Information Retrieval using Modified Genetic Algorithm

Alia Karim Abdul Hassan,Ph.D. (Asst. Prof)*,
Duaa Enteesha mhawi, (Asst. Lecturer)*
Sarah Najm Abdulwahid,(Asst.Lecturer)**

Abstract

Several techniques proposed to retrieve the most relevant HTML documents to user query. Genetic algorithm (GA) one of these technique, which creates several generations iteratively using the operations of: selection, crossover and mutation before producing the results. In this paper, focuses on enhance the operations of GA and proposed new fitness function to enhance the quality of the retrieved results. This technique applied to HTML documents and using recall, precision measures to evaluate. The results show high improvement in the retrieved documents quality in terms of these measures.

Keywords: Information Retrieval (IR), Genetic Algorithm (GA)

---

*University of Technology/Baghdad
** Ministry Of Higher Education &Scientific Research

1. Introduction

Amount of operational data has been increasing exponentially from past few decades, the expectations of data-user is changing proportionally as well. The data-user expects more deep, exact, and detailed results. Retrieval of relevant results is always affected by the pattern, how they are stored indexed. There are various techniques designed to index the documents, which done on the tokens identified with in documents, new techniques by using enhance inverted index. Information retrieval (IR) handles the representation, storage, organization, and access to information items [1]. In IR, one of the main problems is to determine which documents are relevant and which are not relevant to the user's needs. In practice, this problem usually mentioned as a ranking problem, which aims to solve according to the degree of relevance (matching) between all documents and the query of user which deals with information retrieval [1 - 3].Two main problems still encounter by the web user when trying to retrieve the document, which related to the user query, one of these is many of retrieved documents with highly ranked, which not related to the request of the user. The second problem is many related documents found in database but not retrieved [4 and 5].

The aim of thispaperisto develop a new Information Retrieval System based on using a modified GA, which has the ability to answer the user query with high performance in terms the four requirement factors (precision , recall , required storage space and , minimum response time).

Beside to the introduction, this paper contains the following main sections; section two presents some of the related work, while the proposed system is introduced in section three which consists of two stages and each stage is explained with figures, tables, algorithms and examples. The experimental results are illustrated in section four, finally, the conclusion is shown in section five.

2. Literature Survey

Several researchers have used evolutionary computation, especially GA for solving and reducing the problems of IR as much as possible.Most the researcher focuses their work on one or two of IR problems.

The most recent researches on applying GA to information retrieval are summarized as follow beginning from 2010 to 2014 and as following:

1. In 2010 L. Araujo and J. Perez-Iglesias, developed an IR used GA as training classifier to retrieve a set of terms; these terms are close to the user query to retrieve relevant documents to the user query. The performance of the developed system defined in terms of precision and recall, the precision is 68.7%, and the recall is 41.6% [15].
2. In 2011 L. Mohammad and E. Hanandeh, applied modified GA operators to IR by using method of vector space. System performance evaluated by applying two measures (recall, precision) to each operator of modified GA and on the results retrieved to the user query in term of average precision and average recall, so that average precision is 80%, and average recall 90% [16].
3. In 2012, A. Al-Dallal developed an IRS based modification of the GA operators to enhance the inverted index algorithm for documents indexing; can used term-proximity fitness function with hybrid crossover operator. The developed system precision is 89%, and developed system for recall 84%, with storage space 100MB for the used dataset of semi-structured documents. But still the response time is very long exceeding that in. However,the storage space to store this datasets need100MB,inaddition it cannot spendon two problem so IR,and does not remember anything about retrieved time of the related documents [17].
4. In 2013, W. Maitah1, M. Al-Rababaa and G. Kannan, improve the efficiency of IRS by using adaptive GA with method of vector support machine. In this work can be used modified crossover and mutation operators with different probability. The developed system precision measure is 89%, and recall measure is 80% [18].
5. In 2014, A. R. Abdel Late, and A. Mgeid, in this work applied GA to Information Retrieval System (IRS) with proposed new fitness function and applied ordinary crossover and mutation. Testing the results of this work by computing average precision-recall, and the result of this average is 80.3% [19].

3. Proposed system overview

In this paper, the proposed system for IR is based on using GA. Figure 1 shows a general view of the proposed system.

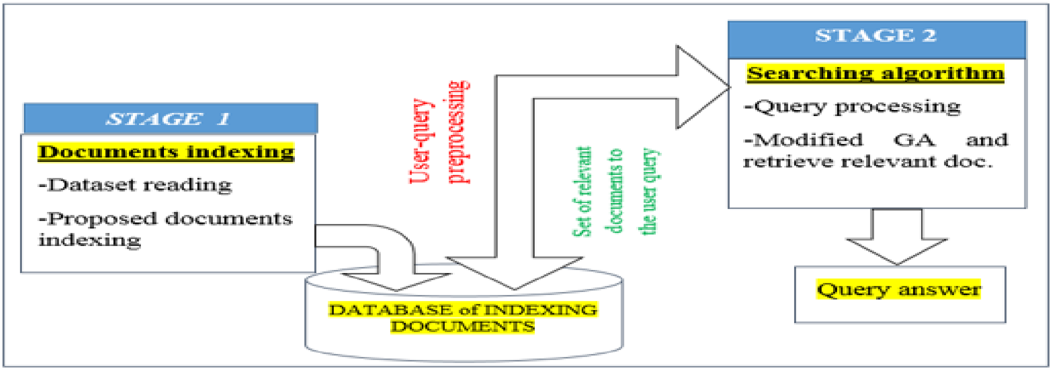


Figure 1:- Block-diagram of the Proposed IRS (general view)

Below is the explanation of the proposed system.

3.1 Documents indexing stage

This stag include read dataset documents and proposed documents indexing .the output of this stage is a set of tables which will be stored in database, words represents dataset with related index .the steps of this stage will be explained as below **:-**

a- Data set reading

The proposed system implemented using a free dataset for simulation purpose Web Kb [14] .This dataset containing World Wide WebPages. Dataset consists of 4582 semi-structured documents, written in Hypertext Mark-up Language (HTML), WebPages documents, which were manually classified into seven directories these directories, are basic materials, energy, financial, healthcare, technology, transportation and utilities. Inside each directory classes which represents contents of materials.Each of these tags reflects a specific level of importance within the document, as well as these tags contain essential information near to the term of the user query.

