Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

ELSEVIER International Journal of Approximate Reasoning 34 (2003) 241–264
www.elsevier.com/locate/ijar

A review on the application of evolutionary computation to information retrieval ☆

O. Cordón [a,*], E. Herrera-Viedma [a], C. López-Pujalte [b], M. Luque [a], C. Zarco [c]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, E.T.S.I. Informatica, C/Daniel Saucedo Aranda, s/n, Granada 18071, Spain*
[b] *Library Science Studies Faculty, University of Extremadura, Badajoz 06071, Spain*
[c] *PULEVA Food S.A. Camino de Purchil, 66, Granada 18004, Spain*

Received 1 January 2003; accepted 1 July 2003

Abstract

In this contribution, different proposals found in the specialized literature for the application of evolutionary computation to the field of information retrieval will be reviewed. To do so, different kinds of IR problems that have been solved by evolutionary algorithms are analyzed. Some of the specific existing approaches will be specifically described for some of these problems and the obtained results will be critically evaluated in order to give a clear view of the topic to the reader.

Keywords: Information retrieval; Evolutionary algorithms; Automatic indexing; Document clustering; Query definition; User profiles; Internet search agents; Image retrieval; Similarity functions; Web pages

☆ This research has been supported by CICYT under project TIC2002-03276 and by the University of Granada under project "Mejora de Metaheurísticas mediante Hibridación y sus Aplicaciones".

* Corresponding author. Tel.: +34-958-246143; fax: +34-958-243317.
E-mail address: ocordon@decsai.ugr.es (O. Cordón).

0888-613X/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2003.07.010

1. Introduction

We are actually living in the information age. This leads to the fact that any media organization has a computer system endowed with data bases, which can structure different information kinds in a correct way, and with hardware which allows it to efficiently store and access to this information. Unfortunately, the large size of these data bases has made the required effort to retrieve useful information increase significantly in the last few years.

Information retrieval (IR) tries to make a suitable use of these data bases, allowing the users to access to the information which is really relevant in an appropriate time interval [47]. Unfortunately, commercial IRSs, usually based on the Boolean IR model [53], have provided unsatisfactory results. Vector space, probabilistic and fuzzy models, which have been developed to extend the Boolean model [2], as well as the application of knowledge-based techniques, have solved some of these problems, but there are still some lacks [11]. In the last few years, an increasing interest on the application of artificial intelligence (AI)-based techniques to IR has been shown with the aim of solving some of those lacks.

One of the AI (or, more specifically, computational intelligence) areas with a considerable growth in the last decades is *evolutionary computation* (EC) [1], based on the use of models of evolutionary process for the design and implementation of computer-based problem solving systems. The different models which have been proposed within this philosophy are named in a generic way as *evolutionary algorithms* (EAs) [1].

In this paper, we review the application of EC to IR, analyzing the different kinds of IR problems that have been solved by EAs, describing some of the specific approaches proposed and critically evaluating the outcomes obtained.

To do so, the paper is structured as follows. In Section 2, some preliminaries are introduced by reviewing the basis of IR and EAs. Then, the different applications of EAs to IR are classified into different groups in Section 3. Sections 4–11 respectively show a review of the different proposals made in each of these groups, namely, automatic document indexing, document and term clustering, query definition, similarity function learning, image retrieval, user profile design, web page classification and Internet search agent design. Finally, Section 12 summarizes several concluding remarks.

2. Preliminaries

2.1. Information retrieval

IR may be defined, in general, as the problem of the selection of documentary information from storage in response to search questions provided by

a user [2]. Information retrieval systems (IRSs) deal with documentary bases containing textual, pictorial or vocal information and process user queries trying to allow the user to access to relevant information in an appropriate time interval. Nowadays, the different world wide web search engines such as Google constitute the main examples of IRSs.

The next three subsections respectively introduce the components of an IRS, the different existing retrieval models and some topics on IRS evaluation.

2.1.1. Components of an information retrieval system

An IRS is basically constituted by three main components (see Fig. 1); whose composition is introduced as follows [2,47].

(1) *The documentary data base.* This component stores the documents and the representation of their information contents. It is associated with the *indexer module*, which automatically generates a representation for each document by extracting the document contents. Textual document representation is typically based on index terms (that can be either single terms or sequences), which are the content identifiers of the documents.
(2) *The query subsystem.* It allows the users to formulate their queries and presents the relevant documents retrieved by the system to them. To do so, it includes a *query language*, that collects the rules to generate legitimate queries and procedures to select the relevant documents.
(3) *The matching mechanism.* It evaluates the degree to which the document representations satisfy the requirements expressed in the query, the *retrieval status value* (RSV), and retrieves those documents that are judged to be relevant to it.

2.1.2. Information retrieval models

Several retrieval models have been studied and developed in the IR area. Next, we analyze some of these models.

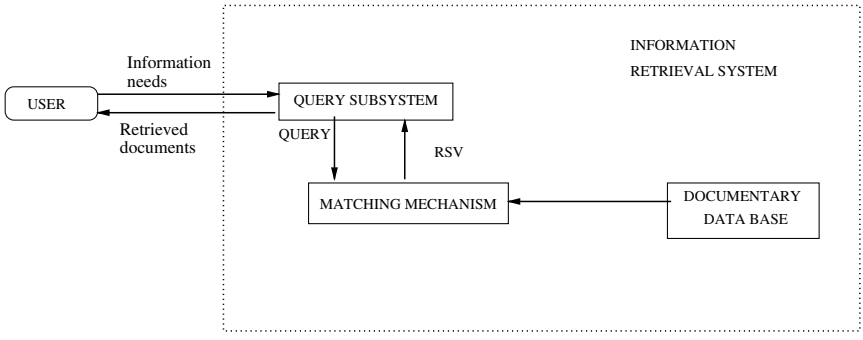

Fig. 1. Graphical representation of an information retrieval system.

