

Research Article

State-of-the-Art Review on Relevance of Genetic Algorithm to Internet Web Search

Kehinde Agbele,¹ Ademola Adesina,¹ Daniel Ekong,² and Oluwafemi Ayangbekun³

¹ Department of Computer Science, Soft Computing and Intelligent Systems Research Group, University of the Western Cape, Private Bag X17, Bellville, Cape Town, South Africa

² Department of Mathematical Sciences (Computer Science Option), Ekiti State University, Ado-Ekiti, PMB 5363, Ado-Ekiti, Ekiti State, Nigeria

³ College of Information and Communication Technology, Crescent University, Abeokuta, Ogun-State, Nigeria

Correspondence should be addressed to Kehinde Agbele, agbelek@yahoo.com

Received 10 April 2012; Revised 12 September 2012; Accepted 26 September 2012

Academic Editor: Cheng-Jian Lin

Copyright © 2012 Kehinde Agbele et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

People use search engines to find information they desire with the aim that their information needs will be met. Information retrieval (IR) is a field that is concerned primarily with the searching and retrieving of information in the documents and also searching the search engine, online databases, and Internet. Genetic algorithms (GAs) are robust, efficient, and optimized methods in a wide area of search problems motivated by Darwin's principles of natural selection and survival of the fittest. This paper describes information retrieval systems (IRS) components. This paper looks at how GAs can be applied in the field of IR and specifically the relevance of genetic algorithms to internet web search. Finally, from the proposals surveyed it turns out that GA is applied to diverse problem fields of internet web search.

1. Introduction

There is a virtual explosion in the availability of electronic information. The advent of the Internet or World Wide Web (WWW) has brought far more information than any human being can absorb. The goal of IR systems is to assist user to organize and store such information and retrieve useful information when a user submits a query to the IR systems. To resolve this problem, many research communities have implemented diverse techniques such as full text, inverted index, keyword querying, Boolean querying, knowledge-based, neural network, probabilistic retrieval, genetic algorithm, and machine learning. Now, increasing numbers of people use web search engines which enable them to access any kind of information from the Internet in order to formulate better, well-informed decisions. However, the ability of search engines to return useful and relevant documents is not always satisfactory. Often users need to refine the search query several times and search through large document collections to find relevant information.

But, according to [1], the results returned by the search engine may not be relevant to the users' information needs and, hence users need to modify and reformulate their queries.

The focus of IR is the capability to search for information relevant to individual user's needs within a documents collection which is relevant to the user's query. According to [2], the authors stated that user is in need of information. The work reported in Agbele et al. [3] describes access to information as an important benefit that can be achieved in many areas including socio-economic development, education, and healthcare. In healthcare, for example, access to appropriate information can minimize visits to physicians and period of hospitalization for patients suffering from chronic conditions, such as asthma, diabetes, hypertension, and HIV/AIDS. Agbele method examines the opening of health information system based on ICT as one fundamental healthcare application area, especially within the context of the Millennium Development Goals to improve the management and quality of healthcare for development at

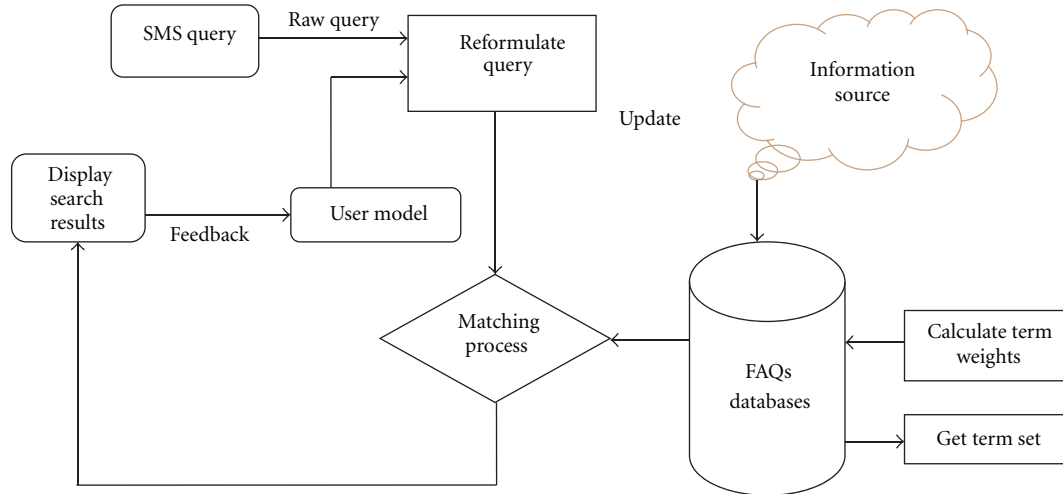


FIGURE 1: Important Processes in Web Information Retrieval-framework.

lower cost. It is the responsibility of a user to formulate query and send the query to the search engine (or IRS).