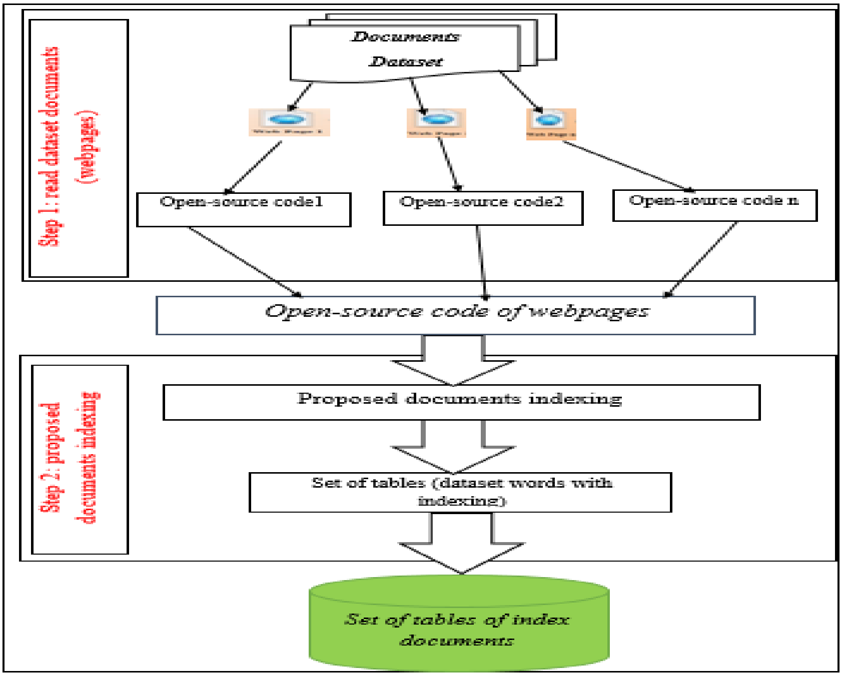
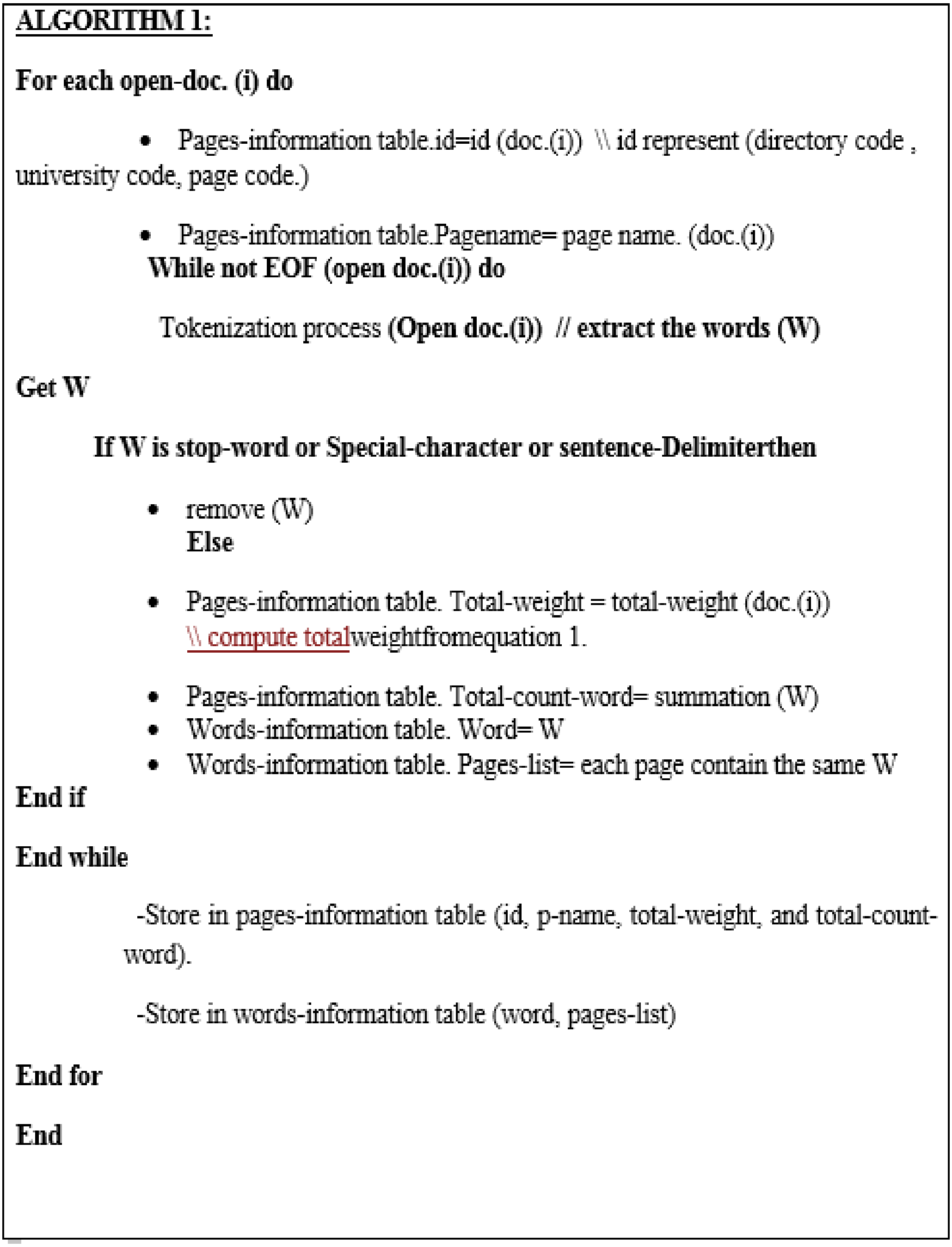