Boolean model [53]. In the Boolean retrieval model, the indexer module performs a binary indexing in the sense that a term in a document representation is either significant (appears at least once in it) or not (it does not appear in it at all). User queries in this model are expressed using a query language that is based on these terms and allows combinations of simple user requirements with the logical operators *AND*, *OR* and *NOT* [47,53]. The result obtained from the processing of a query is a set of documents that totally match with it, i.e., only two possibilities are considered for each document: to be or not to be relevant for the user's needs, represented by the user query.

Vector space model [47]. In this model, a document is viewed as a vector in an n-dimensional document space, where n is the number of distinguishing terms used to describe contents of the documents in a collection, and each term represents one dimension in the document space. A query is also treated in the same way and constructed from the terms and weights provided in the user request. Document retrieval is based on the measurement of the similarity between the query and the documents. This means that documents with a higher similarity to the query are judged to be more relevant to it and should be retrieved by the IRS in a higher position in the list of retrieved documents. This way, the retrieved documents can be orderly presented to the user with respect to their relevance to the query.

Probabilistic model [4,26]. This model tries to use the probability theory to build the search function and its operation mode. The information used to compose the search function is obtained from the distribution of the index terms throughout the collection of documents or a subset of it. This information is used to set the values of some parameters of the search function, which is composed of a set of weights associated to the index terms.

Fuzzy model [5]. Let D be a set of documents and T be a set of unique and significant terms existing in them. An indexing function $F : D \times T \rightarrow [0, 1]$ is defined as a fuzzy relation mapping the degree to which document d belongs to the set of documents "about" the concept(s) represented by term t. Fuzzy queries are expressed using a query language that is based on weighted terms, where the numerical or linguistic weights represent the "subjective importance" of the selection requirements. In Fuzzy IRSs, the query subsystem affords a fuzzy set q defined on the document domain specifying the degree of relevance (the RSV) of each document in the data base with respect to the processed query. There are three different interpretations for the query term weights: importance, threshold and perfect document. When using the *importance* interpretation, the query weights represent the relative importance of each term in the query. In the threshold interpretation, the weights represent minimum values that have to be overcome by the document weights. Finally, in the perfect document interpretation, the term weights represent the preferred user description of an ideal document.

2.1.3. Evaluation of information retrieval systems

There are several ways to measure the quality of an IRS, such as the system efficiency and effectiveness, and several subjective aspects related to the user satisfaction (see, for example, [2, chapter 3]). Traditionally, the retrieval effectiveness—usually based on the document relevance with respect to the user's needs—is the most considered. There are different criteria to measure this aspect, with the *precision* and the *recall* being the most used.

Precision is the rate between the relevant documents retrieved by the IRS in response to a query and the total number of documents retrieved, whilst recall is the rate between the number of relevant documents retrieved and the total number of relevant documents to the query existing in the data base [53]. The mathematical expression of each of them is showed as follows:

$$P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d}; \qquad R = \frac{\sum_d r_d \cdot f_d}{\sum_d r_d}$$

with $r_d \in \{0, 1\}$ being the relevance of document d for the user and $f_d \in \{0, 1\}$ being the retrieval of document d in the processing of the current query. Notice that both measures are defined in $[0, 1]$, with 1 being the optimal value.

On the other hand, a usual measure in vector space IRSs is the averaged precision at a number of recall levels. An increasing set of p equidistant recall values is fixed, considering them as marks in the retrieved document set and then the precision is calculated in each level $(P1, P2, \ldots, Pj)$. The final measure value is the average of these values:

$$P = \frac{1}{j} \cdot \sum_{i=1}^{j} P_i$$

Typical choices for j are 3 (0.25, 0.5 and 0.75) and 11 (0, 0.1, 0.2, ..., 0.9, 1).

2.2. Evolutionary algorithms

EC [1] uses computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems. There is a variety of evolutionary computational models that have been proposed and studied, which are referred as EAs [1]. There have been four well-defined EAs which have served as the basis for much of the activity in the field: *genetic algorithms* (GAs) [41], *evolution strategies* [27], *genetic programming* (GP) [32] and *evolutionary programming* [24].

An EA maintains a population of trial solutions, imposes random changes to these solutions, and incorporates selection to determine which ones are going to be maintained in future generations and which will be removed from the pool of trials. There are some important differences between the existing EAs. GAs [41] emphasize models of genetic operators as observed in nature, such as crossover (recombination) and mutation, and apply these to abstracted

chromosomes with different representation schemes according to the problem being solved. Evolution strategies and evolutionary programming only apply to real-valued problems and emphasize mutational transformations that maintain the behavioural linkage between each parent and its offspring.

As regards GP [32], it constitutes a variant of GAs, based on evolving structures encoding programs such as expression trees. Apart from adapting the crossover and mutation operators to deal with the specific coding scheme considered, the remaining algorithm components remain the same.

3. Applications of evolutionary algorithms to information retrieval

There has been an increasing interest in the application of AI tools to IR in the last few years. Concretely, the machine learning paradigm [42], whose aim is the design of system able to automatically acquire knowledge by themselves, seems to be interesting in this topic [11].

EAs are not specifically learning algorithms but they offer a powerful and domain independent search ability that can be used in many learning tasks, since learning and self-organization can be considered as optimization problems in many cases. Due to this reason, the application of EAs to IR has increased in the last decade. Among others, EAs have been applied to solve the following IR problems:

(1) automatic document indexing,
(2) document and term clustering,
(3) query definition,
(4) matching function learning,
(5) image retrieval,
(6) design of user profiles for IR on the Internet,
(7) web page classification,
(8) design of agents for Internet searching.

Next sections show a review of the different proposals made in these areas in the last few years.

4. Automatic document indexing

The applications in this area are targeted to the learning—by means of adaptation—of the descriptions of the documents in the documentary base with the aim of facilitating document retrieval in the face of relevant queries.

4.1. Gordon's proposal

The first approach to document indexing by means of GAs was developed by Gordon. In [28], the author proposes to associate more than a single description to each document and to adapt them throughout time as a good solution to the problem of the different forms that different users' queries searching for the same documents can present. With this idea in mind, the system matches the query with all the document descriptions to analyze the relevance of a document to a particular query, and makes a decision based on the average of the partial matching.

