

# Web Information Retrieval Using Island Genetic Algorithm

Noha Mezyan, Venus W. Samawi

**Abstract**—World Wide Web (WWW) is a mine of information for most people. Due to the huge amount of information and documents available on the internet, the process of retrieving documents that are most relevant to user needs become a tremendous problem. In addition to that, the time needed for retrieving what a user searches for increases dramatically. In this paper, island genetic algorithm (IGA) is applied to achieve parallelism and speed up the web information retrieval process. To retrieve pages most relevant to user needs, four different islands are developed. Each island has different selection method, and different fitness function. These islands are executed independently on different servers to achieve the parallel behavior. Finally, the results obtained by the four islands are combined and passed to a decision making phase to choose the documents most relevant to user query. Cosine similarity measure is used to evaluate the performance of the proposed technique.

**Index Terms**— Genetic Algorithm, Island Genetic Algorithm, Web Information retrieval, Cosine Similarity Measures

## I. INTRODUCTION

WWW is the largest information source human being had ever known. It contains more than two billion pages created by millions of authors and organizations [1]. When people want to find specific information on the Web, they usually enter a simple query into a search engine. The search engine will search the WWW to determine the pages that are relevant to the user's query. It is important to emphasize that search engines rely heavily on Information Retrieval (IR). Due to the growing number of documents on the www, the process of retrieving information from the www has become a considerable problem that need to be solved. IR is concerned with finding documents that satisfies the user needs within large collections of documents[2]. Combining IR with Artificial Intelligence (AI), and techniques of data mining provide a general frame work for web mining. As a result, web information retrieval, based on web content mining, is considered as a suitable technique to extract information from the internet [3][4].

Manuscript received June 08, 2015; revised July 22, 2015.

Noha Mezyan, Department of computer science, Al-Albait University, Mafrag, Jordan (Email: mezyan\_n@yahoo.com).

Venus W. Samawi, Department of computer information system, Amman Arab University, Amman, Jordan (Email: venus@aau.edu.jo.)

Web mining discovers and extracts information from the web documents in order to improve the relevancy of the retrieved Web pages when a query is entered to a search engine [3][4]. Much research used GAs to enable the search engine to learn the user's interest and help in retrieving most relevant documents. In [5], a genetic search strategy called GeniMiner for a search engine is proposed. Two fitness functions are used to evaluate web pages. GeniMiner retrieves relevant information, but it consumes longer time compared with other standard search engine. Therefore, this approach is recommended to be used as a complementary to standard search engines. Vallim et al. (2003) [6] suggested using a personal agent that mines web information sources and retrieves documents according to user's interests. A GA that learns the importance factors of HTML tags which are used to re-rank the documents retrieved by standard weighting schemes was proposed in [7] using SCAIR is the retrieval engine. Vizine et al. (2005) [8] showed an evolutionary algorithm that mines the web searching for documents according to group user's interests. They introduce an automatic keyword extraction method and a GA to improve the web search. The previous two techniques were designed to be used in an academic virtual community described as a scientific paper collection. The web mining frame work in [9] was again based on genetic search. The fitness function they used in their work depends on the link quality  $F(L)$  and the page quality  $F(P)$  to calculate the mean quality  $M_q$ . In [10], a GA based strategy is designed for finding association rules without user-specified threshold for minimum support. The relative confidence is used as the fitness function. The researchers claim that the computation cost is reduced, and interesting association rule is generated. In [11], GA is applied to perform web content mining. The used fitness function was support probability ratio of each individual. The researchers propose performing cross-over using "And" operator between two selected individuals. In [12], GA for search over XML of different domains is investigated. A steady state tournament selection Microbial is used. Thada and Jaglan in [13] utilize GA in web information retrieval. Mountford coefficient is used for Web-IR. The researchers claim that GA is fruitful in searching and ranking the retrieved documents according to their relevancy to the user query. However, most of the previous research did not investigate the role of the fitness function and its effects on the accuracy of the obtained results (i.e. the degree of relevance between the retrieved

documents and the entered query). The other important point is how to minimize the time needed to retrieve the most relevant documents.