Figure 2: Block-Diagram (stage_1) of the proposed IRS

b- The Proposeddocuments indexing algorithm

In this paper, a proposed algorithm is used to index the documents called Proposed Documents Indexing. Algorithm1 describes this proposed algorithm of indexing.



```
ALGORITHM 1:
For each open-doc. (i) do
        • Pages-information table.id=id (doc.(i))  \\ id represent (directory code , university code, page code.)
        • Pages-information table.Pagename= page name. (doc.(i))
        While not EOF (open doc.(i)) do
            Tokenization process (Open doc.(i))  // extract the words (W)
Get W
    If W is stop-word or Special-character or sentence-Delimiterthen
            • remove (W)
            Else
            • Pages-information table. Total-weight = total-weight (doc.(i))
              \\ compute totalweightfromequation 1.
            • Pages-information table. Total-count-word= summation (W)
            • Words-information table. Word= W
            • Words-information table. Pages-list= each page contain the same W
End if
End while
            -Store in pages-information table (id, p-name, total-weight, and total-count-word).
            -Store in words-information table (word, pages-list)
End for
End
```

The proposed algorithm reduces the storage space of memory required during the query processing. Documents indexing receive the source-code of each document, and does pre-processing to generate two tables. First table is called pages-information table, which contains information about each page in term (id, p-name, words-count, Total-Tags-Weight). The second table is called words-information table, which contains information about each word in term (word, pages-list).

The proposed documents indexing process begins with the following process:

i- Special -Word table construction by doingStop-word removing, Special-characters removing and Sentence-delimiters removing process.

ii- Html tag information table Construction by remove unpreserved tags (html, Head, sub-headers (h1, h2, h3), body), and generate Wight of these tags remove.

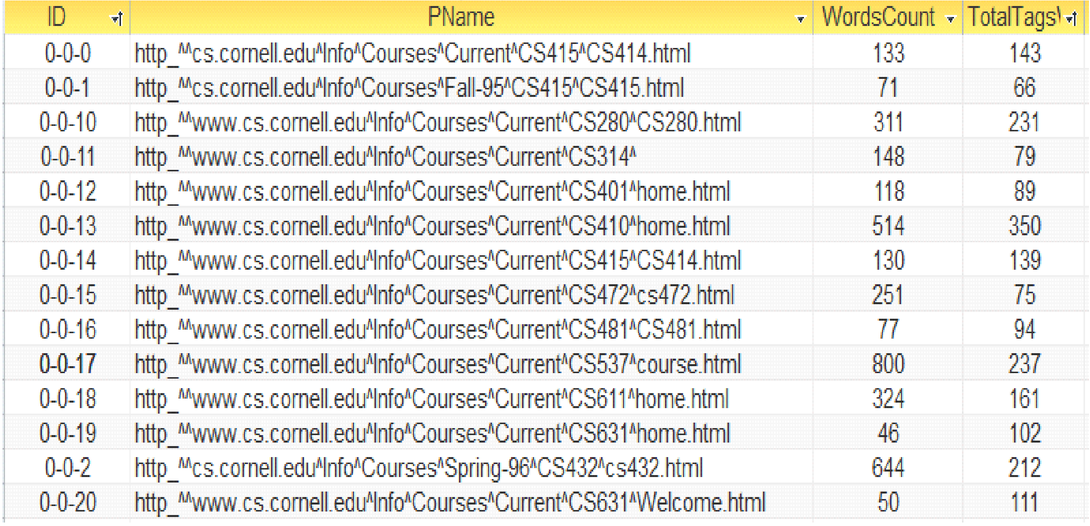
After these pre-processing is done the indexing process begins. Pages-information table generation is made by finding the total-tags-weight, using equation 1.

Total-tag-weight=Weight (W) tag+ …1

where: Weight (W) tag is the weight of word by using weight of tag

Table 1 shows an example of pages-information table.

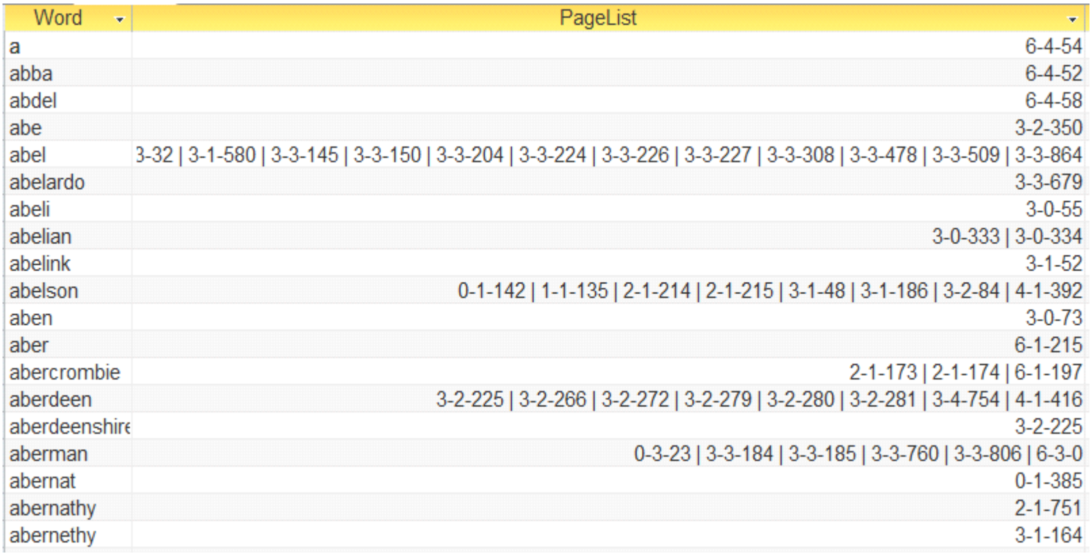
Table 1: Details and contents of pages-information table



