ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i1.16734

Cluster-based information retrieval by using (K-means)-hierarchical parallel genetic algorithms approach

Sarah Hussein Toman[1], Mohammed Hamzah Abed[2], Zinah Hussein Toman[3],
[1]Roads and Transport Department, College of Engineering, University of Al-Qadisiyah, Ad-Diwaniah, Iraq
[2,3]Computer Science Department, College of Computer Science and Information Technology,
University of Al-Qadisiyah, Ad-Diwaniah, Iraq

Article Info

Article history:

Received May 18, 2020
Revised Aug 14, 2020
Accepted Aug 29, 2020

Keywords:

Cluster-based IR
HPGA
Information retrieval
K-means
Web document clustering

ABSTRACT

Cluster-based information retrieval is one of the information retrieval (IR) tools that organize, extract features and categorize the web documents according to their similarity. Unlike traditional approaches, cluster-based IR is fast in processing large datasets of document. To improve the quality of retrieved documents, increase the efficiency of IR and reduce irrelevant documents from user search. In this paper, we proposed a (K-means)-hierarchical parallel genetic algorithms approach (HPGA) that combines the K-means clustering algorithm with hybrid PG of multi-deme and master/slave PG algorithms. K-means uses to cluster the population to k subpopulations then take most clusters relevant to the query to manipulate in a parallel way by the two levels of genetic parallelism, thus, irrelevant documents will not be included in subpopulations, as a way to improve the quality of results. Three common datasets (NLP, CISI, and CACM) are used to compute the recall, precision, and F-measure averages. Finally, we compared the precision values of three datasets with Genetic-IR and classic-IR. The proposed approach precision improvements with IR-GA were 45% in the CACM, 27% in the CISI, and 25% in the NLP. While, by comparing with Classic-IR, (K-means)-HPGA got 47% in CACM, 28% in CISI, and 34% in NLP.

Corresponding Author:

Sarah Hussein Toman
Roads and Transport Department
College of Engineering, University of Al-Qadisiyah
Ad-Diwaniah, Iraq
Email: sarah.toman@qu.edu.iq

1. INTRODUCTION

In the recent years, the information has been overloaded because of the rapid growth of the web. To deal with this information a web document information retrieval task is used to retrieve the most relevant documents to a user query [1, 2]. Information retrieval needs to scan all documents that are found in a database, then give scores according to a relevance degree to the user query, then rank all results and present them to the user [3, 4]. Thus, information retrieval requires long runtime to scan all documents. The cluster analysis tool plays a basic role in information retrieval to improve the information retrieval performance by reducing the search time and to prevent irrelevant results from the retrieved documents. The idea behind the web document clustering is to assign a dataset of web documents to a set of clusters that depend on the similarity's degree among them. Therefore, it becomes easy for search engines to query in the same cluster if each web page is assigned to a similar group [5, 6].

An efficient clustering algorithm and genetic algorithm should represent a document as structured data using the document representation model. The most common aspect used in document representation is the vector space model (VSM) [7]. Besides, a similarity degree between two documents or clusters should be measured by using one of the similarity measures [1]. Hierarchical and partition algorithms are the major kinds of clustering algorithms have been used [8]. A hierarchical clustering algorithm generates a tree of clusters (groups) depending on two methods. The first method starts with one cluster then merges each two similar clusters, which is known as the agglomerative method. The second one starts from the whole data set as one cluster then split it into clusters at each stage, is known as the divisive method [9, 10]. A partition clustering algorithm uses a single step to divide the collection of documents in to predefined number of groups [11]. The most widely used partition clustering algorithm is the K-means algorithm [12]. It is an unsupervised learning algorithm that relies on selecting K clusters as K-centroids. After that, the similarity measure is calculated between each document and the centroids, then the documents will assign to the closest centroid after updating of centroids multiple times [13].