IR system searches for the matches in the document databases and, thus, retrieves search results of the matching process. However, based on the relevance, the user will then evaluate and display the search results. The relevance of the document is very important to the user. If the user feels that it is a relevant document, he finishes the search while else user continues to search in the document database by reformulating the query until the relevant documents that will satisfy users' information needs are retrieved.

GA is a probabilistic algorithm simulating the process of natural selection of living organisms and finally coming up with an approximate solution to a problem [4–6]. In GA implementation, the search space is composed of candidate solutions (called individuals or creatures) to an optimization problem to evolve better solutions; each represented by a string is termed chromosome. Each chromosome has an objective function value, called fitness. A set of chromosomes together with their associated fitness is called a population. This population, at a given iteration of the genetic algorithm, is called a generation. In each generation, the fitness of every individual in the population is evaluated from the current population based on their fitness value and modified to form a new population. The new population is then used in the next iteration of the algorithm.

GA terminates when either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. The working of the genetic algorithm depends upon the constraint of how well we choose our initial random keywords.

The rest of the paper is organized as follows: Section 2 discusses important processes within IR components while Section 3 reviews the relevance of GAs to Internet web search and its applications in IR. Section 4 gives the conclusion.

2. Components of an Information Retrieval System

In the framework for IR as depicted in Figure 1, the user gives a mobile SMS-query (raw query) and the query is reformulated in order to improve the predicted relevance of the retrieved document. The reformulated query is searched against the databases. The IR system searches for the matches in the document databases and, thus, retrieves search results of the matching process. Based on the relevance, the user will then display the search results. The relevance of the document is very important to the user. If the user feels that it is a relevant document, he finishes the search while else user continues to search in the document database by reformulating the query until the relevant documents that will satisfy users' information needs are retrieved. Hence, user query reformulations will apply by updating its model. A user model is a stored knowledge about a particular user. A simple model consists usually of keywords describing user's area of interest. Sort those documents according to TFIDF approach. The documents which have the high retrieval status value (RSV) are considered as the top ranked documents.

The two main components in the proposed IR system framework are document databases and reformulated query processing system. The document databases stores the databases related to documents and the representations of their information contents based on TFIDF approach. An SMS-query keyword term is also associated with this component which automatically generates a representation for each document by extracting the frequency of the SMS-query keyword terms from the document contents. The reformulated query processing System consists of two sub-systems: Searching-Matching Unit and Displaying-Ranking Unit.

Searching unit allows user to search the documents from the document database, and matching unit does a comparison of all documents against the user's query. To

improve the predicted relevance of the retrieved document, the reformulated query is searched against the databases. Searching-Matching unit does a thorough search and finds out which documents match the user query. This unit retrieves almost all the documents that match either part or whole of the entire query, that is, the unit retrieves relevant amid nonrelevant documents. Displaying unit displays the search results based on relevance of the documents to user information needs, and ranking unit ranks the document according to the relevance of the user query. Displaying-Ranking unit does a detailed display of search results and find out which documents have high RSV to be considered as the top ranked documents. Therefore, IR system ranks the documents according to the RSV between document and the query. If a document has got high RSV, that document is closer to the query.

Generally, IR system ranks the list of documents in the descending order. After processing the query effectively, the top most relevant documents are retrieved, and it is given to the user. Though relevance feedback is one of the processes in an IR system that seeks to improve the system's performance based on a user's feedback, it modifies queries using judgments of the relevance of few, highly-ranked documents and has historically been an important method for increasing the performance of IR systems. Specifically, the user's judgments of the relevance or nonrelevance of some of the documents retrieved are used to add new terms to the query and to reweigh query terms. For example, if all the documents, that the user judges as relevant contain a particular term, then, that term may be a good one to add to the original query. It is made known that relevance feedback has improved the system's overall performance by 60% to 170% for different document collections [7]. Given the apparent effectiveness of relevance feedback techniques, it is important that any proposed model of information retrieval includes these techniques.