In this work, we will tackle the problem of retrieving most relevant documents to user needs (query) and enhancing retrieving time. To accomplish these goals, island genetic algorithm (IGA) is used to achieve parallelism (each island is executed independently on different server) and improve retrieving time. Four islands are constructed, each of which has its own specifications (fitness function and selection method). Different specifications for each island are proposed to study the effect of these specifications on the accuracy of the obtained results. In addition to that, using different specifications for each island will also supports the idea of genetic variation produced by IGA (i.e. each island will retrieve different set of documents). The output of each island is passed to a decision making stage DMS, where the obtained results (retrieved documents) from the four islands are combined and ranked using cosine similarity measure. Finally, the ranked documents are sorted in descending order. The suggested approach allows the user to specify similarity threshold. The documents with similarity measure  $\geq$  threshold will be retrieved. In this work, threshold=0.8

The rest of this paper is organized in 5 sections. In section 2, island genetic algorithm is explained. Methodology of the suggested approach is demonstrated in section 3. The experimental results and evaluation are explained in section 4. Finally, we concluded in section 5.

## II. ISLAND GENETIC ALGORITHMS (IGAS)

Multiple efforts were applied to make simple GAs faster, one of the most promising approaches is to use parallel implementations (IGAs). Island genetic algorithm is a distributed model of genetic algorithm where each island executes GA independently with its sub-population [14][15]. In IGA, each island could evolve independently (with its own subpopulation), and could have different genetic operators [14][15]. Therefore, each island could follow a different searching path over the search space causing the generation of genetic diversity (i.e. each island could reach different solutions). In addition, the parallel nature of IGA will accelerate retrieving the documents relevant to user need (based on the entered query). These are the main points that behind using IGA in retrieve the most relevant documents from the WWW. As simple GAs deal with a single population, IGAs deal with a number of populations where each island represents one population. The main steps of a simple GA (selection, crossover, and mutation) are applied in each subpopulation (island) independently. In IGA, individuals are allowed to migrate to another island [16]. The main steps of simple GA are [17]:

- 1) Specify proper coding for each chromosome.
- 2) At the first generation, generate a set of random

population of chromosomes.

- 3) Repeat steps 4 and 5 until the termination criteria is satisfied
- 4) Apply the fitness function on each chromosome to find its fitness value.
- 5) Apply crossover and mutation to the selected chromosomes to generate the new population.

There are many ways to implement the selection operator. In this work, random tournament selection and unbiased tournament selection are used in addition to elitism [18]. Elitism is used to pass the best chromosome through all generations to save it from loss during the reproduction operation.

## III. METHODOLOGY

The need for high speed search on the tremendously rich knowledge repository, which is the WWW, is behind the main objective of this research. In this work, IGA is proposed to be used for web information retrieval. Four islands are suggested to be used (see Fig. 1). Each island has its own selection method and fitness function to achieve genetic variation. IGA could be implemented in parallel leading to a faster search through the WWW. In this work, we did not perform chromosome migration between islands to reduce traffic loading, and to study the behavior of each island independently. Finally, the results obtained by the 4 islands are ranked and combined (using a suggested decision making approach) to get the most relevant web pages. Using 4 islands, with different selection methods and fitness functions, help in studying the behavior of each island and decide the most suitable technique for web retrieving.