ID	PName	WordsCount	TotalTags
0-0-0	http_^^cs.cornell.edu^Info^Courses^Current^CS415^CS414.html	133	143
0-0-1	http_^^cs.cornell.edu^Info^Courses^Fall-95^CS415^CS415.html	71	66
0-0-10	http_^^www.cs.cornell.edu^Info^Courses^Current^CS280^CS280.html	311	231
0-0-11	http_^^www.cs.cornell.edu^Info^Courses^Current^CS314^	148	79
0-0-12	http_^^www.cs.cornell.edu^Info^Courses^Current^CS401^home.html	118	89
0-0-13	http_^^www.cs.cornell.edu^Info^Courses^Current^CS410^home.html	514	350
0-0-14	http_^^www.cs.cornell.edu^Info^Courses^Current^CS415^CS414.html	130	139
0-0-15	http_^^www.cs.cornell.edu^Info^Courses^Current^CS472^cs472.html	251	75
0-0-16	http_^^www.cs.cornell.edu^Info^Courses^Current^CS481^CS481.html	77	94
0-0-17	http_^^www.cs.cornell.edu^Info^Courses^Current^CS537^course.html	800	237
0-0-18	http_^^www.cs.cornell.edu^Info^Courses^Current^CS611^home.html	324	161
0-0-19	http_^^www.cs.cornell.edu^Info^Courses^Current^CS631^home.html	46	102
0-0-2	http_^^cs.cornell.edu^Info^Courses^Spring-96^CS432^cs432.html	644	212
0-0-20	http_^^www.cs.cornell.edu^Info^Courses^Current^CS631^Welcome.html	50	111

The words-information table is constructed by first storing the word then finding the pages-list for this word, table 2 is an example of words-information table.

Table 2: Details and contents of words-information table



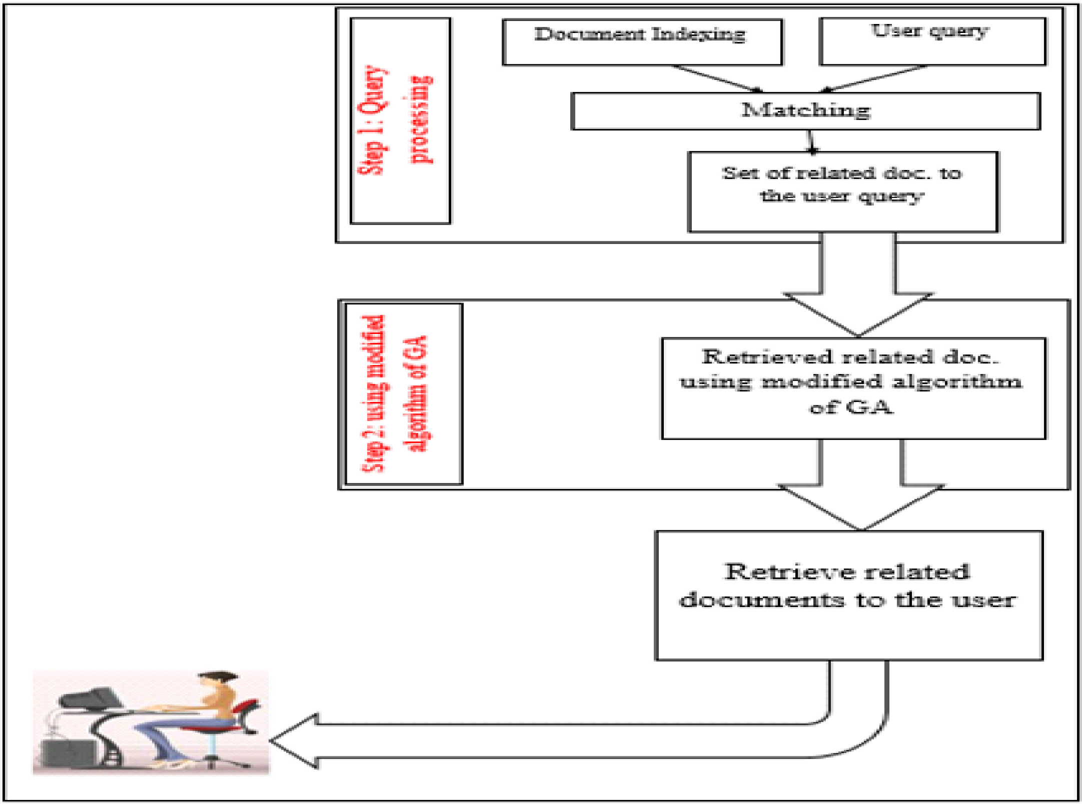
Word	PageList
a	6-4-54
abba	6-4-52
abdel	6-4-58
abe	3-2-350
abel	3-32 | 3-1-580 | 3-3-145 | 3-3-150 | 3-3-204 | 3-3-224 | 3-3-226 | 3-3-227 | 3-3-308 | 3-3-478 | 3-3-509 | 3-3-864
abelardo	3-3-679
abeli	3-0-55
abelian	3-0-333 | 3-0-334
abelink	3-1-52
abelson	0-1-142 | 1-1-135 | 2-1-214 | 2-1-215 | 3-1-48 | 3-1-186 | 3-2-84 | 4-1-392
aben	3-0-73
aber	6-1-215
abercrombie	2-1-173 | 2-1-174 | 6-1-197
aberdeen	3-2-225 | 3-2-266 | 3-2-272 | 3-2-279 | 3-2-280 | 3-2-281 | 3-4-754 | 4-1-416
aberdeenshire	3-2-225
aberman	0-3-23 | 3-3-184 | 3-3-185 | 3-3-760 | 3-3-806 | 6-3-0
abernat	0-1-385
abernathy	2-1-751
abernethy	3-1-164

3.2 Searching algorithm stage

This is stage consists of two steps:-

- First: - user query processed to produce a set of words or keywords.
- Second: - the related documents to the user query will be retrieved using a modified GA.

Figure 3 shows the modified searching algorithm.