3. Literature Review

In designing GA, there are three main components which had to be taken into consideration [8]. This research study presents an application of GA as relevant feedback method aiming to adapt keywords weights. In the following, we shall give the three main components; the first one is coding the problem solutions; subsequent is to find a fitness function that can optimize the performance, and, finally, the set of parameters includes the population size, population structure, and genetic operators. Genetic algorithms are generally used for solving timetabling [9], stock marketing [10], and job scheduling [11] problems. A Genetic Algorithm (GA) is used as a powerful tool to search solutions in the domain of relevant features and is suitable for the IR for the justifications discussed in [12, 13].

Ever since the advent of the public network Internet, the quantity of available information is rapidly rising. One of the most important uses of this public network is to find information. In such a huge and unstable information collection, today's greatest problem is to find relevant information. It is necessary to improve the existing search

agents. Diverse proposals that use GAs in Internet search with this aim are put forward.

According to [14], proposed the problems of existing internet search engines are examined, and, hence, the need for a novel design is warranted. To make search engines work more efficiently, new thoughts on how to improve existing Internet engines are presented, and then an adaptive technique for Internet metasearch engines with a multiagent, especially the mobile agent, is presented. In the technique, the understanding between stationary and mobile agents is used as an indication to make it more competence. However, the metasearch engine gives the user needed documents based on the multiagent mechanism. The combination of the results obtained from the search engines in the network is done in parallel. In this regards, a feedback mechanism gives the metasearch engine the user's suggestions about the found documents, which leads to a new query using a genetic algorithm.

Reference [15] proposed a new technique that, given a keyword query, on the fly generates new pages, called composed pages, which include all query keywords. The composed pages are generated by extracting and mending together relevant pieces from hyperlinked Web pages and retaining links to the original Web pages. To rank the composed pages, both the hyperlink structure of the original pages and the associations between the keywords in each page are considered. The proposed technique is used to evaluate heuristic algorithms to efficiently generate top composed pages.

Reference [16] uses GAs for user modelling of adaptive and exploratory behaviour in an information retrieval system. Maleki Dizaji choices of the underlying genetic operators are mentioned as a major drawback in the use of GAs. However, GA is primarily used to solve optimization problems; their use for information retrieval is gaining ground.

Reference [17] proposed an improved GA which solves the issues in two-generation competitive genetic algorithm. So, it changes the selection technique of the simple genetic algorithms and improves search efficiency, but local best search ability cannot be improved. The proposed algorithm does the adaptive adjustment of the mutation probability and the position of crossover and mutation probability in chromosomes.

According to [18] they surveyed an effective GA that monitor the success of internet database management system by combining functionality, quality, and complexity of query optimizer for finding good solutions to the problem.

Also [19] they proposed a framework for web mining, the applications of data mining and knowledge discovery techniques to data collected in WWW, and a genetic search for search engines. The authors defined an evaluation function that is a mathematical formulation of the user request to define a steady state GA that evolves a population of pages with binary tournament selection. This approach chooses one crossover position within the page randomly and exchanges the link after that position between both individuals (web pages).

According to [20] they proposed a class-based internet document management and access system, ACRID; it

uses machine learning techniques to organize and retrieve Internet documents. The knowledge acquisition process of ACRID automatically learns the classification knowledge from classified-internet documents into one or more classes. The two-phase search engine in ACRID will use the hierarchical structure for responding to user queries.

According to [21] they proposed and applied a dynamically terminated GA to generate page clippings from web search results. The page clipping synthesis (PCS) search method applies a dynamically terminated GA to generate a set best-of-run page clippings in a controlled amount of time. In the proposed approach the dynamically terminated GA yields cost-effective solutions compared with solutions reached by conventional GAs.

According to [22] they proposed an intelligent personal spider approach for Internet searching. The authors implemented Internet personal spiders based on best first-search and GA techniques. The used GA applies stochastic selection based on Jaccard's fitness, with heuristic-based crossover and mutation operators. These personal spiders dynamically take a set of user's selected starting homepages in the web, based on the existing links and keyword indexing.

According to [23] they proposed the use of genetic programming (GP) to derive approach for the combination of three different sources of evidence for ranking documents in the web search engines. The initialization method of the approach defines the method to create the initial population. Two methods can be adopted, grow or full, which represent small changes in the algorithm to construct the trees. The approach is useful for coping with search engines that can request diverse forms of queries for submission.