Gordon proposes a GA to derive the document descriptions. He chooses a binary coding scheme where each description is a fixed length, binary vector. The genetic population is composed of different 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 and non-relevant) by means of the Jaccard's index, and then computing the average adaptation values of the description to the set of relevant and non-relevant queries.

The GA considered is quite unusual as there is no mutation operator and the crossover probability is equal to 1. With regards to the selection scheme, the number of copies of each chromosome in the new population is calculated dividing its adaptation value by the population average.

Gordon validates the system in a real-world environment. He uses a set of 18 documents, with 17 descriptions (provided by users) and two sets of relevant and non-relevant queries for each one. He obtains promising results: after 40 generations, the descriptions are a 19.09% more similar to relevant queries and a 24.81% less similar to irrelevant queries. However, the problem is that an independent run of the GA is needed to obtain the description for each document in the data base, thus reducing the practical application of the proposal in large documentary bases.

4.2. Vrajitoru's proposal

In [54], Vrajitoru presents a different approach for the same problem. On the one hand, she works with the vector space model, and thus document descriptions are real vectors in the $[0, 1]$ interval which represent the weights associated to each term. On the other hand, each document has associated just one description which leads to encode the whole collection in a single chromosome.

The main features of the proposed GA are:

(1) A new, specific crossover operator, called *dissociated crossover*.
(2) Two mechanisms to generate initial document descriptions considering: (a) the content of the different document sections and (b) the relevance of these documents in relation to a set of queries (*query learn*).

(3) Two ways of operation based on: (a) a priori knowing the relevance of the documents and (b) using relevance feedback (see Section 6).

The main problem of this model is that the fitness function only considers one query, and thus the document descriptions are adapted to match with this single query and not with a set of queries as in Gordon's model.

The experiments developed use two of the usual test collections, CACM and CISI (see [2]), and compare two variants of the proposed GA (using two different crossover operators: the classical two-point crossover and the dissociated crossover). The latter variant achieves the best performance and they both obtain better precision than initial descriptions.

4.3. Fan, Gordon and Pathak's proposal

Fan et al. [21] propose an algorithm for indexing function learning based on GP, whose aim is to obtain an indexing function for the key term weighting of a documentary collection to improve the IR process.

To do so, they consider a GP algorithm with the following characteristics:

(1) A term weighting formula is represented by a tree, where the leaf nodes are the terminals (term frequency, document frequency, etc.) and the inner nodes are arithmetical operators.
(2) The fitness function is the averaged precision at eleven recall levels (see Section 2.1.3).
(3) It only uses crossover operator, the classical GP crossover. The parents involved in the crossover are chosen by tournament selection.
(4) The algorithm is based on a generational scheme with elitism.
(5) The evaluation scheme matches the representation of each query against each document, to obtain its retrieval status value (RSV). The similarity between a query and a document is calculated by the product.

The algorithm is tested using one of the TREC collections (AP) [2] and the obtained results are very promising.

5. Clustering of documents and terms

In the next two subsections, two different approaches included in this group are described. The first of them looks for groups of terms appearing with

similar frequencies in the documents of a collection while the other deals with the problem of obtaining user-oriented document clusters.

5.1. Robertson and Willet's proposal

The main idea of [45] is to look for groups of terms appearing with similar frequencies in the documents of a collection. To do so, the authors consider a GA grouping the terms without maintaining their initial order. The main features of the GA are:

(1) *Representation scheme.* Two different coding schemes are considered: *separator method* and *division-assignment method.*
(2) *Initial population.* The generation of the first chromosome depends on the chosen coding; the rest of individuals are randomly generated.
(3) *Operators.* Each operator has an application probability associated and it is selected spinning the roulette. Different crossover and mutation operators are used:
Mutation operator: inversion, random sublist and position.
Crossover operator: order-based, position-based, one-point and two-point crossovers.
(4) *Fitness function.* There are two proposals: (a) a measure of the relative entropy and (b) Pratt's measure.

They run their proposals with five different data sets. The obtained results are good, but similar to those of another algorithm for the same problem which has a lower run time.

5.2. Gordon's proposal

In [29], Gordon designs a philosophy according to which it is possible to make a user-oriented clustering of documents using any classical clustering technique. The basic idea is that the system adapts document descriptions throughout time. In this way, documents being relevant to a query finally have similar descriptions and the system periodically clusters the new adapted descriptions. In order to carry out this, the author proposes to use his GA (see Section 4.1).

Gordon works with the data base used in [28] (Section 4.1) and considers the *relative density of the cluster* as goodness measure. The obtained results are satisfactory enough: after 40 generations, the relative density value decreased to 0.336 when the number of documents per cluster is 4, and to 0.455 when clusters have six documents.

6. Query definition

This is the most extended group of applications of EAs to IR. Every proposal in this group use EAs either like a relevance feedback technique or like an Inductive Query by Example (IQBE) algorithm.

The basis of relevance feedback lie in the fact that either users normally formulate queries composed of terms that do not match the terms used to index the relevant documents to their needs, or they do not provide the appropriate weights for the query terms. The operation mode involving modifying the previous query—adding and removing terms or changing the weights of the existing query terms—taking into account the relevance judgements of the documents retrieved by it, constitutes a good way to solve the latter two problems and to improve the precision, and especially the recall, of the previous query [53].

IQBE was proposed in [11] 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". This way, IQBE is a process for assisting the users in the query formulation process performed by machine learning methods. It works by taking a set of relevant (and optionally, non-relevant documents) provided by a user and applying an off-line learning process to automatically generate a query describing the user's needs.

We can distinguish between three subgroups depending on the query components under adaptation: terms [11], weights [30,37–39,44,48,57] or the whole query [6–8,15–19,23,33,51].

6.1. Term learning: Chen et al.'s proposal

In [11], Chen et al. use a GA as an IQBE technique to learn the query terms that better represent a relevant document set provided by an user. They consider an IRS based on the vector space model. The components of the GA are as follows:

(1) Each chromosome is a fixed size, binary vector where each position is associated to an existing term in the initial relevant document set.
(2) The used operators are one-point crossover, uniform mutation and roulette-wheel selection.
(3) The fitness function measures the degree of similarity between the system—suggested set of documents and the initial searcher—identified set of relevant documents; the Jaccard's score is used.