In the present paper, the k-means cluster with two levels of genetic parallel is used for information retrieval. Multi-deme parallel genetic as first level and master-slave parallel genetic as second level. The idea behind using the K-mean clustering algorithm is to group a set of documents to clusters according to their similarity with a query, then an HPGA algorithm will perform a search in the most relevant clusters to reduce the search time and to provide optimal search results. Next, at each subpopulation there is a fitness evaluation parallelism with hybrid selection and two chromosomes crossover as genetic operators. Then migration among individuals and repeat HPGA steps n time until obtaining the optimal results.

2. TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY (TF-IDF)

Datasets in most clustering algorithms are represented by a set of vectors, V = { V1, V2, V3… Vn}, where, Vi is the feature vector of one object. Term Frequency is a simple and effective term selection method, alike words are used in the documents that belong to the same subject, thus, term frequency can be a respectable indicator for a certain subject. TF is a term occurrence frequency in the document as shown in (1). On another hand, some terms should be removed such as words in the stop list corresponding to the English language, because the occurrence of these words is not relevant to identify the subject of the document [14].

$$TF(j, i) = \text{frequency of } i \text{ th term in document } j \quad (1)$$

TF is not effective to measure the frequent terms in a set of documents. Thus, inverse document frequency (IDF) is used. TDF is the term frequency across a set of documents as shown in (2).

$$IDF(ti) = log \frac{|D|}{|D_{ti}|} \quad (2)$$

|D|, number of documents.

|Dti|, number of documents that contain the term ti.

To determine the weight for each term ti in each document dj, TF and IDF will be combined by multiplication of the resulted values, TF-IDF given as shown in (3) [15]. In document clustering, terms with higher TD-IDF have better clustering.

$$TF\text{-}IDF\ (ti, dj) = TF\ (j, i) * IDF(ti) \quad (3)$$

3. GENETIC ALGORITHM

The genetic algorithm (GA) is a probabilistic meta-heuristic search algorithm inspired by natural genetics [16, 17]. GA gives a good solution in many life fields. Figure 1 demonstrates the flowchart of the genetic algorithm steps. The basic operations of a genetic algorithm are [18, 19]:

- Generate random solutions that are called a population.
- Determine Fitness value to evaluate each solution.
- Select the best solutions according to the fitness.
- Produce a new population by genetic operators (crossover and mutation).

As employ the parallelism feature to reduce the process duration. There are three models of parallel genetic algorithms (PGA) as exhibited in Figure 2: a) master/slave PGA which deals with single population and parallel fitness calculation; b) multi deme PGA which deals with multi-population and parallel genetic operations followed by migration among them; c) cellular which deals with a single population running on a parallel processing system based closely-linked massively. The previous models can be hybridized to produce hierarchical PGA (HPGA) models [20, 21].

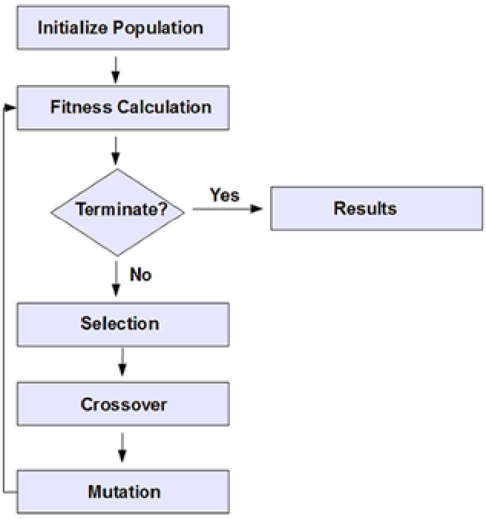
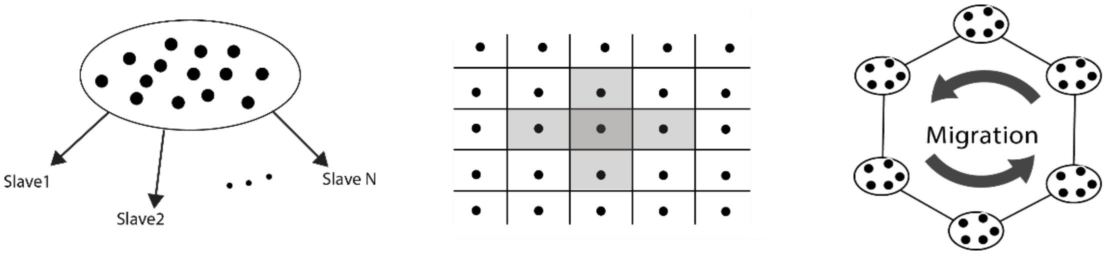