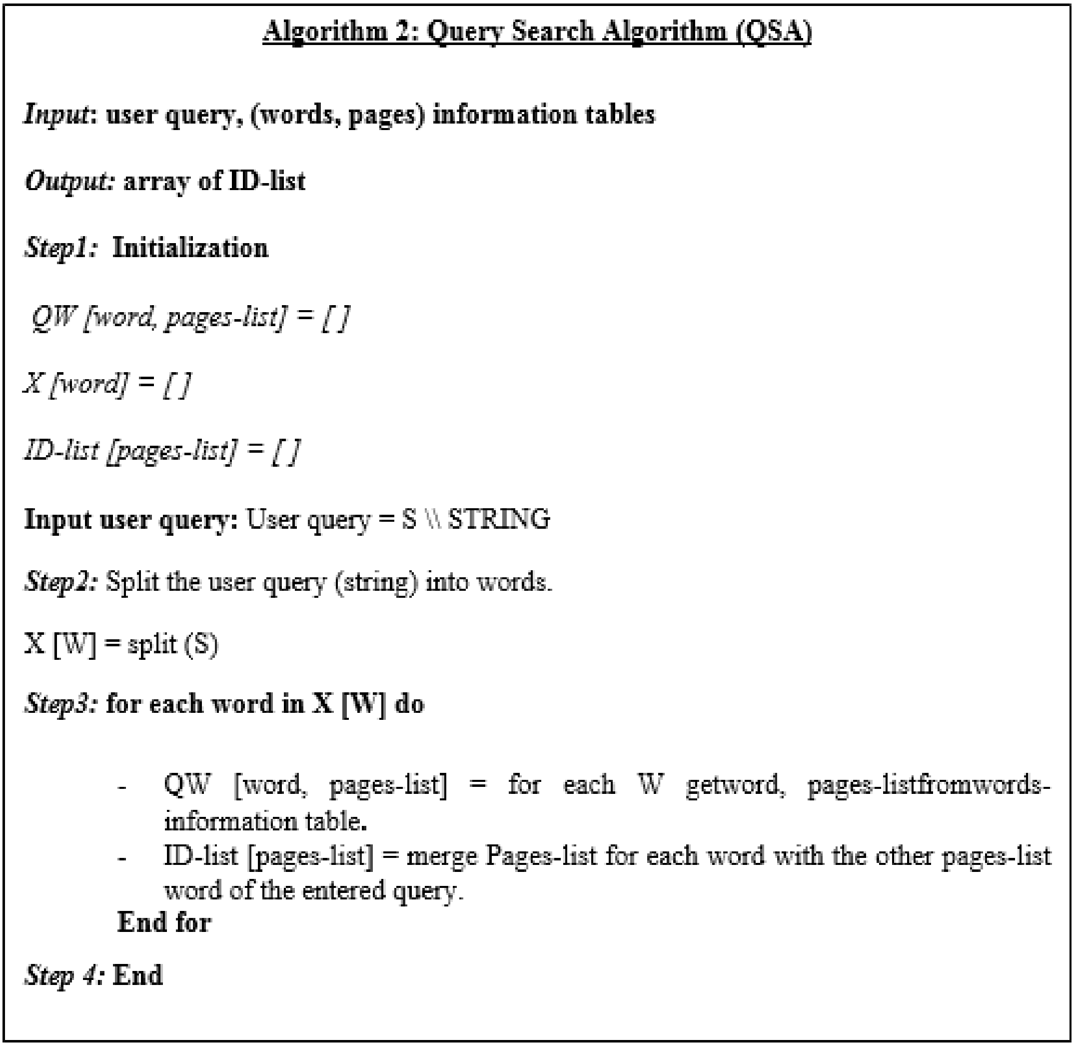


Figue3: Modified search algorithm

The detail explanation of this stage is as below.

a- Query Processing

The processed system converts string of input user query to a set of words. These words are stored in an array. Then get (word, page-list) from words-information table and save it (word, pages-list) in array called Query Words (QW). After acquiring the information about each word from database then the pages-list for each word merges with the other pages-list word of the entered query into one array, it is called ID-list, which is used as initial generation of GA, but this ID-list contains repeated ID. This repeat is useful in initial generation, because if the page contains most frequent ID, this increases the probability for the population. Algorithm 2 describes the main steps of the Query Search.



Algorithm 2: Query Search Algorithm (QSA)

```
Input: user query, (words, pages) information tables
Output: array of ID-list
Step1: Initialization
 QW [word, pages-list] = [ ]
X [word] = [ ]
ID-list [pages-list] = [ ]
Input user query: User query = S \\ STRING
Step2: Split the user query (string) into words.
X [W] = split (S)
Step3: for each word in X [W] do
        -  QW [word, pages-list] = for each W getword, pages-listfromwords-
           information table.
        -  ID-list [pages-list] = merge Pages-list for each word with the other pages-list
           word of the entered query.
       End for
Step 4: End
```

b- Modified GA

The proposed system based on using a modified GA, which defined to achieve paper aim. The modified GA differs from the classic GA through using new fitness function, basic parameters and operators are defined as the following:

i. Initial Generation

In this paper, each chromosome is represented as a set of pages. Each page is represented by a page-ID, which is a list of three integer numbers, first number represents the university code, second represents directory code, and last one represents page code. Figure 4 is example of chromosome representation) [6 and 7].

Gene1	gene2	gene3	gene4	gene5	gene6
0-1-362,	***0-3-53,***	***0-2-18,***	***0-1-383,***	***0-1-399,***	***0-4-45***

Figure 4: chromosome representation

Chromosome length represents the number of shared page between query and QW, this represents ID-list, output of algorithm 2.

ii. B. Fitness Function (FF)

Proposed fitness function is used; equation 2 describes the used FF in this paper.

$$\mathrm{FF} = \sum_{\mathrm{C}=0}^{15-1} \sum_{\mathrm{g}=0}^{\mathrm{sp}-1} (\mathrm{SPT(g)} + 0.1) + \mathrm{WOT(g)} + (\mathrm{PID(g)} * 0.5) + \propto \quad \ldots 2$$

where:**C** is the chromosome,

g is thegene (page),

sp is the number of shared pages (pages which contain words of user query),

SPT (g)+0.1represents number of shared pages of words user query together,

PID is the page_ID (each page contains word of user query together),and can be multiplied, each page contains all query

together with a value of (0.5) which is determined by experiment, to ensure the fitness of this page it stays high
$\propto$ is simple value used to keep priority of more relevant documents to the user querystays at the first position of each chromosome. The value is determined by subtracting FF value of each page in the chromosome with the index shown in equation 3.

$$\propto = FF\ (page) - index\ (page)\ \ldots.\ (3)$$

From initial generation all pages can be checked, to see which page contains the words of user query together. Each appearance of query words together can increase the fitness value of the page with 0.1. This value is determined by experiment. In the following example, chromosome consists of six shared pages. Only three of these shared pages contain the words of user query together, and FF is 1100 after computing total-weight and words-count from pages-information table.