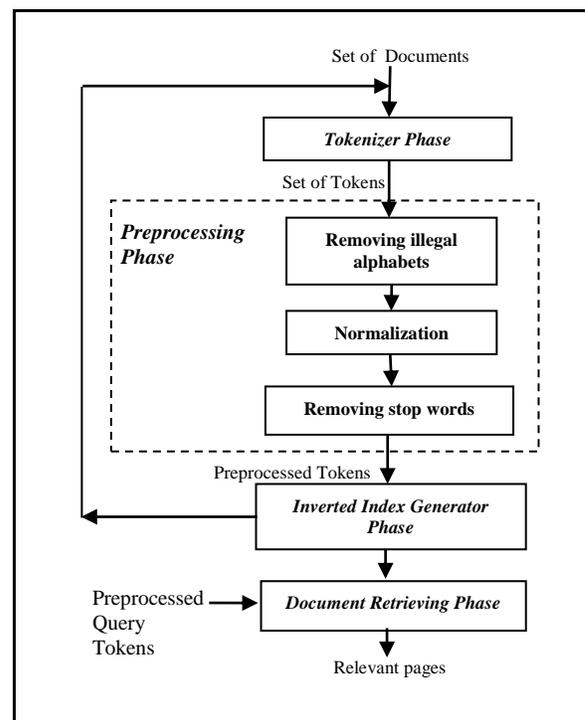


Fig. 1. The proposed system phases

The suggested approach mainly consists of four phases(as shown in Fig. 1): tokenizing phase, preprocessing phase, indexing phase, and document retrieving phase

#### A. Preprocessing Phase

In this research the following preprocessing tasks are performed:

- 1) *Removing illegal alphabets and irrelevant information:* each language consists of set of alphabets. In this step, remove each alphabet  $\notin$  to the language alphabets (in this research, the used language is English). Also, it is desirable to remove information that is not a part of document main topic, such as banner ads. This kind of information is not desirable to be indexed, otherwise, poor results may appear.
- 2) *Stop-words removal:* remove the frequently occurring words in the text (such as, about, a, is, for, the, what, etc.). These words help in constructing sentences but have slight value in document retrieving.
- 3) *Normalization (letters case unification):* All the letters of the texts are converted to the lower case.

#### B. Inverted Index Generator Phase

In web information retrieval, it is not possible to scan each web page (every time) to find whether it contains the query terms or not. Instead of that, a data structure, called inverted index, is used to speed up the searching process. In this work, an inverted index table has been built where each term (token) is associated with a list of all documents that contain that term or its derivatives.

#### C. Document Retrieving Phase

As mentioned before, the suggested retrieving module depends on the IGA to achieve parallelism (to speed up the retrieving process). It mainly consists of four islands, each of which works independently from the others to achieve parallel search through the WWW. Each island can be used in a separate server (depending on different index tables), then their results are combined with each other according to a filtering stage suggested in this research. In this work, the same inverted index is used with the four islands for comparison purposes (to compare the behavior of the 4 islands under the same circumstances). Fig. 2 illustrates the document retrieving phase.

In this phase, the user enters the query into the proposed system, to start the genetic search for the most relevant documents to the entered query. At first, the entered query is tokenized and preprocessed to determine the main terms (keywords) that will be used to retrieve the relevant documents. These terms will be passed to the genetic search stage. In this stage, an initial population of chromosomes is randomly generated, where each chromosome is a string of 0's and 1's representing the presence or absence of a query term in a document. Consequently, the length of the chromosome for a particular query will vary depending on number of the entered query keywords. Actually each chromosome will

indicate set of documents that might be relevant or irrelevant to the entered query. The fitness function determines the degree of relevance for the initial population depending on the number of query's keywords that occurs in a document. The next step is selecting the suitable chromosomes depending on their fitness values to be the parent of the new children by applying the crossover operator (1 point).