Figure 1. Genetic algorithm steps



(a) (b) (c)

Figure 2. (a) Master/slave PGA, (b) Multi deme PGA, (c) Celullar PGA

4. THE PROPOSED APPROACH

The Information Retrieval systems process a large amount of text in documents index and user query stages. Parallelism is a way to improve the query average time. The elaborated procedure uses a parallel genetic algorithm (PGA) with K-means to retrieve the most relevant documents to a user query that relies on the steps enumerated below, Figure 3 presents the proposed (K-mean)-HPGA approach:

4.1. Web document data extraction

Web page extraction represents the interaction with web page source (HTML) to scrap the information, respectively to identify structured data as a post-processing stage that is composed of two steps:

a. Tree-based extraction

Web pages have a semi-structured feature, therefore, this feature is considered the most important feature to represent the HTML tags and text as a labeled tree, which is called a document object model (DOM) [22], and addressing the element's tag in the tree via XPath language.

b. Text tokenizer

Its purpose is to break the text in tokens, eliminating stop words and stemmer from tokens. The Stop Wordlist that we used, contains 1300 words which include articles (a, an, the), prepositions (in, into, on, at), conjunctions (and, or, but, and so on), pronouns (she, he, I, me), and other words irrelevant for the query process. Porter Stemming is used in our approach to enhance accuracy via dropping morphological variants of words. Thus, tokens with common stems such as -ED,-ING,-ION, and -IONS will have similar meanings.

4.2. Document and query representation

In this approach, vector space model (VSM) is used, a features vector is generated from each document content and the given query, depending on the occurrence of words in the document by using TF-IDF function (the frequency occurrence of the term in the document (TF) with the frequency of occurrence of the term in the data set of documents (TF-IDF), as shown in (3).

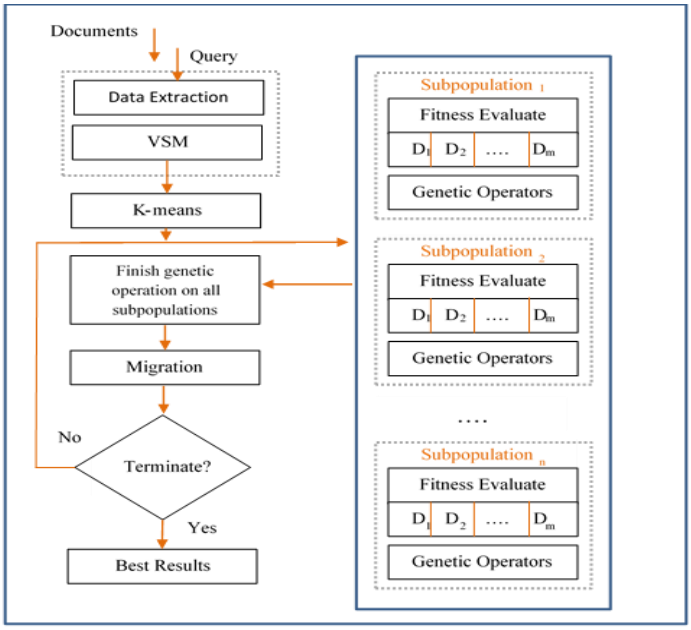


Figure 3. (K-means)-HPGA approach

4.3. K-means-hierarchical parallel genetic algorithm approach

The idea behind using the Parallel algorithm is to split the task into a set of subtasks that will exhibit a divide-and-conquer behavior. In our approach we use multi-deme parallel genetic (multiple population) with k-means clustering. Steps bellow explain the algorithm operation in details:

4.3.1. Generate population

Create the subpopulations from the web document dataset via the K-means algorithm. K-means split the documents to be indexed into k clusters then evaluate the last centroid with a query and select just clusters that are near from the query. The K-means steps are descriped by the following algorithm:

K-means algorithm

```
Input: D = {$d_1$, $d_2$, $d_3$,…,$d_n$}, set of documents.
K: number of clusters.
Output: C = {$C_1$, $C_2$, $C_3$,…,$C_k$}, set of clusters.
Step1: Let centroid $c_j$ = random number // j= 1,…,k
Step2: Foreach ($d_i$ in D)
Calculate CosDistance ($d_i$, $c_j$), i = 1,…, n, j = 1,…,k
end
Step3: Assign each document $d_i$ with minCosDistance ($d_i$, $c_j$) to cluster $C_j$
Step4: Update centroid $c_j$, for all j
Step5: Repeat (step2 and step 3) Until (no change in cluster Cj)
Step6: End.
```

4.3.2. Fitness evaluation

The second level of the parallel algorithm is applied to evaluate the fitness function in each cluster (subpopulation), i.e all documents in the cluster will be evaluated at the same time under the slave/master parallel concept. This evaluation starts by forwarding user query to each cluster then calculate the fitness function to each document of the cluster. In the present approach, a cosine similarity function is used as a fitness function [23]. The cosine similarity function is given in (4).

$$\text{Scos}=\frac{\sum_{i=1}^{n} \text{Pi Qi}}{\sqrt{\sum_{i=1}^{n} \text{Pi}^2}\sqrt{\sum_{i=1}^{n} \text{Qi}^2}} \tag{4}$$

4.3.3. Genetic operators

Generate a new population by applying genetic operators (selection and crossover). To improve genetic performance, we move 4% of chromosomes with the highest probability in the next generation without change (i.e. apply elitism feature). Genetic operators in (K-means)-HPGA flow the following steps:

a. Calculate the probability for each chromosome, where P[i] = Fitness[i]/Total
b. Rank the Probability values and take the top 4% Elitism to avoid the loss of fittest chromosomes in the new population.
c. Hybrid roulette-tournament selection (HRTS): It is the process of selecting a pair of parents from the population to emphasize fitter offsprings in a new population. In our approach we used a hybrid method to take advantage of both selection methods (roulette wheel and tournament). The selection process is explained by the following algorithm:

HRTS algorithm

```
Output: parent1, parent2

Begin
for j = 1 : 2
r = randi[1, pop_size] //Select random number for subpopulation
for i = 1 : r
sum_fitness = sum (fitness)
P_sum = randi[1, sum_fitness];
S = 0; index = 1;
S = S + fitness[i];   index++;
if (s < P_sum) goto 10, else subPop[i] = current chromosome
end
Parent[j] = maxFitness(subPop) // select parent
end
End
```

d. Crossover operation aims to get better offspring by generating a new child from two selected parents. In this approach, we proposed to represent the population as a matrix, each chromosome vector representing a row in the matrix, then select two random positions in the range [1, vector_length]. The crossover is described by the following algorithm:

Two chromosomes crossover algorithm

```
Input: subP = subP-Elite Count.
Output: offsprings
Begin
subP_length = length(subP);
repeat
Call selection function to select two parents;
10 Call pickTwoPosition (subP_length);
Exchange two positions betweentwo selected parents;
until (index <= subPsize) Goto 10;
End
function [ position1, position2 ] = pickTwoPosition (subP_length)
r = randi([1, subP_length],2)// generate 2 random integer numbers to vector r
position1 = r(1);
position2 = r(2);
end
```

4.3.4. Migration

Migration is the synchronous process that means the exchanging of memebers. It waits for the evaluation of all chromosomes in all subpopulations to exchange the individuals. Migration has an interval that is set to 5 in our approach.