0-1-362, 0-3-53, 0-2-18, 0-1-383, 0-1-399, 0-4-45 1100

Check if that query found in these pages came together, if this condition verified, appearance of words query can be increased by 0.1 if these words of query come together. Suppose the query words appear four times in these three pages, so the value of FF become compute as follows:

For first appearance, it becomes 1100+0.1= 1100.1
For second appearance, it becomes 1100.1+0.1= 1100.2
For third appearance, it becomes 1100.2+0.1= 1100.3
For third appearance, it becomes 1100.3+0.1= 1100.4
This is the final FF value of the chromosome and the chromosome become:

0-1-362, 0-3-53, 0-2-18, 0-1-383, 0-1-399, 0-4-45 1100.4

WOT is the weight of query word tags (1 to 6) (html, head, sub-header (h1, h2, h3), and body). For each word of user query, assign weight to determine tag of this query. For example; if the query is: ***Operating system*** and this query word comes together, to split pages, which are near to the user query, if the query comes in the tag title, this query increases with six because the weight of this tag is six. Moreover, with other page in body, the weighting becomes one, because the weighting of body is one and so on.

Suppose the FF value of page contains the query, ***Operating system*** is 1000 and this page contains this query in title, then this value increases with six as follows:

FF= 1000+6= 1006
Suppose this query appears in the other page but in the body, then FF value will be:

FF=1000+1= 1001

The WOT gives importance for the query which appeared in the title than the query appeared in the body.

ii.Parent Selection

Elitism technique has been used in the process of parent selection. It means the best individual is copied in the next generation; this technique prevents weak individuals participating in the process of making solutions, and allows only strong individuals to participate in the process of generate of solutions, the best individuals have maximum fitness[8].

iii. Single-Point Crossover Operator

A proposed single-point crossover operator is used, where first a Cross Point (CP) is determined by take the midpoint of chromosome length's splits the chromosome into two parts (left and right), then each gene in the first chromosome is swapped with the genes as that in the second chromosome if this gene is the same in the second, this gene stays in the first chromosome and not exchange [9-11].But, if the gene is not found in the second chromosome this gene can be exchanged with the second gene etc. so that, the fitness value after this process can be increased or decreased according of the gene exchange. Algorithm 3 illustrates the proposed single-point crossover operation.

Algorithm 3: pseudo-code of proposed single-point crossover operation

```
Input: Two random chromosome (R1, R2), CP

Output: Chromosome with high fitness

Step 1:

CP=chrom.length/2

Step 2:

For i=0 to selected.length-1

R1=selected (R1)     \\ select random chromosome 1

R2= selected (R2)    \\ select random chromosome 2

Step 3: Swapping

            Cost1=population.chrom (R1).fitness

            Cost2=population.chrom (R2).fitness

                    Swapping (R1, R2)
           Newcost1=get-fitness (R1)
           Newcost2=get-fitness (R2)
Step 4: Fitness function check
                      If (Newcost1 > cost1 OR Newcost2 > cost2) then
Population.chrom (R1).fitness= newcost1
Population.chrom (R2).fitness= newcost2
Else
Swapping (R1,R2)
End if
End for
Step 5: End
```

iv. **Mutation Operation**

Last operation of GA is mutation, and can be used to increase the probability of retrieved related documents. A proposed mutation operation is used. A simple method of this operation is random exchange gene on the right chromosome with the random gene on the left of the same chromosome. Figure **5** shows the mutation operation.

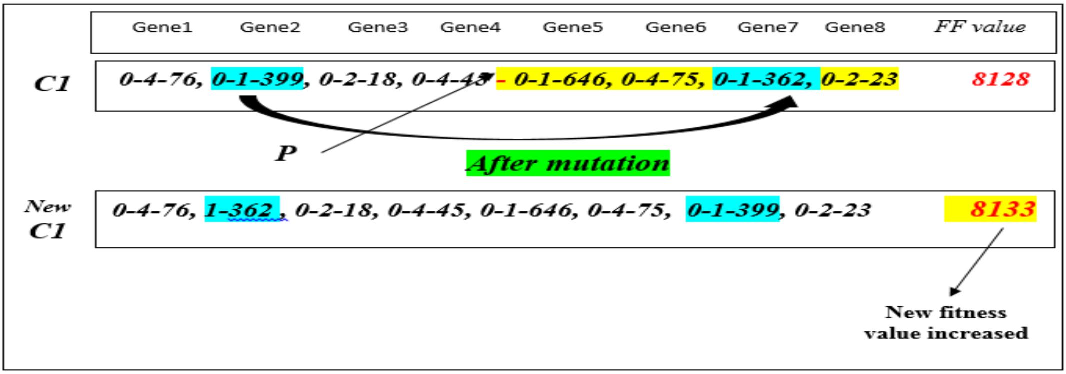


Figure 5: Mutation operation

Fitness value of generated chromosome (new chromosome), may be increased or decreased depending of the gene exchange. The relevant documents to the user query are return and display to the user with high FF in the last iteration (15 iteration), for example using the following query:

Query: Airborne Information Technology Services

For this query, there are five documents, which are the shared pages, but only one relevant of these documents contains all words of query together.

4. Experimental results

Different length of 50 queries sets have been used to test the proposed system. The accuracy has been measured by using two measures, precision in equation 4 and recall equation 5 [12, 13]. In addition the retrieval speed has been computed by computing time of retrieval. Table 3 showthe experimental results. Also, figure 6 show the accuracy result while figures 7 show the performance.