The authors compare their proposal with an other tree techniques: a classical relevance feedback technique, simulated annealing and the ID3 algorithm. They use a documentary base with 8000 documents indexed by means of the

key concepts provided by the data base, and propose a benchmark experiment and a user-evaluation experiment. In both cases, the GA achieves the best results.

6.2. Weight learning

6.2.1. Roberton and Willet's proposal

In [44], Robertson and Willet propose a GA to investigate an upperbound for relevance feedback techniques in vector space IRSs and compare its results with Robertson and Spark Jones's F4 [46] retrospective relevance weights technique. The main characteristics of the GA are as follows:

(1) The authors consider two different coding schemes: real and Gray [41]. The initial population is randomly generated.
(2) The steady-state evolution model is applied [41].
(3) Two crossover and two mutation operators are used, one-point and two-point crossover, and random and creep mutation, respectively.
(4) The fitness function measures the degree in which the weight vector maintains the optimal order of documents in the retrieval process.

The experiments involve seven well-known documentary test collections: *CRANFIELD, KEEN, HARDING, EVANS, LISA, SMART* and *UKCIS*. The authors use the Cranfield collection for an extensive series of initial trials, with the best combination of parameters then being used for the remaining collections. They run the best obtained query and measure the recall of the 10, 20 and 50 first retrieved documents to evaluate the method goodness.

The results are very good, overcoming the performance of the F4 method in most of the cases. Only with *HARDING*, the results of both methods are similar. To solve this problem, they propose to work with negative weights.

6.2.2. Yang and Korfaghe's proposal

Yang and Korfaghe propose a similar GA to that of Robertson and Willet's in [57]. They use a real coding, and 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.

They propose three different fitness functions, one where the relevant documents are known and other two where this set is not known. All of them are based on the precision and recall measures.

The test document collection used for the experiments is the Cranfield collection. Sixteen different population sizes, ranging from 10 to 40 in steps of

two, were used for the 225 queries. For each one of these queries, they measure the precision and recall levels, in each 0.1 size interval and compare them with the figures associated to the initial queries without weights.

The results are very satisfactory. However, while the average precision for some levels of recall increases after this process, the same behaviour is not noticed in the average recall for some levels of precision.

6.2.3. Sanchez, Miyano and Brachet's proposal

In [48], Sanchez et al. propose a GA to learn the term weights of extended Boolean queries for fuzzy IRS in a relevance feedback process.

Binary-coded chromosomes encode the n term weights as well as the similarity threshold considered in the document retrieval (stored in the $n+1$th gene). The genetic operators are the classic ones and the fitness function is based on a linear combination of precision and recall.

The behaviour of the system is studied on a 479 document collection about patents. Unfortunately, they do not show the obtained results in the paper.

6.2.4. Horng and Yeh's proposal

In [30], the authors use a GA to adapt the query term weights in order to get the closest query vector to the optimal one. Firstly, they combine the Bigram's model and a Pat-tree approach to retrieve Chinese documents. The features of the system are:

(1) It is based on the vector space model, and both documents and queries are represented as real number vectors. The initial population is randomly generated.
(2) The used objective function is the non-interpolated average precision [9,34] which takes into account the order of appearance of the documents as well as the number of relevant and non-relevant documents retrieved.
(3) The GA uses two kinds of crossovers (weight-selection and natural crossover) and the mutation operator is defined as the inversion of a weight.
(4) The system applies a local search to find better solutions.

The experiments use Chinese documents extracted from the web (http://vita.fju.edu.tw and http://tw.yahoo.com/headlines). In both cases, the collections are divided in training and test data. Each experiment considers 45 different queries to retrieve documents.

The conventional measures, recall and precision, are used in the evaluation. They are combined in two different forms. The authors compare their proposal with Yang and Korfhage's GA [57], and the new genetic approach outperforms it.

6.2.5. López-Pujalte, Guerrero and Moya's proposals

In [37–39], López-Pujalte et al. evaluate the efficacy of a GA with different fitness functions for relevance feedback in the vector space model and compare the results with the classic Ide dec-hi method. The common characteristics of the GA used in the three works are:

(1) The chromosomes have the same number of genes as there are terms with non-zero weights in the query and in the documents of the feedback.
(2) The initial population is generated taking into account the available information. The original query is first run on the IRS and the first 15 documents retrieved are evaluated. Then, the initial population is built from the vectors of:
    (a) the initial query,
    (b) the relevant documents,
    (c) the non-relevant documents,
    (d) the non-relevant documents with the weights negated.
    This way, the population size will depend on the retrieval efficacy of the original query.
(3) The GA uses simple random sampling as selection mechanism.
(4) The authors use the simple one-point crossover and random mutation.
(5) The GA returns as the solution both the best chromosome (query) found during the process and the centroid of all the best chromosomes found in the final population.

Hence, the only difference between the different GAs run in the three papers is the fitness function. In [37,38], the authors run the GA with different, previously proposed fitness functions in the vector space model. From these two studies, they determine that the best performance is obtained with those taking into account the document ranking. Hence, in [39], they run the GA with three different order-based fitness functions, a function given in [30] and two new proposals.

To perform their experiments, they generate a experimental testbed from the Cranfield collection [37,39] or from the Cranfield, CISI, Medline and NPL collections [38]. Around 30 queries are selected from each collection for the evaluation, those having at least three relevant documents retrieved among the first 15, and at least five relevant documents yet to be retrieved.

The retrieval efficacy is evaluated by means of the averaged precision at eleven and three recall values (see Section 2.1.3). They also use the *residual collection* method [10], based on not considering previously seen documents to measure the IRS efficacy. The results are very good; all the proposed GAs improve the obtained results with the classical Ide dec-hi method and those GAs with order-based fitness functions get better results than the remainder.

