Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 42 (2006) 615–632
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Improving the learning of Boolean queries by means of a multiobjective IQBE evolutionary algorithm

O. Cordón [a], E. Herrera-Viedma [a,*], M. Luque [b]

[a] *Department of Computer Science and A.I., University of Granada, Granada, Spain*
[b] *Department of Computer Science and N.A., University of Córdoba, Córdoba, Spain*

Received 17 December 2003; accepted 23 February 2005

Abstract

The Inductive Query By Example (IQBE) paradigm allows a system to automatically derive queries for a specific Information Retrieval System (IRS). Classic IRSs based on this paradigm [Smith, M., & Smith, M. (1997). The use of genetic programming to build Boolean queries for text retrieval through relevance feedback. *Journal of Information Science, 23*(6), 423–431] generate a single solution (Boolean query) in each run, that with the best fitness value, which is usually based on a weighted combination of the basic performance criteria, precision and recall.

A desirable aspect of IRSs, especially of those based on the IQBE paradigm, is to be able to get more than one query for the same information needs, with high precision arid recall values or with different trade-offs between both.

In this contribution, a new IQBE process is proposed combining a previous basic algorithm to automatically derive Boolean queries for Boolean IRSs [Smith, M., & Smith, M. (1997). The use of genetic programming to build Boolean queries for text retrieval through relevance feedback. *Journal of Information Science, 23*(6), 423–431] and an advanced evolutionary multiobjective approach [Coello, C. A., Van Veldhuizen, D. A., & Lamant, G. B. (2002). *Evolutionary algorithms for solving multiobjective problems*. Kluwer Academic Publishers], which obtains several queries with a different precision–recall trade-off in a single run. The performance of the new proposal will be tested on the Cranfield and CACM collections and compared to the well-known Smith and Smith's algorithm, showing how it improves the learning of queries and thus it could better assist the user in the query formulation process.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Boolean information retrieval systems; Genetic programming; Inductive query by example; Multiobjective evolutionary algorithms; Query learning

* Corresponding author. Tel.: +34 958 244258; fax: +34 958 243317.
E-mail addresses: ocordon@decsai.ugr.es (O. Cordón), viedma@decsai.ugr.es (E. Herrera-Viedma).

0306-4573/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2005.02.006

1. Introduction

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 (Baeza-Yates & Ribeiro-Neto, 1999; Salton & McGill, 1983). Information retrieval systems (IRSs) are a kind of information systems that deal with data bases composed of information items—documents that may consist of 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 development of the *WWW* has increased the interest on the study of IRSs.

Many IRSs still consider the Boolean IR model (Van Rijsbergen, 1979), based on the use of Boolean queries where the query terms are joined by the logical operators AND and OR. This way, the user needs to have a clear knowledge on how to connect the query terms together using the Boolean operators in order to build a query defining his information needs. The difficulty found by nonexpert users to formulate these kinds of queries sometimes makes necessary the design of automatic methods for this task. The paradigm of Inductive Query by Example (IQBE) (Chen, Shankaranarayanan, She, & Iyer, 1998), where a query describing the information contents of a set of documents provided by a user is automatically derived, can be useful to assist the user in the query formulation process. Focusing on the Boolean IR model, the most known existing approach is that of Smith and Smith (1997), which is based on a kind of evolutionary algorithm (EA) (Bäck, Fogel, & Michalewicz, 1997), genetic programming (GP) (Koza, 1992). As usual in the topic (Cordón, Herrera-Viedma, López-Pujalte, Luque, & Zarco, 2003), this approach is guided by a weighted fitness function combining two retrieval accuracy criteria, precision and recall. The main characteristic of this approach is that it provides a single query in each run.