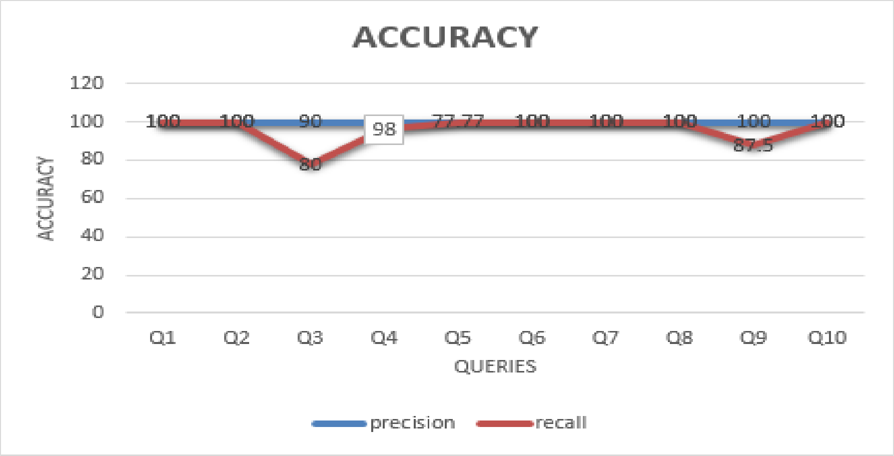


Figure 6: curacy results

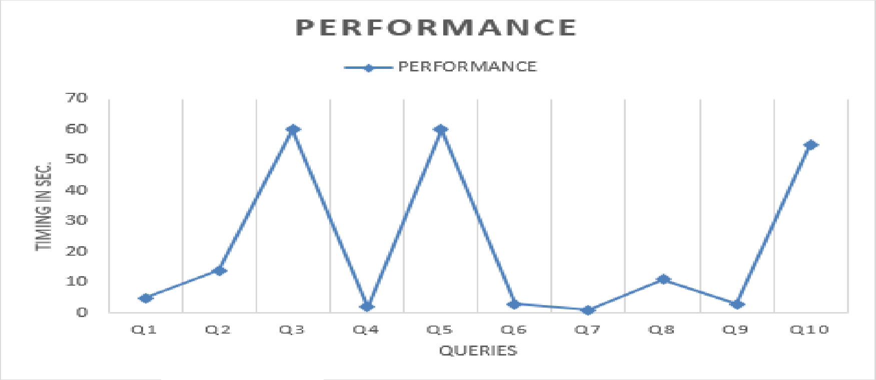


Figure 7: **rformance of 10 queries in sec.**

It can be noticed that the accuracy of precision measure is 100%, this means that one problems of information retrieval can be solved. In addition, the second problem also can be prevented but not finally.

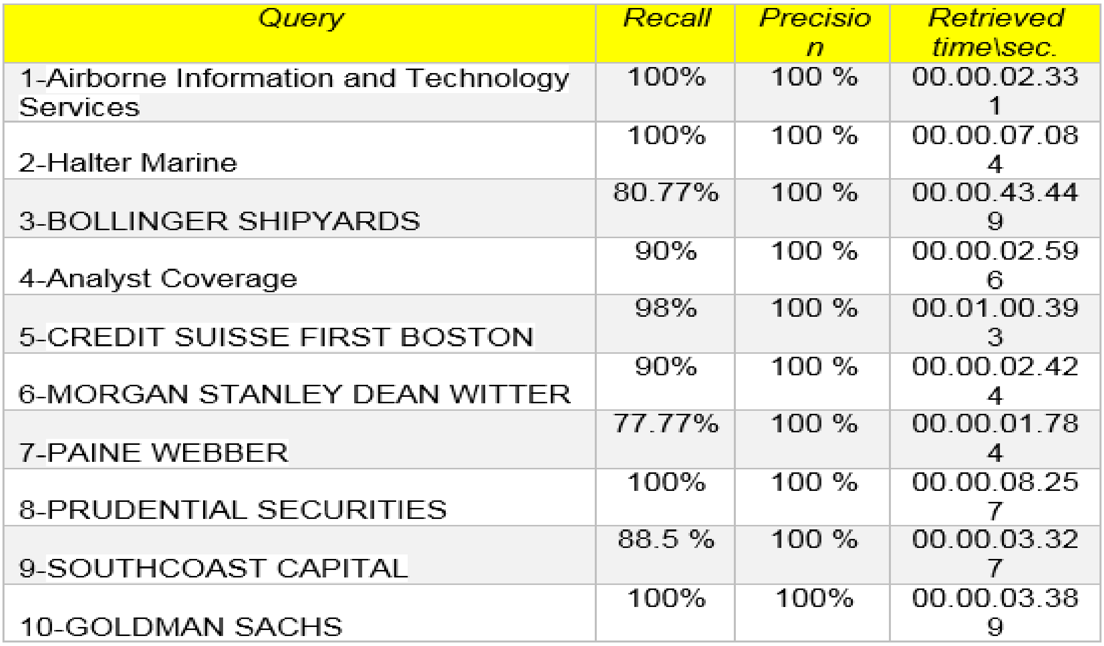
P=PL\all PL ….4

P: precision,PL: related documents,All PL: all related document

R=RL\RT …5

R:recall,RL:relateddocuments

Table 3: set of testing queries with results



Query	*Recall*	*Precision*	*Retrieved time\sec.*
1-Airborne Information and Technology Services	100%	100 %	00.00.02.331
2-Halter Marine	100%	100 %	00.00.07.084
3-BOLLINGER SHIPYARDS	80.77%	100 %	00.00.43.449
4-Analyst Coverage	90%	100 %	00.00.02.596
5-CREDIT SUISSE FIRST BOSTON	98%	100 %	00.01.00.393
6-MORGAN STANLEY DEAN WITTER	90%	100 %	00.00.02.424
7-PAINE WEBBER	77.77%	100 %	00.00.01.784
8-PRUDENTIAL SECURITIES	100%	100 %	00.00.08.257
9-SOUTHCOAST CAPITAL	88.5 %	100 %	00.00.03.327
10-GOLDMAN SACHS	100%	100%	00.00.03.389

5- Conclusion

GA has been used in IR. The tow problems of IR have been solved. Also, the time of retrieve the relevant documents to the user query is reduced by using the proposed FF. In addition, the storage space of storing documents is also reduced by using the proposed system.Traditional method needs a large space of memory, for each entry read document needs 2-byte (2*8), and dataset used is 8280 documents. It is required to multiple total words in dataset (67,672 words), with all documents in dataset (8280 documents) with 16, the result is 8965,186,560 MB. On the other hand, the proposed documents indexing in the first stage of this proposed system, only needs 19.9MB to store data in memory see next Figure8.