6.3. Query learning

6.3.1. Smith and Smith's proposal

Smith and Smith propose a GP-based algorithm for learning queries for Boolean IRSs in [51]. Although they introduce it as a relevance feedback algorithm, the experimentation developed in the paper is actually closer to the IQBE framework. The algorithm components are described next:

(1) The Boolean queries are encoded in expression trees, whose terminal nodes are query terms and whose inner nodes are the Boolean operators *AND*, *OR* and *NOT*.
(2) Each generation is based on selecting two parents, with the best fitted having a larger chance to be chosen, and generating two offspring from them. Both offspring are added to the current population which increments its size in this way.
(3) The usual GP crossover is considered [32]. No mutation operator is applied.
(4) The initial population is generated by randomly selecting the terms included in the set of relevant documents provided by the user, having those present in more documents a higher probability of being selected.
(5) The fitness function gives a composite retrieval evaluation encompassing the two main retrieval parameters (precision and recall).

To perform the experiments, the authors use the Cranfield collection. In practice, the algorithm only was able to generate perfect queries when the initial collection of relevant documents is small.

6.3.2. Cordón, Herrera-Viedma and Luque's proposal

In [15], Cordón et al. propose an extension of Smith and Smith's algorithm, based on incorporating the multiobjective approach to the latter.

The differences between both proposals are:

(1) Cordón et al. consider a classical generational scheme where the selection probabilities are assigned by the proportional scheme instead of a steady-state evolution model.
(2) Cordón et al. use mutation operators which change a randomly selected term or operator by a random one, or a randomly selected subtree by a randomly generated one.
(3) The Pareto-based multiobjective EA incorporated to the basic Smith and Smith's GP algorithm is Fonseca and Fleming's MOGA [25]. In addition, a double niching scheme is applied in the genotypic and the phenotypic spaces to better cover the Pareto front deriving a large number of queries with a different composition and a different precision-recall tradeoff.

The performance of the new technique is very significant since it overcomes the basic Smith and Smith's algorithm in all cases as the results of the latter when considering typical values for the weighted combination are dominated by the solutions in the Pareto front of the former. Furthermore, the Pareto fronts obtained are very well distributed, including a high number of solutions in them.

6.3.3. Fernández-Villacañas and Shackleton's proposal

In [23], Fernández-Villacañas and Shackleton propose two evolutionary IQBE techniques for Boolean query learning and compare their behaviour. The first algorithm (BTGP, British Telecom Genetic Programming) is very similar to Smith and Smith's GP proposal. It is composed of a roulette-wheel selection, an usual GP crossover and a mutation operator based on swapping subtrees. Two fitness functions are investigated, a classic linear combination of precision and recall, and a new variant (coming from the field of classification systems design in machine learning) where both the number of missed positives (non-retrieved relevant documents) and the number of false positives (retrieved non-relevant documents) are minimized.

The other algorithm, MGA (Mapping Genetic Algorithm) is a classic binary GA which adapts query trees. The coding scheme allows to represent expression trees as binary strings; each node is encoded in a binary substring and a chromosome is obtained linking together node representations. It is composed of roulette-wheel selection, one-point crossover and random mutation, but the authors do not use the crossover operator in the experimentation developed as they mention that it causes premature convergence.

To perform the experiments, the authors consider two documentary bases which are divided into training and test document sets. In the first base, the results obtained are very bad; the first function generates queries that do not appropriately summarize the information needs of the set of documents and the use of the second function produces an overlearning when the generated queries are run against the test document set. In the second base, good results are obtained, in training and test, with MGA improving to BTGP in both cases.

6.3.4. Kraft, Petry, Buckes and Sadasivan's proposal

In [33], Kraft et al. propose an IQBE technique to learn the whole composition of extended Boolean queries for Fuzzy IRSs. The algorithm is based on GP and its components are described next:

(1) The fuzzy queries are encoded in expression trees, whose terminal nodes are query terms with their respective weights and whose inner nodes are the Boolean operators.
(2) A classical generational scheme is applied.

(3) The usual GP crossover is considered and three possibilities are randomly selected for the GP mutation: change of an operator, negation of a term or change of negative term into a positive term.
(4) Two different fitness functions are considered based on the classical precision and recall measures. While one of them only considers the recall value, the other also takes the precision into account.

For the experiments, the authors use a collection consisting of 483 abstracts taken from consecutive issues of the ACM. The preliminary results indicate that randomly selecting terms from the set of all terms to populate queries does not work efficiently; to solve this drawback the terms are selected from the predetermined documents specified as relevant.

6.3.5. Cordón, Moya and Zarco's proposals

Although Kraft et al.'s algorithm obtains good results, it suffers from one of the main limitations of the GP paradigm: the learning of the weights considered in the encoded structure can only be performed by mutation. Hence, it is very difficult for the algorithm to obtain the term weights, which constitutes an important drawback due to their importance in the query. In order to solve this drawback, *Cordón et al.* propose some new approaches aiming at improving Kraft et al.'s algorithm performance.

In the first proposal, they make use of the GA-P paradigm [17] based on combining traditional GAs with the GP technique. While the GP part evolves the expressions, the GA part concurrently evolves the coefficients used in them. The expressional part (GP part) encodes the query composition—terms and logical operators—and the coefficient string (GA part) represents the term weights.

This proposal uses a selection based on the steady-state approach and induces niches in the GA-P population [49]. Two different kinds of crossover are considered, *intra-niche crossover* and *inter-niche crossover*, depending whether the parents to be crossed are encoding the same query tree or not. The GA part is crossed using the BLX-α crossover and Michalewicz's non-uniform operator is considered to perform mutation. The classical GP operators are applied in the GP part.

On the other hand, in [18], a hybrid simulated annealing-genetic programming EA is considered to extend Kraft et al.'s proposal. The coding scheme is the same that in the previous proposal. The neighbourhood operator *macro-mutation* generates a neighbour query to the current one by changing the expressional part or the value string.

In both proposals, an extension to adapt the retrieval threshold (as done in [48], Section 6.2.3) is proposed. In every case, the obtained results are significantly good, outperforming Kraft et al.'s proposal to a large degree in the test collections considered (Cranfield among them).

In [16,19], a new IQBE process is developed to being able to simultaneously generate several extended Boolean queries with a different precision-recall tradeoff in a single run. It is based on an adaptation of the GA-P algorithm, specially designed to tackle with multiobjective problems by means of a Pareto-based multiobjective technique.