Given the retrieval performance of an IRS is usually measured in terms of these two criteria, precision and recall (Van Rijsbergen, 1979), the optimization of any of its components, and concretely the automatic learning of Boolean queries, is thus a clear example of a multiobjective problem. EAs have been commonly used for IQBE purposes and their application in the area has been usually based on combining both criteria in a single scalar fitness function by means of a weighting scheme (Cordón, Herrera-Viedma, López-Pujalte, et al., 2003). However, there is a kind of EA specially designed for multiobjective problems, *multiobjective evolutionary algorithms*, which are able to obtain different nondominated solutions to the problem in a single run (Coello, Van Veldhuizen, & Lamant, 2002; Deb, 2001). In IR, specifically in the IQBE paradigm, they would allow us to derive a number of queries with a different precision–recall trade-off in a single run of the IQBE algorithm, and in such a way to improve the aid possibilities to the users in the formulation of their queries.

In this paper, we present a new evolutionary tool to learn Boolean queries that improves the Smith and Smith's (1997) approach, called multiobjective IQBE EA. We define it by extending the Smith and Smith's approach incorporating Pareto-based evolutionary multiobjective components into GP. To do so, we consider one of the most known and well performing Pareto-based multiobjective EAs, SPEA (Zitzler & Thiele, 1999). The main feature of this EA is the maintenance of the elitism concept in a multiobjective evolutionary algorithm. This improves the performance of our multiobjective GP algorithm. In order to represent a real-world text retrieval IQBE environment where a user provides a relatively small number of relevant and irrelevant documents, the experimental testbed will be based on two of the most known small size IR benchmarks, the Cranfield and CACM document collections (Baeza-Yates & Ribeiro-Neto, 1999; Salton & McGill, 1983). With our proposal we improve and increase the user assistance possibilities in the formulation of queries by means of evolutionary computation tools.

With this aim, this contribution is structured as follows. Section 2 is devoted to introduce the preliminaries, including the basis of Boolean IRSs, the definition of both precision and recall criteria, the main aspects of IQBE techniques, a review on EAs and on their application to IR, tasks, arid finally, the main aspects of multiobjective EAs. Section 3 is devoted to introduce the main aspects of the Smith and Smith's proposal and to extend the latter algorithm to deal with the multiobjective problem of simultaneously optimizing both precision and recall by means of the SPEA Pareto-based approach while the experiments

developed to test the new proposal and the results obtained are shown in Sections 4 and 5, respectively. Finally, several concluding remarks are pointed out in Section 6.

2. Preliminaries

2.1. Boolean IRS

An IRS is basically constituted by three main components, as shown in Fig. 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.

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). Let D be a set of documents and T be a set of unique and significant terms existing in them. The indexer module of the Boolean IRS defines an indexing function: $F : D \times T \rightarrow \{0,1\}$, where $F(d,t)$ takes value 1 if term t appears in document d and 0 otherwise.

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.

Boolean queries are expressed using a query language that is based on query terms and permits combinations of simple user requirements with logical operators AND, OR and NOT (Van Rijsbergen, 1979). 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 his query.

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.

As said, the RSV has only two values associated, 0 and 1, in Boolean IRSs. In order to match a query, a document has to fulfill it completely, i.e., it has to include the positive query terms specified in the search expression and not to include those that have been specifically given in a negative way. In order to obtain the set of relevant documents for a query, it is represented as a parse tree and is evaluated from the leaves to the root. Each leaf is associated to the set of documents including (or not including) the corresponding (negative) query term. Then, the retrieved document sets in the inner nodes are computed by applying set arithmetic (with the AND operator being the set intersection and the OR operator standing for the set union). The final set of retrieved documents is that associated to the root when finishing the evaluation of the tree.

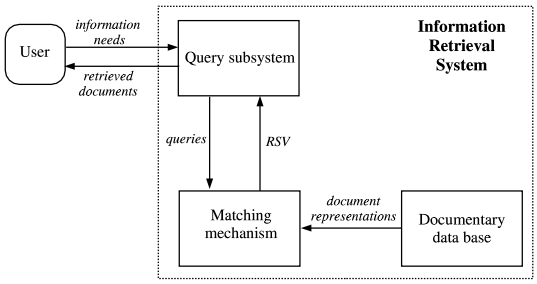

Fig. 1. Generic structure of an IRS.