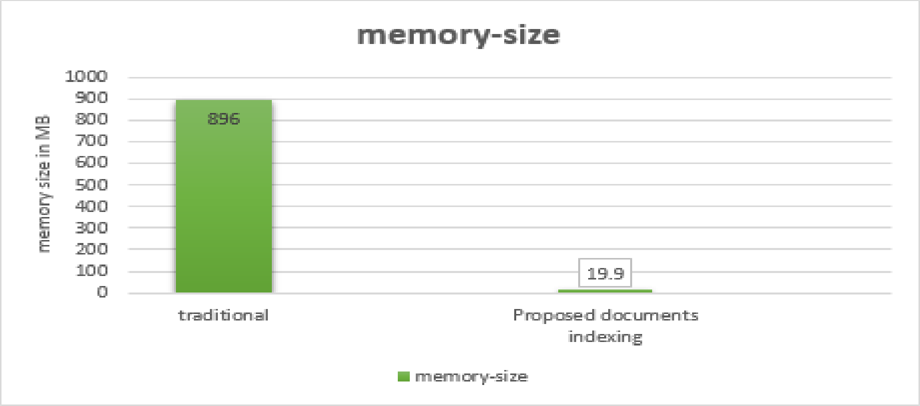


Figure 8: Memory space for document indexing

References

[1] Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze, "An Introduction to Information Retrieval", Book, Cambridge University Press, February 16, 2008.

[2] Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote, Felix de Moya-Anegon, “Genetic algorithms in relevance feedback: a second test and new contributions”, Proceedings in Information Processing and Management 39, 2003.

[3] A. P. Siva Kumar, Dr. P. Premchand, Dr. A. Govardhan, “Query-Based Summarizer Based on Similarity of Sentences and Word Frequency”, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.1, No.3, May 2011.

[4] Vrajitoru, D. (2000). Large population or many generations for genetic algorithms?Implications in information retrieval. In F. Crestani, and G. Pasi (Ed.), Soft Computing in Information Retrieval. Techniques and Applications (pp. 199-222). Physica-Verlag, Heidelberg.

[5] Al-Hajri, M.T.; Abido, M.A., (2009), Assessment of Genetic Algorithm selection, crossover and mutation techniques in reactive power optimization, IEEE Congress on Evolutionary Computation, 2009. CEC '09. Publication Year: 2009 , pp. 1005 - 1011

[6] Aly, A. (2007). Applying genetic algorithm in query improvement problem. Information Technologies and Knowledge , vol.1, pp. 309-316.

[7] Asllani, A. and Lari, A. (2007). Using genetic algorithm for dynamic and multiple criteria web-site optimizations. European Journal of Operational Research, vol. 176, no.3, pp. 1767-1777.

[8] Beasley, D., Bull, D. R. and Martin, R. R. (1993). An overview of genetic algorithms: part 2, Research Topics. *University Computing*, vol. 15, no. 4, pp. 170-181.

[9] Billhardt, H., Borrajo, D. and Maojo, V. (2002). Using genetic algorithms to find suboptimal retrieval expert combinations.In Proceedings of SAC, pp. 657-662.

[10] Húsek, D., Snášel, V., Owais, J., and Krömer, P. (2005). Using genetic algorithms for boolean queries optimization. Proceeding of the Ninth IASTED International Conference internet and multimedia ststems and applications, pp. 178-184. Honolulu, Hawaii, USA.

[11] Klabbankoh, B., and Pinngern, O. (2008). Applied GeneticAlgorithms in Information Retrieval.Retrieved Aug 22, 2009, fromhttp://www.ils.unc.edu/~losee/gene1.pdf.

[12] Lopez-Pujalte, C., Guerrero-Bote, V. P., and de Moya-Anegon, F. (2003). Genetic algorithms in relevance feedback: a second test and new contributions. Information Processing and Management, vol. 39, pp. 669–687.

[13] Marghny, M. H., and Ali, A. F. (2005). Web mining based on genetic algorithm. AIML 05 Conference.Cicc, Cairo, Egypt.

[14] Data Set. [online] 7 sector. Available http://www.cs.cmu.edu/~webkb/].

[15] L. Araujo and J. Perez-Iglesias, "Training a Classifier for the Selection of Good Query Expansion Terms with a Genetic Algorithm," in Proc. IEEE Conference on Evolutionary Computation, Barcelona, pp. 1-8, 2010.

[16] M. Latih AND S. Essam, "Applying Genetic Algorithms to Information Retrieval Using Vector Space Model", 2011: search Methods, [Online]. Available at:http://airccse.org/journal/ijcsea/papers/5115ijcsea02.pdf.

[17] A. Al-Dallal and R. S. Abdul-Wahab, apply, "Enhancing Recall and Precision of Web Search by Using Genetic Algorithm", 2012. Search Methods, [Online]. Available at:http://bura.brunel.ac.uk/bitstream/2438/7379/1/FulltextThesis.pdf.

[18] M. Wafa, A. Mamoun and K. Ghasan, "Improving the Effectiveness of Information Retrieval System Using Adaptive Genetic Algorithm", 2013: search Methods, [Online].
Available at: http://airccse.org/journal/jcsit/5513ijcsit06.pdf.

[19] A. Ahmed A. Bahgat, and M. Abdel, "Using Genetic Algorithm to Improve Information Retrieval Systems", 2014: search Methods,
[Online]Available at:
http://www.researchgate.net/publication/242286994_Using_Genetic_Algorithm_to_Improve_Information_Retrieval_Systems.

إسترجاع المعلومات باستخدام الخوارزمية الجينية المُحَدَّثة

أ.م.د.علياء كريم عبد الحسن*   م.م .دعاء نتيشة مهاوي*   م.م سارة نجم عبد الواحد**

المستخلص

العديد من التقنيات تم اقتراحها لغرض استرجاع وثائق HTML المطلوبة من قبل المستخدم. الخوارزمية الجينية تعتبر واحدة من هذه التقنيات والتي تعمل على خلق عدة اجيال بواسطة استخدام عمليات الاختيار، التزاوج والطفرة الوراثية لانتاج النتائج. في هذا البحث تم التركيز على تحسين عمليات الخوارزمية الجينية واقتراح دالة مفاضلة جديدة لغرض تحسين نوعية الوثائق المسترجعة. هذه التقنية المقترحة تمتطبيقها على وثائق HTML وتم استخدام مقاييس مثل اعادة الاستدعاء والدقة لغرض تقييم هذه التقنية المقترحة . هذا وقد اظهرت النتائج درجة عالية من التحسن في نوعية الوثائق المسترجعة من خلال هذه المقايس التي تم استخدامها .

---

*الجامعة التكنولوجية

**وزارة التعليم العالي والبحث العلمي