Consequently, there has been an increasing interest in the application of GA tools to IR in the last few years. The machine learning concept [24], whose aim is the design of system able to automatically acquire knowledge by themselves, seems to be interesting [22]. GAs are not specifically learning algorithms, but also offering a powerful and domain-independent search ability that can be used in many learning tasks, since learning and selforganization can be considered as optimization problems in many cases. As a result of this reason, the applications of GAs to IR have increased in the last few years. Among others, in the following, we shall examine some of the diverse proposals made in these fields in the last few years.

3.1. Clustering of Document and Terms. In this field, two approaches have been applied for obtaining user-oriented document clusters. According to [25], look for groups of terms appearing with similar frequencies in the documents of collection. The authors consider a GA grouping the terms without maintaining their initial order. The main features of the GA are as follows.

- (i) Representation scheme: two different coding schemes are considered to include division-assignment and separator methods.
- (ii) Initial population: the first generation of the chromosomes depends on the chosen coding, and the rest of individual are randomly generated.

- (iii) Operators: each operator has an application probability associated and is selected spinning the roulette. Different crossover and mutation operators are used.
- (iv) Fitness function: a measure of the relative entropy and Pratt's measure are two proposals adopted.

3.2. Matching Function Learning. The objective of matching function learning is to use a GA to generate a similarity measure for a vector space IR system to improve its retrieval efficiency for a defined user. This constitutes new relevance feedback beliefs since matching functions are adapted instead of queries. In this regards, two different variants have been proposed in the specialized literature.

- (i) Automatic similarity measure learning: according to [26, 27], they introduced a GA that automatically learn a matching function with relevance feedback. Besides, the similarity functions are represented as trees and a classical generational scheme; the usual GA crossover are considered.
- (ii) Linear combination of existing similarity functions: [28] propose a new weighted-matching function, which is the linear combination of different existing similarity functions. The weighting parameters are estimated by a genetic algorithm based on relevance feedback from users. The authors use real coding, a classical generational scheme, two-point crossover, and Gaussian noise mutation. Finally, the algorithm is tested on the Cranfield collection.

Automatic Document Indexing. The applications in this area adapt the descriptions of the documents in the documentary base with the aim of facilitating document retrieval in the face of relevant queries. According to [28] they propose a GA to derive the document descriptions. They choose a binary coding scheme where each description is a fixed length and a binary vector. The genetic population is composed of diverse descriptions for the same document. The fitness function is based on calculating the similarity between the current document description and each of the queries (for which the document is relevant or nonrelevant) by means of the Jaccard's index and, then, computing the average adaptation values of the description for the set of relevant and nonrelevant queries. In Gordon work, GA is considered quite unusual as there is no mutation operator, and the crossover probability is equal to 1. With regard to the selection scheme, the number of copies of each chromosome in the new population is calculated and dividing its adaptation value by the population average. Also [29] propose an algorithm for indexing function learning based on GA, whose aims to obtain an indexing function for the key term weighting of a documentary collection to improve the IR process.

Query Learning. This is the most extended group of applications of genetic algorithms in information retrieval. Every proposal in this group use genetic algorithms either like a relevance feedback method or like an Inductive Query By

TABLE 1: Comparison of Diverse Proposals that use genetic algorithm for internet search.