2.2. Evaluation criteria of IRSs

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, Baeza-Yates & Ribeiro-Neto, 1999, 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* arid 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 relevant documents retrieved and the total number of relevant documents to the query existing in the data base (Van Rijsbergen, 1979). The mathematical expression of each of them is shown as follows:

$$P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d}; \quad R = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d} \tag{1}$$

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.

We should also notice that the only way to know all the relevant documents for a query existing in a documentary base (needed to compute the recall measure) is to evaluate all of them one by one. Due to this and to the relevance subjectivity, there are several classical documentary test collections available, each of them with a set of queries with known relevance judgments, that can be used to test the different new proposals in the field of IR (Baeza-Yates & Ribeiro-Neto, 1999; Salton & McGill, 1983). In this contribution, we will deal with the well-known Cranfield and CACM collections.

As said, up to our knowledge, all the previous applications of machine learning techniques to any of the IRS components trying to optimize both criteria have considered a weighted combination of the said two criteria.

2.3. The IQBE paradigm and its application to IR

IQBE was proposed in Chen et al. (1998) 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—that can be obtained from a preliminary query or from a browsing process in the documentary base—and applying an off-line learning process to automatically generate a query describing the user's needs (as represented by the document set provided by him). The obtained query can then be run in other IRSs to obtain more relevant documents. This way, there is no need that the user interacts with the process as in other query refinement techniques such as relevance feedback (Salton & McGill, 1983).

There have been designed several IQBE algorithms for the different existing IR models. As said, Smith and Smith (1997) proposed the GP algorithm to derive Boolean queries that will be considered in this paper. On the other hand, all of the machine learning methods considered in Chen et al.'s (1998) paper (regression trees, genetic algorithms and simulated annealing) dealt with the vector space model (Salton & McGill, 1983). Moreover, there are several approaches for the derivation of weighted Boolean queries for fuzzy IRSs (Bordogna, Carrara, & Pasi, 1995), such as the GP algorithm of Kraft, Petry, Buckles, and Sadasivan (1997), the niching GA-P method (Cordón, Moya, & Zarco, 2000) and the simulated annealing-GP hybrid (Cordón, Moya, & Zarco, 2002). For descriptions of some of the previous techniques based on EAs, the interested reader can refer to Cordón, Herrera-Viedma, López-Pujalte, et al. (2003).

2.4. EAs and their application to IR

Evolutionary computation (Bäck et al., 1997) 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. There have been four well-defined EAs which have served as the basis for much of the activity in the field: *genetic algorithms* (GAs) (Michalewicz, 1996), *evolution strategies* (Schwefel, 1995), GP (Koza, 1992) and *evolutionary programming* (Fogel, 1991).

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.

GP is based on evolving structures encoding programs such as expression trees. As Boolean and extended Boolean queries can be easily represented in the form of expression trees, GP has been widely used in the IR query learning topic.

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 problems:

(1) *Automatic document indexing*, either by learning the relevant terms to describe them (Gordon, 1988) or their weights (Vrajitoru, 1998), and to design a customized term weighting function (Fan, Gordon, & Pathak, 2004).
(2) *Clustering of documents* (Gordon, 1991) *and terms* (Robertson & Willet, 1994). In both cases, a GA is considered to obtain the cluster configuration.
(3) *Query definition*, by means of an on-line relevance feedback procedure in vector space (Horng & Yeh, 2000; López-Pujalte, Guerrero, & Moya, 2002, 2003; Robertson & Willet, 1996; Yang & Korfhage, 1994) and fuzzy (Sanchez, Miyano, & Bracket, 1995) IRSs, or an off-line IQBE process (Chen et al., 1998) for Boolean (Fernández-Villacanas & Shackleton, 2003; Smith & Smith, 1997) and fuzzy (Cordón, Herrera-Viedma, Luque, Moya, & Zarco, 2003; Cordón et al., 2000; Cordón et al., 2002, Cordón, Moya, & Zarco, 2004; Kraft et al., 1997) IRSs. Also, we find genetic techniques for solving multimodal problems in IR (Boughanem, Chrisment, & Tamine, 1999, 2002, 2003). Notice that, the IQBE approach tackled in the current contribution is included in this group as the final goal is to automatically derive a query representing the user's needs.
(4) *Design of user profiles for IR in the Internet*. IRSs are limited by the lack of personalization in the representation of 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. In Chen and Shahabi (2001), Larsen, Marín, Martín-Bautista, and Vila (2000), Martín-Bautista, Larsen, and Vila (1999), we can find different approaches involving user profiles and GAs.