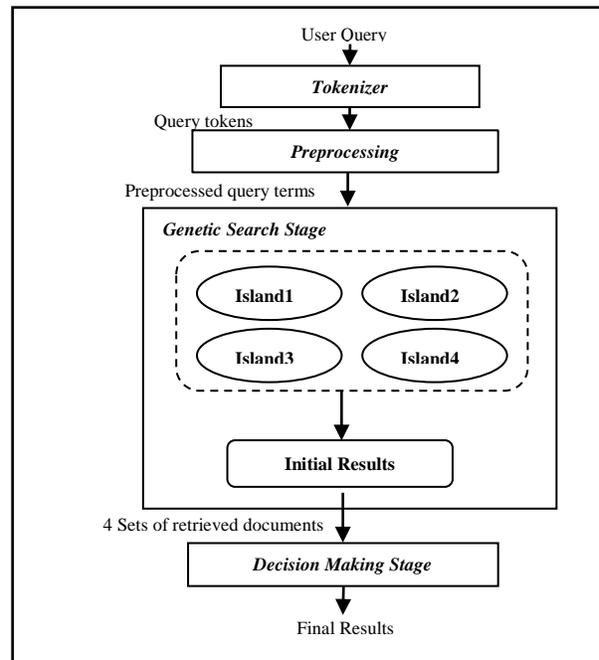


Fig. 2. Document retrieving phase

Elitism is an important process in the genetic search, which is also used in the four suggested islands. It passes the best two chromosomes to the new generation directly. Mutation is applied to some chromosomes in order to delay convergence (handle premature convergence).

#### D. Genetic Search Stage

This stage consists of four different islands. The four islands are similar in the way they apply the one-point crossover operator, elitism, and mutation (bit flipping) on chromosomes. The islands use different selection methods and fitness functions. Random tournament selection is used in this work because it needs small time complexity. It is also easy to be implemented in parallel environment [18]. In random tournament selection, k chromosomes (in this work, k=2) are randomly chosen, and the one with the highest fitness value is selected to be a parent of the new individual in the next population. Tournament selection could suffer from selection bias [18]. Therefore, unbiased tournament selection is used as another selection method, to overcome selection bias problem. The algorithm of unbiased tournament selection is simple. For population of size S (in this work, S=30), generate k random permutations in the range [1,S]. Arrange the k permutations as table of k rows and S columns. Each cell contains the fitness value of the corresponding individual. Compare the

fitness value in each column (for the k individuals), and specify the individual with the best fitness value (in this work, the highest fitness value is the best one) per column. This will ensure that the best individual is selected k times in the mating, and worst one is not selected [18].

The used fitness functions are Jaccard's coefficient in (1), and Ochiai coefficient (also called Ochiai-Barkman coefficient, or Otsuka-Ochiai coefficient) [19] in (2).

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (1)$$

$$O(X,Y) = \frac{|(X \cap Y)|}{\sqrt{|X| \times |Y|}} \quad (2)$$

Where X is the query and Y is a document. Both are represented by weightless keywords (terms). |X| indicates number of elements in X. Example (1) explains how to calculate the fitness value of each chromosome according to Jaccard's coefficient.

**Example (1):** assume that the user entered a 6-keywords query. The island randomly generates a chromosome, let it be "111010". This chromosome represents the documents that contain the 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup> and 5<sup>th</sup> query's keywords at least. According to Jaccard's coefficient, documents that contain more query-keywords will get higher fitness values. Assuming that there are three documents d1, d2 and d3, where d1 contains the 4 query-keywords represented by the chromosome, d<sub>2</sub> contains 5 of the query-keywords and d<sub>3</sub> contains all of the query-keywords, then by applying (1), the fitness value of the three documents will be: 0.6667, 0.8333 and 1 respectively. The average of the three fitness values represents the fitness value of the chromosome.

We have to emphasize that the value of Jaccard's coefficient  $\in [0, 1]$ , where 0 represents the absence of all query's keywords in the document, and 1 represents the presence of all query's keywords in the document. Any other value between 0 and 1 means the presence of some of the query-keywords in the document, hence, a fraction is assigned to the document. Example (2) explains how chromosomes get their fitness values according to Ochiai's coefficient.