| Diverse proposals | Reason for GA | Chromosomes | Fitness function adopted | Genetic operators used |
|-------------------------------------|--|---|--|---|
| (Eissa and Alghamdi, 2005) [30] | Genetic algorithm is used to optimize the profiles whereas the relevance feedback is used to adapt it. | Represent a gene as a term, an individual as a document and the population as the profile. | $F(P_i) = k \sum S(D_{pk}, P_i) / \#D_p$. | Selection. |
| (Vallim and Coello, 2003) [31] | Combines user's feedback to new documents retrieved by the agent with a genetic algorithm. | Individuals represented by a query vector and its adaptation rate. | $Q = Q + \alpha f$ $F(Q) = f + \beta f$. | Two point crossover and Mutation operator. |
| (Li et al., 2000) [32] | Realize the scheduling strategy of agent manager. | Search space is represented as weight field in the search engine. Field are search parameters. | Adaptation function $\Phi(\text{agent}) = \Gamma(f, p, c, u, t)$. | One point Cross over and Single point Mutation. |
| (Caramia et al., 2004) [33] | Select a subset of original pages for which the sum of scores is large. | Chromosomes represent subsets of pages of bounded cardinality. Each page is a gene. | $Ff(c) = \alpha \cdot t_1(C) + \beta \cdot t_2(C) + \gamma \cdot t_3(C)$. | Single point crossover. |
| (Rocio et al., 2008) [34] | Evolving lofty quality query. | Chromosome is represented as a list of terms where each term corresponds to a gene. | Fitness (q) = $\max(\sigma(c, d_i))$ $d_i \in A_q$. | Roulette Wheel Selection, Single point crossover, One point mutation. |
| (Abe et al., 1999) [35] | For evolving information retrieval agents. | Genes are represented by the search parameters. | $F = (\text{SH}/\text{MH} + \text{SI}/\text{MI}) * (1 - \text{ST}/\text{TL}) + (1 - \text{ME}/\text{MM})$. | Selection uses ranking strategy, Uniform crossover, and Single Point Mutation. |
| (Martin-Bautista et al., 1999) [36] | Adaptive internet information retrieval. | Each gene represents a fuzzy subset of the document set by means of a Keyword term and number of occurrences in a document. | $f_i^j = f_i^{j-1} + P_i^j - L_i^j$ where P_i^j and L_i^j is the pay off of life tax and chromosome number respectively. | Random selection, Double point Crossover, Random Mutation. |
| (Marghny and Ali, 2005) [19] | Steady state genetic algorithm for optimizing web search. | Initial population is generated by heuristic creation operator which queries standard engines to obtain pages. | Fitness function evaluates web pages is a mathematical formulation of Link quality, Page quality and Mean quality function. | Binary tournament selection, Single point crossover. |
| (Cheng et al., 1998) [22] | GA implemented as a spider to find most relevant home pages in the entire internet. | Chromosomes represent all input home pages in a set. | Jaccard's coefficient function. | Heuristic based cross over, Simple mutation. |
| (Lin et al., 2002) [20] | Improving the searching performance. | Initial population represented by binary coding selected at random. | $F_d = (F_{\max} - F_{\min}) / F_{\text{aver}}$. | Fitness proportion selection, Adaptive adjusting crossover, Mutation operation range. |
| (Fan et al., 2003) [37] | Genetic Programming to the ranking function discovery problem leveraging the structural information of HTML documents. | Chromosomes represent html pages. | The fitness evaluation of each ranking tree is done at the level multiple queries. $P_{\text{Avg}} = \sum_{i=1}^{T_{\text{Rel}}} P_i / T_{\text{Rel}}$, $P_i = i / \text{Rank}_i$. | Single point Crossover, One point Mutation. |

TABLE 1: Continued.

| Diverse proposals | Reason for GA | Chromosomes | Fitness function adopted | Genetic operators used |
|-------------------------------------|---|--|--------------------------|--|
| (Milutinovic et al., 2000) [38] | Genetic search algorithms enable intelligent and efficient internet searches. | Chromosomes represent set of input Web sites given by a user. | Jaccard's Function. | Topic Mutation, Spatial Mutation, Temporal Mutation. |
| (Koorangi and Zamanifar, 2007) [14] | Query reformulation in search engine. | Initial population consists of first five keywords of the user dictionary. | CHK fitness function. | One point crossover, Inversion mutation operator. |

Example (IQBE) algorithm. The fundamental of relevance feedback lies in the fact that either users normally formulate queries composed of terms, which do not match the terms (which used to index the relevant documents to their needs) or they do not provide the appropriate weights for the query terms. The operation mode is involving and modifying the previous query (adding and removing terms or changing the weights of the existing query terms), with taking into account the relevance judgements of the documents retrieved by it, and constitutes a good way to solve the latter two problems and to improve the precision, and especially the recall of the previous query [39].

Therefore, IQBE was proposed as “a process in which searchers provide sample documents (examples), and the algorithms induce (or learn) the key concepts in order to find other relevant documents” [22]. This technique is a process for assisting the users in the query formulation process performed by machine learning techniques. It works by taking a set of relevant (and optionally, nonrelevant documents) provided by a user and applying an offline learning process to automatically generate a query describing the user's information needs. Besides, [40] propose a GA for learning queries for Boolean IR system. Although the authors introduce concept approach as a relevance feedback algorithm, the experimentation is actually closer to IQBE framework.