For a review of several of the previous approaches, see Cordón, Herrera-Viedma, López-Pujalte, et al. (2003).

2.5. Multiobjective EAs and IR

Most of the IQBE approaches in IR evaluate the performance of the derived queries using the two usual criteria, precision and recall (see Section 2.2). Therefore, the optimization of the components of an IRS becomes a clear example of a multiobjective problem.

EAs are very appropriate to solve multiobjective problems. These kinds of problems are characterized by the fact that several objectives have to be simultaneously optimized. Hence, there is not usually a single best solution solving the problem, i.e. being better than the remainder with respect to every objective, as in single-objective optimization. Instead, in a typical multiobjective optimization framework, there is a set of solutions that are superior to the remainder when all the objectives are considered, the *Pareto* set. These solutions are known as *nondominated solutions* (Chankong & Haimes, 1983), while the remainder are known as *dominated solutions*. Since none of the Pareto set solutions is absolutely better than the other nondominated solutions, all of them are equally acceptable as regards the satisfaction of all the objectives.

This way, thanks to the use of a population of solutions, EAs can search many Pareto-optimal solutions in the same run, specifically, many queries with different precision–recall trade-offs in our case.

Evolutionary approaches in multiobjective optimization can be classified into three groups: *plain aggregating approaches*, *population-based nonPareto approaches*, arid *Pareto-based approaches* (Coello et al., 2002; Deb, 2001).

The first group constitutes the extension of classical methods to EAs. The objectives are artificially combined, or aggregated, into a scalar function according to some understanding of the problem, and then the EA is applied in the usual way.[1]

Population-based nonPareto approaches allow us to exploit the special characteristics of EAs. A nondominated individual set is obtained instead of generating only one solution. In order to do so, the selection mechanism is changed. Generally, the best individuals according to each of the objectives are selected, and then these partial results are combined to obtain the new population. An example of a multiobjective GA belonging to this group is Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1985).

Finally, *Pareto-based approaches* seem to be the most active research area on multiobjective EAs nowadays. In fact, algorithms included within this family are divided into two different groups: first and second generation (Coello et al., 2002). They all attempt to promote the generation of multiple nondominated solutions, as the former group, but directly making use of the Pareto-optimality definition.

The difference between the first and the second generation of Pareto-based approaches arises on the use of elitism. Algorithms included within the first generation group, such as Niched Pareto Genetic Algorithm (NPGA), Non-dominated Sorting Genetic Algorithm (NSGA) and Multiple-Objective Genetic Algorithm (MOGA), do not consider this characteristic. On the other hand, second generation Pareto-based multiobjective EAs are based on the consideration of an auxiliary population where the nondominated solutions generated among the different iterations are stored. Examples of the latter family are Strength Pareto EA (SPEA) (Zitzler & Thiele, 1999) (the one considered in this contribution) and SPEA2, NSGA2 and NPGA2, among others. For the description of all of these algorithms, the interested reader can refer to Deb (2001) and Coello et al. (2002).

When multiobjective optimization is tackled, the definition of the quality is substantially more complex than for single-objective optimization problems, since the optimization process itself involves several objectives:

(1) The distance of the resulting nondominated set to the Pareto-optimal front should be minimized.
(2) A good (in most cases uniform) distribution of the solutions found is desirable. The assessment of this criterion might be based on a certain distance metric.
(3) The extent of the obtained nondominated front should be maximized.