**Example (2):** The same assumptions of island-one example are employed, where the query consists of 6 keywords, and the assumed chromosome is "111010". Same to Jaccard's coefficient, Ochiai's coefficient gives higher fitness values to the documents that contain more query keywords. By applying (2), the fitness value of the same three documents d<sub>1</sub>, d<sub>2</sub> and d<sub>3</sub> mentioned in island-one's example will be: 0.8165, 0.9129 and 1 respectively. The average of the three fitness values is considered as the fitness value of the chromosome.

As in Jaccard's coefficient, the value of Ochiai's coefficient  $\in [0, 1]$ , where 0 denotes the absence of all query-keywords in the document, and 1 denotes the presence of all query-keywords in the document, and any

other value between 0 and 1 denotes the presence of some of the query-keywords in the document.

The specifications of the 4 islands are shown in Table I.

TABLE I THE SPECIFICATIONS OF THE FOUR ISLANDS ARE

| Island # | Selection Method |          | Fitness function |               |
|----------|------------------|----------|------------------|---------------|
|          | Random           | Unbiased | Jaccard's coe.   | Ochiai's coe. |
| 1        | X                |          | X                |               |
| 2        | X                |          |                  | X             |
| 3        |                  | X        | X                |               |
| 4        |                  | X        |                  | X             |

After entering the user's query to the system, the genetic search stage is activated, where the four islands will search independently (as illustrated in Fig. 2). The last generation is the generation that will generate the initial results, which will be fed to the decision making phase.

As the optimal solution is not always reached, the proposed system offers an important option to the user, which is determining the fitness value of the last generation's chromosomes that will be considered to compute the initial results.

At the end of the second genetic search stage, a collection of documents is represented as the final result of the proposed system. These documents are ranked depending on the degree of relevance to the entered query. In this research, the vector space model is used to assign a weight to each indexed term. The degree of relevance is calculated using cosine similarity method (see (7)). Equations (3 and 4) are used to find the frequency of term *i* in document *j* (*tf<sub>ij</sub>*), and *term frequency-inverse document frequency* (*idf<sub>i</sub>*) respectively. In (4), N represents the total number of document in the collection; *df<sub>i</sub>* is the number of documents that contain the term *t<sub>i</sub>*. |V| is the vocabulary size of the collection.

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \dots, f_{|V|j}\}} \quad (3)$$

$$idf_i = \log \frac{N}{df_i} \quad (4)$$

The weight for documents is calculated as in Eq. (5).

$$w_{ij} = tf_{ij} \times idf_i \quad (5)$$

In vector space model, queries *q* has the same representation of documents. Thus, the weight of each term *i* in the query *q* is calculated as in Eq. (6).

$$w_{iq} = \left( 0.5 + \frac{0.5 f_{iq}}{\max\{f_{1q}, f_{2q}, \dots, f_{|V|q}\}} \right) \times \log \frac{N}{df_i} \quad (6)$$

$$\cosine(d_j, q) = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}} \quad (7)$$

After assigning a weight for the indexed terms, which are presented either in the document or in the query, equation (7) is applied to calculate the relevance degree

between a document and a query depending on the cosine similarity.

E. Decision Making Stage (Filtering)

This stage is the final one in the proposed system. As mentioned before, each island of the four islands works independently and generates its own results. Therefore, the results of the four islands need to be combined. Thus, the proposed system contains a decision making stage (DMS) that merges the results of the four islands, eliminates any duplications (in case of duplication, keep only the document with the highest rank), then order the retrieved documents depending on their ranks (cosine values). The document rank indicates its degree of relevance to the query. Fig. 3 illustrates the *filtering process*. The suggested approach allows the user to specify similarity threshold. All documents with similarity measure  $\geq$  threshold will be retrieved. In this work, threshold =0.8.