4.4.5. Terminate

The last step in our approach is repeating the previous steps (from fitness to migration). These steps will reapeat n times, where n is the size of the population. After complete the repeation, the documents will rank according to fitness probability values. Then the best results will have retrieved from the documents that have hiegher rank.

5. EXPERIMENTAL RESULTS

Three datasets were used for experimental results. NPL dataset (DS1) containing 11,429 electronic engineering documents, CISI dataset (DS2) with 1,460 computer science documents and CACM dataset (DS3) consisting of 3204 communications documents. To evaluate the web documents retrieval, the recall, precision, and F-measure are used for 100 queries in three datasets as defined in the following equations [24, 25]:

$$Recall\ (R) = \frac{\text{relevant items retrieved}}{\text{relevant items}} \tag{5}$$

$$\textit{Precision (P)} = \frac{\text{relevant items retrieved}}{\text{retrieved items}} \quad (6)$$

$$\text{F-measure} = \frac{2\ .\ R\ .\ P}{R+P} \quad (7)$$

The results are shown in Tables 1-3. For the NPL dataset (DS1) where precision average is 0.688889 and F-measure average is 2.0667, while in the CISI dataset (DS2), the precision average was 0.65889 and the F-measure average was 1.97667. Finally, the CACM dataset (DS3) the average for precision and F-measure were 0.748889 and 45.22222 respectively. After the analysis of the previous results, the third dataset gave higher results in both measures.

Table 1. The results of recall, precision and F-measure for 100 query in NPL dataset (DS1) by using (K-means)-HPGA approach

Recall	Precision	F-measure%
0.1	0.9	2.7
0.2	0.87	2.61
0.3	0.84	2.52
0.4	0.77	2.31
0.5	0.74	2.22
0.6	0.66	1.98
0.7	0.58	1.74
0.8	0.46	1.38
0.9	0.38	1.14
AVG	0.6888	2.0666

Table 2. The results of recall, precision and F-measure for 100 query in CISI dataset (DS2) by using (K-means)- HPGA approach

Recall	Precision	F-measure %
0.1	0.89	2.67
0.2	0.84	2.52
0.3	0.78	2.34
0.4	0.76	2.28
0.5	0.69	2.07
0.6	0.55	1.65
0.7	0.51	1.53
0.8	0.47	1.41
0.9	0.44	1.32
AVG	0.6588	19.766

Table 3. Displays the results of recall, precision and F-measure for 100 query in CACM dataset (DS3) by using (K-means)-HPGA approach

Recall	Precision	F-measure %
0.1	0.94	2.82
0.2	0.9	2.7
0.3	0.87	2.61
0.4	0.85	2.55
0.5	0.8	2.4
0.6	0.77	2.31
0.7	0.66	1.98
0.8	0.54	1.62
0.9	0.41	1.23
AVG	0.7488	22.466

We measured the improvements that were achieved by the proposed approach, with a precision of information retrieval by genetic algorithm (GA-IR) for three datasets. Tables 4-6 presents a comparison between our approach and GA-IR. Improvement average is calculated for three datasets and the results were 25.6666, 27.4444, and 45.2222 respectively. Finally, we compared the proposed approach with classic Information Retrieval (classic-IR) precision and the improvements were 34.4444% in NLP, 28.6666% in CISI, and 47% in CACM as shown in Tables 7-9.

