximate Reasoning, vol. 34, no. 3, November 2003.

[3] L.Chen, C.Luh, and C.Jou, "Generating page clippings from web search results using a dynamically terminated genetic algorithm," Elsevier Information Systems, vol. 30, pp. 299-316, 2005.

[4] R.L. Cecchini, C. M. Lorenzetti, A.G. Maguitman, and N.B. Brignole, "Using genetic algorithms to evolve a population of topical queries," Elsevier Information Processing and Management, vol. 44, pp. 1863-1878, 2008.

[5] A.Trotman, "An artificial intelligence approach to information retrieval," in Information Processing and Management, 2004 , pp. 619-632.

[6] A.Leuski, "Evaluating document clustering for interactive information retrieval.," in Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, 2001, pp. 33–44.

[7] M Caramia, G Felici, and A Pezzoli, "Improving search results with data mining in a thematic search engine," Computers & Operations Research, pp. 2387–2404, 2004.

[8] A. A. Radwan, B. A. Abdel Latef, A. A. Ali, and Osman A. Sadek, "Using Genetic Algorithm to Improve Information Retrieval Systems," in word academy of science, Engineering and Technology, 2006, p. 17.

[9] M Gorden, "Probabilistic and genetic algorithms in document retrieval," Communications of the ACM, vol. 31, no. 10, pp. 1208–8, October 1988.

[10] M.Gordon, "User-based document clustering by redescribing subject descriptions with a genetic algorithm," Journal of the American Society for Information Science, vol. 42, no. 5, pp. 311–22, 1991.

[11] F. Dashti and S. A. Zad, "Optimizing the data search results in web using Genetic Algorithm, "International Journal of Advanced Engineering Sciences and Technologies" vol. 1, no. 1, pp. 016 – 022, 2010.

[12] A. Al-Dallal and R.S. Abdul-Wahab, "Genetic Algorithm Based to Improve HTML Document Retrieval," in Developments in eSystems Engineering, Abu Dhabi , 2009, pp. 343 - 348.

[13] Z.S. Ma , and Q.He, "Web Mining: Extracting Knowledge from the World Wide Web," in Data Mining for Business Applications, Longbing Cao et al., Eds.: Springer, 2009, ch. 14, pp. 197-208.

[14] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, New York, 2006.

[15] Y. Du and H. Li, "An Intelligent Model and Its Implementation of Search Engine," Journal of Convergence Information Technology, vol. 3, no. 2, pp. 57-66, June 2008.

[16] G. Salton and M.H McGill, Introduction to Modern Information Retrieval., 1983.

[17] T. Huang, Y.Liaw and J..C. Lai, "A fast -means clustering algorithm using cluster center displacement," Pattern Recognition, vol. 42, no. 11, pp. 2551-2556, November 2009.

[18] F Picarougne, N Monmarchle, A. Oliver, and G. Venturini, "Web mining with a genetic algorithm," In Proceedings of the Eleventh International, 2002.

[19] W. Fan, P. Pathak, and Mi Zhou, "Genetic-based approaches in ranking function discovery and optimization in information retrieval — A framework," Decision Support Systems, vol. 47, no. 4, pp. 398–407, November 2009.

[20] W. Fan, E.A. Fox, P. Pathak, and H. Wu, "The effects of fitness functions on genetic programming-based ranking discovery for web search," Journal of the American Society for Information Science and Technology, vol. 55, pp. 628–636., 2004.

[21] W.Shengli, B.Yaxin, and Zeng, Xiaoqin, "Using the Euclidean Distance for Retrieval Evaluation," in Advances in Databases, Alvaro Fernandes, Alasdair Gray, and Khalid Belhajjame, Eds. Berlin / Heidelberg, German: Springer , 2011, pp. 83-96.

[22] D.Hawking, N.Craswell, P.Bailey, and K.Griffihs, "Measuring Search Engine Quality," Information Retrieval, vol. 4, no. 1, pp. 33-59, 2001.