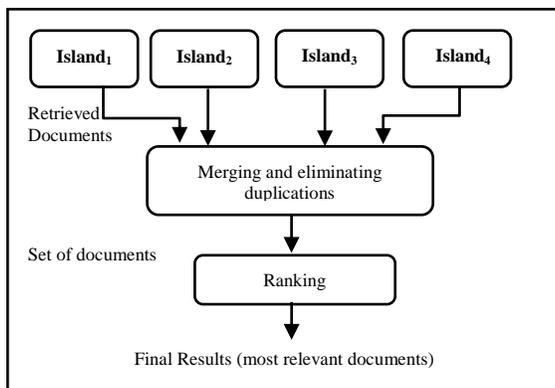


Fig. 3. Decision making stage

IV. ASSESSMENT OF EXPERIMENTAL RESULTS

To evaluate the proposed system, 5-fold cross validation is applied on Cranfield dataset (Cranfield dataset contains 1400 documents). Twelve queries (out of original 225 queries) are chosen for simulation purposes ( $Q_2, Q_3, Q_5, Q_8, Q_{28}, Q_{34}, Q_{38}, Q_{45}, Q_{47}, Q_{71}, Q_{203}, Q_{204}$ ). These queries are chosen due to the variation in number of terms and covering different topics. Documents and queries are preprocessed, as mentioned previously. The inverted index is built depending on terms extracted from the documents. To study the behavior of each island, the same initial population is used, in addition to fixing the population size (population size=30). As stopping condition, number of generations is set to 20. The small number of generations is due to the population size, chromosome size and the search space. Threshold fitness values of chromosomes in the last generation, crossover and mutation probabilities ( $P_c=0.8$  and  $P_m=0.1$ ).

The behavior of the four islands is compared from number of retrieved documents (with cosine similarity $\geq 0.8$ ) point of view. Sometimes such comparison is not enough since number of retrieved documents does not reflect the good or bad behavior of the retrieving system. Therefore, another comparison is made, which shows the average

cosine similarity measure of the results of the four islands. From Figs (4 and 5), it is clearly seen that island 4 shows the best behavior from both number of retrieved documents and the average cosine similarity measure of the retrieved documents (the best documents for 7 queries out of 12). Island 1 and 2 both use the same selection method (random tournament selection), but island 1 uses Jaccard's coefficient as fitness function, while island 2 uses Ochiai's coefficient. Also, islands 3 and 4, both use the same selection method (unbiased tournament selection), but island 3 uses Jaccard's coefficient as fitness function, while island 4 uses Ochiai's coefficient. By comparing the behavior of the two islands, island 1 shows better performance. By comparing islands 3 and 4, island 4 shows better performance. This indicates that, Ochiai's coefficient generates better results when used with unbiased tournament selection methods, while Jaccard's coefficient have better performance when used with random tournament selection method. Finally, by comparing the behavior of the suggested approach (DMS) with the behavior of the 4 islands, it is clearly seen that DMS outperforms the 4 islands behavior (see Figs (4 and 5)). The average cosine similarity measure of the retrieved documents using DMS is approximately 0.9.