Table 4. Comparison analysis of (K-means)-HPGA approach and GA [26] in NPL dataset (DS1)

Recall	HPGA-(K-means) (p)	GA-IR(P)	Improvements %
0.1	0.9	0.88	2
0.2	0.87	0.66	21
0.3	0.84	0.59	25
0.4	0.77	0.44	33
0.5	0.74	0.4	34
0.6	0.66	0.31	35
0.7	0.58	0.27	31
0.8	0.46	0.19	27
0.9	0.38	0.15	23
AVG	0.6888	0.4322	256.666

Table 5. Comparison analysis of (K-means)-HPGA approach and GA [26] in CISI dataset (DS2)

Recall	HPGA-(K-means) (p)	GA-IR(P)	Improvements %
0.1	0.89	0.8	9
0.2	0.84	0.55	29
0.3	0.78	0.48	30
0.4	0.76	0.39	37
0.5	0.69	0.36	33
0.6	0.55	0.28	27
0.7	0.51	0.24	27
0.8	0.47	0.2	27
0.9	0.44	0.16	28
AVG	0.6588	0.3844	274.444

Table 6. Comparison analysis of (K-means)-HPGA approach and GA [26] in CACM dataset (DS3)

Recall	HPGA-(K-means) (p)	GA-IR (P)	Improvements %
0.1	0.94	0.79	15
0.2	0.9	0.47	43
0.3	0.87	0.42	45
0.4	0.85	0.27	58
0.5	0.8	0.23	57
0.6	0.77	0.16	61
0.7	0.66	0.14	52
0.8	0.54	0.1	44
0.9	0.41	0.09	32
AVG	0.7488	0.2966	452.222

Table 7. Comparison analysis of (K-means)-HPGA approach and classic IR [20] in NPL dataset (DS1)

Reca	HPGA-(K-means)	Classic IR	Improvements
ll	(p)	(P)	%
0.1	0.9	0.73	17
0.2	0.87	0.5	37
0.3	0.84	0.44	40
0.4	0.77	0.34	43
0.5	0.74	0.31	43
0.6	0.66	0.24	42
0.7	0.58	0.22	36
0.8	0.46	0.17	29
0.9	0.38	0.15	23
AVG	0.6888	0.3444	344.444

Table 8. Comparison between (K-means)-HPGA approach and classic IR [20] in CISI dataset (DS2)

Reca	HPGA-(K-means)	Classic IR	Improvements
ll	(p)	(P)	%
0.1	0.89	0.68	21
0.2	0.84	0.56	28
0.3	0.78	0.46	32
0.4	0.76	0.4	36
0.5	0.69	0.35	34
0.6	0.55	0.3	25
0.7	0.51	0.25	26
0.8	0.47	0.2	27
0.9	0.44	0.15	29
AVG	0.6588	0.3722	286.666

Table 9. Comparison analysis of (K-means)-HPGA approach and classic IR [20] in CACM dataset (DS3)

Recall	HPGA-(K-means) (p)	Classic IR (P)	Improvements %
0.1	0.94	0.72	22
0.2	0.9	0.45	45
0.3	0.87	0.37	50
0.4	0.85	0.25	60
0.5	0.8	0.22	58
0.6	0.77	0.16	61
0.7	0.66	0.14	52
0.8	0.54	0.11	43
0.9	0.41	0.09	32
AVG	0.7488	0.2788	47

6. CONCLUSIONS

After the tests and research for this paper, we concluded an information retrieval performance improvement: (K-means)-HPGA achieved higher precision and better quality in document retrieval. Also a reduction of irrelevant results in user search was observed. Our results were determined by comparing three common datasets (NLP, CISI, and CACM) with classic IR and GA. The range of precision improvements for three datasets with classic-IR was (28-47%) while with GA-IR the precision was (25-45%).

REFERENCES

[1] C. D. Manning, P. Ragahvan, and H. Schutze, "An introduction to information retrieval," *Cambridge University Press*, 2009.

[2] J. M. Kassim and M. Rahmany, "Introduction to Semantic Search Engine," *2009 Int. Conf. Electr. Eng. Informatics*, vol. 2, pp. 380-386, August 2009.