The performance of the algorithm is tested on the usual Cranfield collection, but while [19] uses the classical experimental setup, a training-test operation model is used in [16]. Training and test sets are used to validate if the derived queries are able to retrieve new relevant documents when are applied to a different documentary collection. The results are very promising in both cases.

6.3.6. Boughanem, Chrisment and Tamine's proposals

All Boughanem et al.'s proposals [6–8] are based on the use of genetic techniques for solving multimodal problems. The main characteristics of these models are the use of knowledge-based genetic operators, instead of the usual blind operators, and of niching techniques.

In [6], they define a specific GA for IR based on knowledge-based operators and guided by a heuristic for relevance multimodal problem solving. They measure the effects of P_c and P_m probabilities and the population size, and then they the knowledge-based operators versus blind operators. The goal is to find an optimal set of documents which best match the user's needs.

In [7,8], a genetic approach combining the results from multiple query evaluations is copared. The GA's components are described next:

(1) The genetic individual is a query formulation and each gene is an index term o concept. The population is organized into several niches.
(2) The fitness function measures the effectiveness of a query during the retrieval stage.
(3) The selection scheme is based on a variant of the usual roulette-wheel selection. The crossover operator is based on term-weight and the mutation operator explores the terms occurring in the relevant documents to expand and/or reweight the query.

Furthermore, the GA in [6] uses blind operators and term co-occurrence based crossover; while those in [7,8] use merging methods.

Several experiments of different kind are carried out on TREC collections to validate the proposed approaches. In the three works, the presented results demonstrate the effectiveness of the genetic approaches in performing multiple query evaluation and the interesting use of knowledge domain to develop genetic operators and of the niching methods to improve the retrieval effectiveness.

7. Matching function learning

The aim is to use an EA to generate a similarity measure for a vector space IRS to improve its retrieval efficacy for an specific user. This constitutes a new relevance feedback philosophy since matching functions are adapted instead of queries.

Two different variants have been proposed in the specialized literature:

(1) *Linear combination of existing similarity functions.* In [43], Pathak et al. propose a new weighted matching function, which is the linear combination of different existing similarity functions. The weighting parameters are estimated by a GA based on relevance feedback from users. They use real coding, a classical generational scheme, two-point crossover and Gaussian noise mutation. The algorithm is tested on the Cranfield collection and the results look very encouraging.
(2) *Automatic similarity measure learning.* A GP algorithm to automatically learn a matching function with relevance feedback is introduced in [20,22]. The similarity functions are represented as trees, and a classical generational scheme and the usual GP crossover are considered. However, no experiments are showed.

8. Image retrieval

There are different proposals combining EAs and image retrieval. Some of them are summarized below.

Cho and Lee [14] develop an image retrieval system based on human preferences and emotions by using an interactive genetic algorithm (IGA) with the purpose of supplementing the lack of the user's expression capability. The system extracts the features from images by wavelet transform, and uses the IGA to search the image that the user has in mind adopting the user's choice as fitness when the fitness function cannot be explicitly defined.

In [31], Kato and Iisaku describe an image retrieval system also based on a GA with an interactive mechanism that dynamically reflects the subjectivity of the individual user in the retrieval results and propose a novel method for enhancing circumstantial dependence.

In [52], a new method for computing image similarity is proposed, called local similarity pattern. It is based on the idea that distinguishing different objects in the image requires different similarity criteria for each object. In addition, a GA-based method is proposed for finding an optimal assignment of similarity criteria to image regions, and incorporated in the relevance feedback mechanism.

9. Design of user profiles for information retrieval on the Internet

IRSs are limited by the lack of personalization in the representation of the user's needs. An important issue in this situation is the construction of user profiles which maintain previously retrieved information associated with previous user's needs. Next paragraphs show three proposals involving on user profiles and GAs.

In [40], an agent is proposed to model the user's information needs for searches in the web by an adaptative process based on a GA with fuzzy genes. The GA keeps the knowledge about the user's preferences and gets the feedback from the user. Fuzzy set theory is considered to handle the imprecision of the user's preferences and the user's evaluation of the retrieved documents. It constitutes a viable approach to a system that can learn the information needs of the user, and keep up with the evolution of these needs, utilizing only relevance feedback from the user.

In [35], Larsen et al. present a scheme for maintaining experience-based knowledge on user preferences in a user profile. The user profile is generated from a filtering process instead of a retrieval process where no information about the user is considered. In this way, the authors filter the document collection using the information retrieved in the first query. In order to generate the profile, the proposed system uses a GA to find the most discriminatory terms, i.e., those allowing the system to discern between relevant and non-relevant documents, which are selected and stored as a part of the user's profile to be used in future queries to the system.

In [13], Chen and Shahabi propose an adaptative soft query (ASQ) system to improve the accuracy of user profiles. ASQ consists of two main components: an on-line soft query system and an off-line GA-based learning mechanism. The former system provides personalized query results by integrating image information and user profiles with a fuzzy-logic based aggregation technique. The GA is then used to improve the user profiles through user's feedback. The experimental results indicate that the retrieval accuracy is significantly increased.

10. Web page classification

Catalogues play an important role in most of the current web search engines. In [36], Loia and Luengo present an evolutionary approach useful to automatically construct a catalogue as well as to perform the classification of web documents. The proposal faces the two fundamental problems of web clustering: the high dimensionality of the feature space and the knowledge of the entire document. The first problem is tackled with genetic computation

while the authors perform a clustering based on the analysis of context in order to face the second one.

The genome is defined as a tree-based structure and two different evaluation functions are used (*clustering fitness* and *quality of distribution*). As genetic operators, the one-point crossover and five different mutation operators (Cutting, Merging, Specialization Grade, Exchange Parent and Change Parent) are defined.

The system is verified taking the Open Directory Project (http://dmoz.org) as target and the results show that it allowed the authors to classify a web document with a precision comparable to a directory approach and with a dimensionality and updating speed comparable to those of the existing indexing techniques.