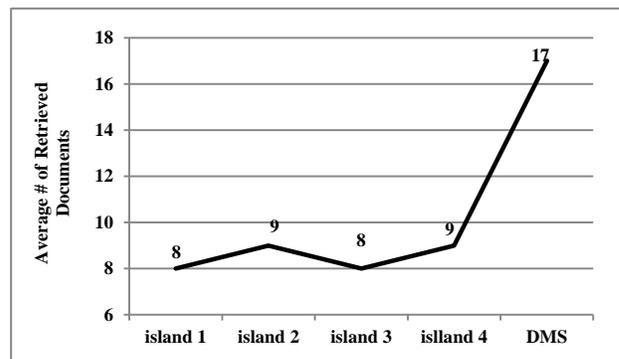


Fig. 4. Average number of retrieved documents with similarity  $\geq 0.8$

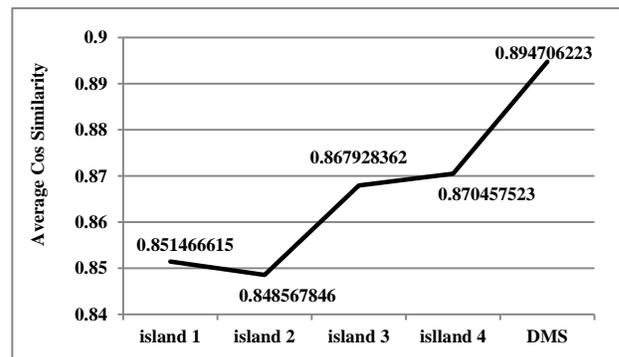


Fig. 5. Average similarity measure for documents with similarity $\geq 0.8$

The cosine similarity of 11 queries out of 12 is shown in figs (6 and 7). The results of  $Q_{38}$  are very weak (only islands 1 and 4 retrieve one document with cosine similarity measure  $\geq 0.8$ ). Therefore, it is not included in the Figs. The results shown in these two figures are constructed as follows:

-For each island,

1. Rank the retrieved documents, and specify 10 documents with the heist similarity measure.
2. From these 10 documents, get the documents with similarity measure  $\geq 0.8$ .

-For DMS, merge the four set of retrieved documents (resulted from 2), and rank these documents.

From Figs (6 and 7), it is clearly seen that DMS outperforms each of the four islands from both, number of retrieved documents, and the degree of relevance of the retrieved documents to the user query, point of views.

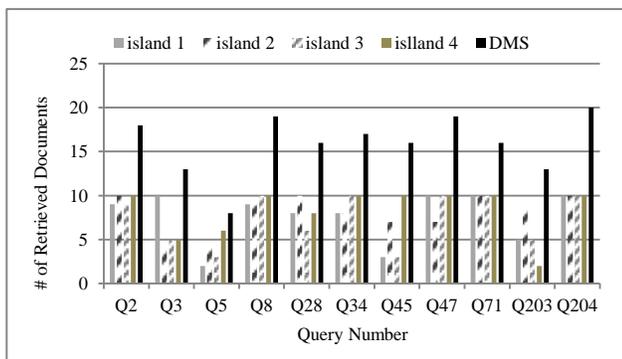


Fig. 6. Number of retrieved documents with cosine similarity  $\geq 0.8$  for 11 queries

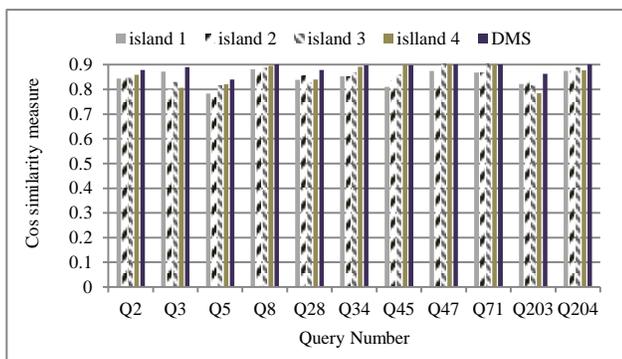


Fig. 7. Similarity measure for 11 queries with cosine similarity  $\geq 0.8$

## V. CONCLUSION

When query is entered to a search engine, millions of web pages are sifted to find the relevant ones. But search engines could retrieve relevant and irrelevant documents, or may not reach all relevant documents. Therefore, we suggest using IGA since this might maintain genetic diversity. In this work, each island retrieves different set of relevant document, which proves genetic diversity in IGA. The suggested approach, DMS, utilizes the genetic diversity nature of IGA. The documents retrieved by the four islands are combined and ranked based on cosine similarity measure. The documents with similarity measure  $\geq 0.8$  are chosen. Fig. 6 shows that by applying DMS, more documents (with similarity measure  $\geq 0.8$ ) are retrieved compared to the number of documents retrieved by each island separately (for the same similarity measure). Fig. 7, on the other hand, shows that DMS improves the type of retrieved documents (i.e. retrieve documents more relevant

to user needs), which evince the main objective of this work. As future work, semantic features will be used in additional islands to improve the relevance degree. Also, chromosome migration will be applied to check the migration effect on the speed and accuracy of the retrieving process.