According to [41] they propose a similar GA to that of [25]. They use a real coding with the two-point crossover and random mutation operators (besides, crossover and mutation probabilities are changed throughout the GA run). The selection is based on a classic generational scheme where the chromosomes with a fitness value below the average of the population are eliminated, and the reproduction is performed by Baker's mechanism.

A comprehensive comparison of the diverse proposals prepared by different authors is summarized in Table 1.

4. Conclusion

This paper has dealt with the fundamentals of the information retrieval and genetic algorithm. Issues that can be solved using Genetic Algorithm and research areas in Internet web search are discussed in this paper. It also deals with diverse proposals in Internet web search which are promising and growing research areas. This paper examines the relevance

of genetic algorithm in diverse fields of internet web search some applications of genetic algorithms to information retrieval and a survey of the research works done in Internet web search area have been examined carefully, and the results so far have, thus, been very promising and encouraging.

References

- [1] F. G. Erba, Z. Yu, and L. Ting, “Using explicit measures to quantify the potential for personalizing search,” *Research Journal of Information Technology*, vol. 3, no. 1, pp. 24–34, 2011.
- [2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, New York, NY, USA, 1999.
- [3] K. Agbele, H. Nyongesa, and A. Adesina, “ICT and information security perspectives in E-health systems,” *Journal of Mobile Communication*, vol. 4, pp. 17–22, 2010.
- [4] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [5] K. A. DeJong, *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, University of Michigan, 1975.
- [6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, Machine Learning*, Addison Wesley, 1989.
- [7] G. Salton and C. Buckley, “Improving retrieval performance by relevance feedback,” *Journal of the American Society for Information Science*, vol. 41, no. 4, pp. 288–297, 1990.
- [8] L. M. Schmitt, “Fundamental study, theory of genetic algorithms,” *Theoretical Computer Science*, vol. 259, no. 1-2, pp. 1–61, 2001.
- [9] K. Milena, “Solving timetabling problems using genetic algorithms,” in *Proceedings of the IEEE 27th International Spring Seminar Electronics Technology: Meeting the Challenges of Electronics Technology Progress*, vol. 1, pp. 96–98, 2004.
- [10] L. Lin, L. Cao, J. Wang, and C. Zhang, “The applications of genetic algorithms in stock market data mining optimization,” in *Proceedings of the Capital Market*, CRC, Sydney, Australia, 2000.
- [11] W. Ying and L. Bin, “Job-shop scheduling using genetic algorithm,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1994–1999, October 1996.
- [12] J. F. Frenzel, “Genetic algorithms, a new breed of optimization,” *IEEE Potentials*, vol. 12, pp. 21–24, 1993.
- [13] L. Tamine, C. Chrisment, and M. Boughanem, “Multiple query evaluation based on an enhanced genetic algorithm,” *Information Processing and Management*, vol. 39, no. 2, pp. 215–231, 2003.
- [14] M. Koorangi and K. Zamanifar, “A distributed agent based web search using a genetic algorithm,” *International Journal of*