11. Design of agents for Internet searching

While the Internet services are popular and appealing to many on-line users, difficulties with search are expected to worsen as the amount of on-line information increases. Therefore, it is necessary to improve the existing search agents. Next paragraphs show different proposals that uses EAs in Internet search with this aim.

In [3], Bergström et al. present a method to find textual relations in electronic documents using GP and semantic networks. The system aims at enhancing IR by automatically extracting relations from text. This technique could be an important countermeasure against information overload and can also be used to enable feasible user interfaces on small screens. The GP uses tournament selection and the classical GP crossover. The experiments are performed on pre-processed texts from the web and the initial results confirm the feasibility of the method.

In [12], an intelligent personal spider approach for Internet searching is proposed. Chen et al. implement 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 and search for the most closely related homepages in the web, based on the existing links and keyword indexing. In benchmarking experiments, both algorithms are compared. The GA spider does not outperform the best first search spider, but both results are comparable and complementary.

In [50], Shih proposes a theoretical computation model for mobile agent evolution on the Internet which is based on "food web", the law of natural balancing. As working model, Shih defines a logical network for agent connections/communications, called *Agent Communication Network*. In order to simulate agent evolution, he develops a set of algorithms for the distributed

computing of agent programs. Finally, a simulation environment based on JATLite is designed to support the proposed theory.

In [55,56], different search strategies behind the current search engines for the world wide web are studied to determine the applicability of a GP model for the diversity set of web documents. Walker uses GP to search in the web. In addition, in [56] he also uses a parallel implementation. In both works, he studies the refinement of the database indexers over a period of 8 days. The results show that the databases for the distinct search engines stabilized after the completion of the initial search, that is, the user can optimize the initial results by repeating the search over a period of time.

12. Concluding remarks

In this paper, we have reviewed the different applications of EC to IR, analyzing the different kinds of IR problems that have been solved by EAs. EAs has been applied, among other problems, to: *automatic document indexing*, *document and term clustering*, *query definition*, *matching function learning*, *image retrieval*, *design of user profiles for IR on the Internet*, *web page classification* and *design of agents for Internet searching*. In several cases, the obtained results are very promising. Hence, EAs can be useful in the future to: (1) information fusion and text extraction, (2) multimedia retrieval and (3) ranking and web mining.

References

[1] T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing and Oxford University Press, 1997.
[2] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Adisson, 1999.
[3] A. Bergström, P. Jaksetic, P. Nordin, Enhancing information retrieval by automatic acquisition of textual relations using genetic programming, in: Proc. 2000 International Conference on Intelligent User Interfaces, New Orleans, USA, 2000, pp. 29–32.
[4] A. Bookstein, Outline of a general probabilistic retrieval model, Journal of Documentation 39 (2) (1983) 63–72.
[5] G. Bordogna, P. Carrara, G. Pasi, Fuzzy approaches to extend Boolean information retrieval, in: P. Bosc, J. Kacprzyk (Eds.), Fuzziness in Database Management Systems, 1995, pp. 231–274.
[6] M. Boughanem, C. Chrisment, L. Tamine, Genetic approach to query space exploration, Information Retrieval 1 (1999) 175–192.
[7] M. Boughanem, C. Chrisment, L. Tamine, On using genetic algorithms for multimodal relevance optimization in information retrieval, Journal of the American Society for Information Science and Technology 53 (11) (2002) 934–942.
[8] M. Boughanem, C. Chrisment, L. Tamine, Multiple query evaluation based on an enhanced genetic algorithm, Information Processing and Management 39 (2003) 215–231.

[9] C.H. Chang, C.C. Hsun, The Design of an Information System for Hypertext Retrieval and Automatic Discovery on WWW, PhD thesis, Department of CSIE, National Taiwan University 1999.
[10] Y.K. Chang, C. Cirillo, Evaluation of feedback retrieval using modified freezing, residual collection, and test and control groups, in: G. Salton (Ed.), The Smart Retrieval System—Experiments in Automatic Document Processing, Prentice Hall, 1971, pp. 335–370.
[11] H. Chen et al., A machine learning approach to inductive query by examples: an experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing, Journal of the American Society for Information Science 49 (8) (1998) 693–705.
[12] H. Chen, C. Yi-Ming, M. Ramsey, C. Yang, An intelligent personal spider (agent) for dynamic Internet/Intranet searching, Decision Support Systems 23 (1998) 41–58.
[13] Y. Chen, C. Shahabi, Automatically improving the accuracy of user profile with genetic algorithm, in: Proc. International Conference on Artificial Intelligence and Soft Computing, Cancun, Mexico, 2001.
[14] S. Cho, J. Lee, A human-oriented image retrieval system using interactive genetic algorithm, IEEE Transactions on System, Man and Cybernetics. Part A: Systems and Humans 32 (3) (2002) 452–458.
[15] O. Cordón, E. Herrera-Viedma, M. Luque, Evolutionary learning of Boolean queries by multiobjective genetic programming, in: Proc. PPSN-VII, Granada, Spain, 2002, pp. 710–719, LNCS 2439.
[16] O. Cordón, E. Herrera-Viedma, M. Luque, F. Moya, C. Zarco, Analyzing the performance of a multiobjective GA-P algorithm for learning fuzzy queries in a machine learning environment, in: International Fuzzy Systems Association World Congress, 2003, Istanbul, Turkey, LNAI 2715.
[17] O. Cordón, F. Moya, C. Zarco, A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system, Mathware & Soft Computing 7 (2–3) (2000) 309–322.
[18] O. Cordón, F. Moya, C. Zarco, A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems, Soft Computing 6 (5) (2002) 308–319.
[19] O. Cordón, F. Moya, C. Zarco, Automatic learning of multiple extended Boolean queries by multiobjective GA-P algorithms, in: V. Loia, M. Nikravesh, L.A. Zadeh (Eds.), Fuzzy Logic and the Internet, Springer, 2003.
[20] W. Fan, M. Gordon, P. Pathak, Automatic generation of a matching function by genetic programming for effective information retrieval, in: America's Conference on Information System, Milwaukee, USA, August 1999.
[21] W. Fan, M.D. Gordon, P. Pathak, Personalization of search engine services for effective retrieval and knowledge management, in: Proc. 2000 International Conference on Information Systems (ICIS), Brisbane, Australia, 2000.
[22] W. Fan, M.D. Gordon, P. Pathak, Discovery of context-specific ranking functions for effective information retrieval using genetic programming, IEEE Transactions on knowledge and Data Engineering, in press.
[23] J.L. Fernández-Villacañas, M. Shackleton, Investigation of the importance of the genotype–phenotype mapping in information retrieval, Future Generation Computer Systems 19 (2003) 55–68.
[24] D.B. Fogel, System Identification trough Simulated Evolution: A Machine Learning Approach, Ginn Press, USA, 1991.
[25] C.M. Fonseca, P.J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: Proc. Fifth International Conference on Genetic Algorithms, 1993, pp. 416–423.