[3] S. M. Weiss, N. Indurkhya, T. Zhang, and F. J. Damerau, "Information retrieval and text mining," *Springer Berlin Heidelb*, pp. 75-90, 2010.

[4] Y. Wang, "Design of information retrieval system using rough fuzzy set," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 12, no. 1, pp. 844-851, January 2014.

[5] Y. Djenouri and *et al*," Fast and effective cluster-based information retrieval using frequent closed itemsets," *Information Sciences,* vol. 453, pp. 154-167, July 2018.

[6] C. Cobos, *et al.*, "Web document clustering based on Global-Best Harmony Search, K-means, frequent term sets and bayesian information criterion," *IEEE*, August 2010.

[7] S. E. Pratama, et al, “Weighted inverse document frequency and vector space model for hadith search engine,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 1004-1014, May 2020.

[8] R. F. Hassan and *et al.*, “Improving the web indexing quality through a website-search engine coactions,” *International Journal of Computer and Information Technology*, vol. 3, no. 2, March 2014.

[9] S. S. Tandel, A. Jamadar, and S. Dudugu, “A survey on text mining techniques,” *5th Int. Conf. Adv. Comput. Commun. Syst*, pp. 1022-1026, March 2019.

[10] S. H. Toman, *et al.,* “Content-based audio retrieval by using elitism GA-KNN approach”, *Journal of AL-Qadisiyah for computer science and mathematics*, vol. 9, no. 1, May 2017.

[11] A. Konar, “Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain,” *Jadavpur University, CRC Press LLC*, 2000.

[12] T. Munakata, “Fundamentals of the new artificial intelligence,” *Springer*, 2008.

[13] C. C. Aggarwal and C. Xhai, “A survey of text clustering algorithms,” *Mining text data*, August 2012.

[14] A. M. Siregar and A. Puspabhuana, “Improvement of term weight result in the information retrieval systems,” *Proceedings of 4th International Conference on New Media Studies*, November 2017.

[15] T. Tokunaga, T. Tokunaga, I. Makoto, and I. Makoto, “Text categorization based on weighted inverse document frequency,” *Spec. Interes. Groups Inf. Process Soc. Japan* (SIG-IPSJ, 1994.

[16] P. Simon and S. S. Sathya, “Genetic algorithm for information retrieval,” *2009 International Conference on Intelligent Agent & Multi-Agent Systems*, July 2009.

[17] Z. Wang and B. Feng, "Optimal genetic query algorithm for information retrieval,” *Springer*, 2009.

[18] H. Imran, “Genetic algorithm based model for effective document retrieval,” *Intelligent Control and Computer Engineering*, 2011.

[19] P. Pathak, M. Gordon and W. Fan, “Effective information retrieval using genetic algorithms based matching.” *Hawaii International Conference on System Sciences*, IEEE, February 2000.

[20] M. Ebrahimi and A. Jahangirian, “A hierarchical parallel strategy for aerodynamic shape optimization with genetic algorithm,” *Scientia Iranica*, vol. 22, no. 6, pp. 2379-2388, January 2015.

[21] K. I. Abuzanouneh, “Parallel and distributed genetic algorithm with multiple-objectives to improve and develop of evolutionary algorithm,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, 2016.

[22] J. Marini, “The document object model, processing structured documents,” *McGraw-Hill/Osborne*, 2002.

[23] S-H. Cha, “Comprehensive survey on distance/similarity measures between probability density functions”, *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, January 2007.

[24] L. T. Su, “The relevance of recall and precision in user evaluation,” Journal of the American Society for Information Science, vol. 45, no. 3, 1994.

[25] M. Junker, *et al.*, “On the evaluation of document analysis components by recall, precision, and accuracy,” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, September 1999.

[26] A. A. Radwan, *et al.*, “Using genetic algorithm to improve information retrieval systems”, *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, vol. 2, no. 5, 2008.