- Computer Science and Network Security*, vol. 7, no. 1, pp. 65–76, 2007.
- [15] R. Varadarajan, V. Hristidis, and T. Li, “Beyond single-page web search results,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 3, pp. 411–424, 2008.
- [16] S. Maleki-Dizaji, *Evolutionary learning multi-agent based information retrieval systems [Ph.D. thesis]*, Sheffield Hallam University, 2003.
- [17] J. Cheng, W. Chen, L. Chen, and Y. Ma, “The improvement of genetic algorithm searching performance,” in *Proceedings of 1st International Conference on Machine Learning and Cybernetics*, pp. 947–951, Beijing, China, November 2002.
- [18] M. Sinha and S. V. Chande, “Query optimization using genetic algorithms,” *Research Journal of Information Technology*, vol. 2, no. 3, pp. 139–144, 2010.
- [19] M. H. Marghny and A. F. Ali, “Web mining based on genetic algorithm,” in *Proceedings of the AIML O5 Conference*, CICC, Cairo, Egypt, December 2005.
- [20] S. H. Lin, M. C. Chen, J. M. Ho, and Y. M. Huang, “ACIRD: intelligent Internet document organization and retrieval,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, pp. 599–614, 2002.
- [21] L. C. Chen, C. J. Luh, and C. Jou, “Generating page clippings from web search results using a dynamically terminated genetic algorithm,” *Information Systems*, vol. 30, no. 4, pp. 299–316, 2005.
- [22] H. Cheng, C. Yi-Ming, R. Marshal, and Y. Christopher, “An intelligent personal spider (agent) for dynamic Internet/Intranet searching,” *Decision Support Systems*, vol. 23, no. 1, pp. 41–58, 1998.
- [23] T. P. C. Silva, E. S. de Moura, J. M. B. Cavalcanti, A. S. da Silva, M. G. de Carvalho, and M. A. Gonçalves, “An evolutionary approach for combining different sources of evidence in search engines,” *Information Systems*, vol. 34, no. 2, pp. 276–289, 2009.
- [24] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [25] A. M. Robertson and P. Willett, “Generation of equiproportional groups of words using a genetic algorithm,” *Journal of Documentation*, vol. 50, no. 3, pp. 213–232, 1994.
- [26] M. Gordon, “Probabilistic and genetic algorithms for document retrieval,” *Communications of the ACM*, vol. 31, no. 10, pp. 1208–1218, 1988.
- [27] W. Fan, M. D. Gordon, and P. Pathak, “Discovery of context-specific ranking functions for effective information retrieval using genetic programming,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 523–527, 2004.
- [28] P. Pathak, M. Gordon, and W. Fan, “Effective information retrieval using genetic algorithms based matching functions adaptation,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS ’00)*, January 2000.
- [29] W. Fan, M. D. Gordon, and P. Pathak, “Personalization of search engine services for effective retrieval and knowledge management,” in *Proceedings International Conference on Information Systems (ICIS ’00)*, Brisbane, Australia, 2000.
- [30] F. Eissa and H. Alghamdi, “Agent based information retrieval system,” in *Proceedings of the International Conference Proceedings*, pp. 265–279, 2005.
- [31] M. S. Vallim and J. M. A. Coello, “An agent for web information dissemination based on a genetic algorithm,” in *IEEE, International Conference on Systems, Man and Cybernetics*, vol. 4, no. 5–8, pp. 3834–3836, 2003.
- [32] W. Li, B. Xu, H. Yang, W. C. Chung, and C.-W. Lu, “Application of genetic algorithm in search engine,” in *Proceedings of the Proceedings of the International Conference on Microelectronic Systems Education (MSE ’00)*, pp. 366–371, IEEE, 2000.
- [33] M. Caramia, G. Felici, and A. Pezzoli, “Improving search results with data mining in a thematic search engine,” *Computers and Operations Research*, vol. 31, no. 14, pp. 2387–2404, 2004.
- [34] L. Rocio, L. Cecchini, M. Carlos, Lorenzetti, G. Ana, and M. Nelida, “Using genetic algorithms to evolve a population of topical queries,” *Information Processing and Management*, vol. 44, no. 6, pp. 1863–1878, 2008.
- [35] K. Abe, T. Taketa, and H. Nunokawa, “An efficient information retrieval method in WWW using genetic algorithms,” *ICPP Workshops*, pp. 522–527, 1999.
- [36] M. J. Martin-Bautista, H. Larsen, and M. A. Vila, “A fuzzy genetic algorithm approach to an adaptive information retrieval agent,” *Journal of the American Society for Information Science*, vol. 50, no. 9, pp. 760–771, 1999.
- [37] W. Fan, M. D. Gordon, P. Pathak, W. Xi, and E. A. Fox, “Ranking function optimization for efficient web search By genetic programming, an empirical study,” Department of Computer Science of Virginia Tech, Florida Universities, 2003.
- [38] V. Milutinovic, D. Cvetkovic, and J. Mirkovic, “Genetic search based on multiple mutations,” *IEEE Computer*, vol. 33, no. 11, pp. 118–119, 2000.
- [39] V. Rijsbergen, *Information Retrieval*, Butterworth, 2nd edition, 1979.
- [40] M. P. Smith and M. Smith, “The use of genetic programming to build Boolean queries for text retrieval through relevance feedback,” *Journal of Information Science*, vol. 23, no. 6, pp. 423–431, 1997.
- [41] J. J. Yang and R. R. Korfhage, “Query modification using genetic algorithms in vector space models,” *International Journal of Expert Systems*, vol. 7, no. 2, pp. 165–191, 1994.