## REFERENCES

- [1] Chen and Michael, Hsichun and Chau, "Web mining: machine learning for web applications," Annual Review of Information Science and Technology, Vol. 38, 2004, pp. 289-329.
- [2] Manning, Christopher D., Raghavan, Prabhakar and Schütze, Hinrich. Introduction to information retrieval, Cambridge University Press, 2008.
- [3] Herrouz, Abdelhakim, Khentout, Chabane and Djoudi, Mahieddine, *Overview of web content mining tools*, The International Journal of Engineering And Science (IJES), Vol. 2, 2013, pp. 1-6.
- [4] Johnson, Faustina and Gupta, Santosh K., "Web content mining techniques: a survey," International Journal of Computer Applications, Vol. 47, 2012, pp. 44-50.
- [5] Picaroungne, F., Monmarché, N., Oliver, A. and Venturini, G., "Web mining with a genetic algorithm," 11<sup>th</sup> International World Wide Web Conference, Honolulu, Hawaii, 7-11 May 2002.
- [6] Vallim, Max Streicher and Coello, Juan Manuel Adán, "An agent for web information dissemination based on a genetic algorithm," International Conference on Systems, Man and Cybernetics, IEEE Press, 2003.
- [7] Kim, Sun and Zhang, Byoung-Tak, "Genetic mining of HTML structures for effective web-document retrieval," *Applied Intelligence*, Kluwer Academic Publishers, Vol. 18, 2003, pp. 243-256.
- [8] Vizine, André L., Castro, Leandro N. de and Gudwin, Ricardo R., "An evolutionary algorithm to optimize web document retrieval," International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2005.
- [9] Marghny, M. H. and Ali, A. F., "Web mining based on genetic algorithm," Cairo : AML 05 Conference, 2005. pp. 82-87.
- [10] Yan, Xiaowei, Zhang, Chengqi and Zhang, Shichao, "Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support," Elsevier, Expert Systems with Applications, Vol. 36, 2009, pp. 3066-3076.
- [11] Sabnis, Vikrant and Thakur, R. S. "GA based model for web content mining," IJCSI International Journal of Computer Science Issues, Vol. 10, 2013, pp. 308-313.
- [12] Tungal, Dipali and Potgantwar, Amol D, "Investigation of web mining optimization using microbial genetic algorithm," Journal of Engineering Research and Applications, Vol. 4, 2014, pp. 593-597.
- [13] Thada, Vikas and Jaglan, Vivek., "Use of genetic algorithm in web information retrieval," International Journal of Emerging Technologies in Computational and Applied Sciences, Vol. 7, 2014, pp. 278-281.
- [14] Whitley, D., Rana, S. and Heckendorn, R. B. "Island model genetic algorithms and linearly separable problems," Evolutionary Computing: Proc. AISB Workshop, Lecture Notes in Computer Science. Vol. 1305, 1997, pp. 109-125.
- [15] Belal, Mohamed A. and Haggag, Mohamed H. "A structured-population genetic-algorithm based on hierarchical hypercube of henes expressions," International Journal of Computer Applications, Vol. 64, 2013, pp. 5-18.
- [16] Engelbrecht, Andries P. England, Computational intelligence: an introduction, John Wiley and Sons Ltd., 2002.
- [17] Simon, Dan, Evolutionary optimization algorithms, Wiley, 2013.
- [18] Xinjie Yu, Mitsuo Gen, Introduction to Evolutionary Algorithms, Springer Science & Business Media, 2010.
- [19] Choi, Dongjin, Kim, Jeongin and Kim, Pankoo. A method for normalizing non-standard words in online social network services: a case study on twitter. [ed.] Phan Cong Vinh, et al. *Context-Aware Systems and Applications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer International Publishing, Vol. 128, 2014, pp. 359-368.