[26] N. Fuhr, Probabilistic models in information retrieval, Computer Journal 35 (3) (1992) 243–255.
[27] H.-P. GenerSchwefel, Evolution and Optimum Seeking, in: Sixth Generation Computer Technology Series, John Wiley and Sons, 1995.
[28] M. Gordon, Probabilistic and genetic algorithms for document retrieval, Communications of the ACM 31 (10) (1988) 1208–1218.
[29] M. Gordon, User-based document clustering by redescribing subject description with a genetic algorithm, Journal of the American Society for Information Science 42 (5) (1991) 311–322.
[30] J. Horng, C. Yeh, Applying genetic algorithms to query optimization in document retrieval, Information Processing and Management 36 (2000) 737–759.
[31] S. Kato, S. Iisaku, An image retrieval method based on a genetic algorithm, in: Proc. Twelfth International Conference on Information Networking (ICOIN-12), 1998, pp. 333–336.
[32] J. Koza, Genetic Programming. On the Programming of Computers by means of Natural Selection, The MIT Press, 1992.
[33] D.H. Kraft, F.E. Petry, B.P. Buckes, T. Sadasivan, Genetic algorithm for query optimization in information retrieval: relevance feedback, in: E. Sanchez, T. Shibata, L.A. Zadeh (Eds.), Genetic Algorithms and Fuzzy Logic Systems, 1997, pp. 155–173.
[34] K.L. Kwok, Comparing representations in Chinese information retrieval, in: ACM SIGIR'97, Philadelphia, USA, 1997, pp. 34–41.
[35] H. Larsen, N. Marín, M.J. Martín-Bautista, M.A. Vila, Using genetic feature selection for optimizing user profile, Mathware & Soft Computing 7 (2000) 275–286.
[36] V. Loia, P. Luengo, An evolutionary approach to automatic web page categorization and updating, in: First Asia–Pacific Conference, Maebashi City, Japan, 2001, pp. 292–302.
[37] C. López-Pujalte, V. Guerrero, F. Moya, A test of genetic algorithms in relevance feedback, Information Processing & Management 38 (2002) 793–805.
[38] C. López-Pujalte, V. Guerrero, F. Moya, Genetic algorithms in relevance feedback: a second test and new contributions, Information Processing & Management 39 (5) (2003) 669–807.
[39] C. López-Pujalte, V. Guerrero, F. Moya, Order-based fitness functions for genetic algorithms applied to relevance feedback, Journal of the American Society for Information Science and Technology 54 (2) (2003) 152–160.
[40] M.J. Martin-Bautista, H. Larsen, M.A. Vila, A fuzzy genetic algorithm approach to an adaptive information retrieval agent, Journal of the American Society for Information Science 50 (9) (1999) 760–771.
[41] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1996.
[42] T. Mitchell, Machine Learning, McGraw-Hill, 1997.
[43] P. Pathak, M. Gordon, W. Fan, Effective information retrieval using genetic algorithms based matching functions adaption, in: Proc. 33rd Hawaii International Conference on Science (HICS), Hawaii, USA, 2000.
[44] A. Robertson, P. Willet, An upperbound to the performance for ranked-output searching: optimal weighting of query terms using a genetic algorithm, Journal of Documentation 52 (4) (1996) 405–420.
[45] A.M. Robertson, P. Willet, Generation of equifrequent groups of words using a genetic algorithm, Journal of Documentation 50 (3) (1994) 213–232.
[46] S.E. Robertson, K. Spark Jones, Relevance weighting of search terms, Journal of the American Society for Information Science 27 (1976) 129–145.
[47] G. Salton, M.H. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.
[48] E. Sanchez, H. Miyano, J. Brachet, Optimization of fuzzy queries with genetic algorithms. Applications to a data base of patents in biomedical engineering, in: Proc. VI IFSA Congress, Sao-Paulo, Brazil, 1995, pp. 293–296.

[49] L. Sánchez, A niching scheme for steady state GA-P and its application to fuzzy rule based classifiers induction, Mathware & Soft Computing 7 (2–3) (2000) 337–350.
[50] T.K. Shih, Mobile agent evolution computing, Information Sciences 137 (2001) 53–73.
[51] M.P. Smith, M. Smith, The use of genetic programming to build Boolean queries for text retrieval through relevance feedback, Journal of Information Science 23 (6) (1997) 423–431.
[52] Z. Stejic, Y. Takama, K. Hirota, Genetic algorithm-based relevance feedback for image retrieval using local similarity patterns, Information Processing & Management 39 (2003) 1–23.
[53] C.J. Van Rijsbergen, Information Retrieval, second ed., Butterworth, 1979.
[54] D. Vrajitoru, Crossover improvement for the genetic algorithm in information retrieval, Information Processing and Management 34 (4) (1998) 405–415.
[55] R.L. Walker, Assessment of the web using genetic programming, in: Proc. Genetic and Evolutionary Computation Conference, San Francisco, 1999, pp. 1750–1755.
[56] R.L. Walker, Search engine case study: searching the web using genetic programming and MPI, Parallel Computing 27 (2001) 71–89.
[57] J. Yang, R. Korfhage, Query modifications using genetic algorithms in vector space models, International Journal of Expert Systems 7 (2) (1994) 165–191.