Several quantitative metrics have been proposed in the literature to formalize the above definition (or parts of it) (Coello et al., 2002; Deb, 2001; Zitzler, Deb, & Thiele, 2000). Some of them are defined below.

[1] As said, this has been the approach usually followed in the application of EAs to IR.

Given a set of pairwise nondominated decision vectors $X' \subseteq X$, a neighborhood parameter $\sigma > 0$ (to be chosen appropriately), and a distance metric $\|\cdot\|$:

(1) The function $\mathscr{M}_1$ gives the average distance to the Pareto-optimal set $\overline{X} \subseteq X$:

$$\mathscr{M}_1(X') := \frac{1}{|X'|} \sum_{a' \in X'} \min\{\|a' - \bar{a}\|; \bar{a} \in \overline{X}\} \tag{2}$$

(2) The function $\mathscr{M}_2$ takes the distribution in combination with the number of nondominated solutions found into account:

$$\mathscr{M}_2(X') := \frac{1}{|X' - 1|} \sum_{a' \in X'} |\{b' \in X'; \|a' - b'\| > \sigma\}| \tag{3}$$

(3) The function $\mathscr{M}_3$ considers the extent of the front described by X':

$$\mathscr{M}_3(X') := \sqrt{\sum_{i=1}^{m} \max\{\|a'_i - b'_i\|; a', b' \in X'\}} \tag{4}$$

Analogously, Zitzler et al. (2000) define three metrics. $\mathscr{M}^*_1$, $\mathscr{M}^*_2$, and $\mathscr{M}^*_3$ on the objective space. Let Y', $\overline{Y} \subseteq Y$ be the sets of objective vectors that correspond to X' and $\overline{X}$, respectively, and $\sigma^* > 0$ and $\|\cdot\|^*$ as before:

$$\mathscr{M}^*_1(Y') := \frac{1}{|Y'|} \sum_{p' \in X'} \min\{\|p' - \bar{p}\|^*; \bar{p} \in \overline{Y}\} \tag{5}$$

$$\mathscr{M}^*_2(Y') := \frac{1}{|Y' - 1|} \sum_{P' \in X'} |\{q' \in Y'; \|p' - q'\|^* > \sigma^*\}| \tag{6}$$

$$\mathscr{M}^*_3(Y') := \sqrt{\sum_{i=1}^{m} \max\{\|p'_i - q'_i\|^*; p', q' \in Y'\}} \tag{7}$$

3. A multiobjective IQBE EA to learn multiple Boolean queries

Our main objective is to improve Smith and Smith's results obtaining several queries instead of just one in a single run. To do so, we will use a multiobjective focus, incorporating Pareto-based evolutionary multiobjective components into GP, whose good behaviour was demonstrated in Rodríguez-Vazquez, Fonseca, and Fleming (1997).

Firstly we will review Smith and Smith's approach and then introduce our proposal.

3.1. The Smith and Smith's approach to learn Boolean queries

Smith and Smith (1997) proposed an IQBE process to derive Boolean queries based on GP which we extend in the following section to improve the aid possibilities to the users in the query formulation process. Its components are described as follows:

Coding scheme: The Boolean queries are encoded in expression trees, whose terminal nodes are query terms and whose inner nodes are the Boolean operators *AND*, *OR* or *NOT*, as shown in Fig. 2.

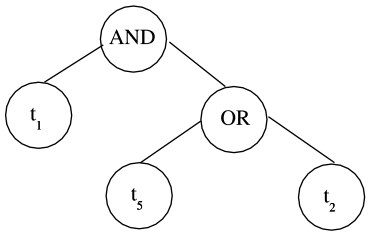

Fig. 2. GP individual representing the query t_1 *AND* (t_2 *OR* t_5).

Selection scheme: Each generation is based on selecting two parents, with the best fitted one having a greater chance to be chosen, and generating two offspring from them. Both offspring are added to the current population.[2]

Genetic operators: The usual GP crossover is considered (Koza, 1992), which is based on randomly selecting one edge in each parent and exchanging both subtrees from these edges between the both parents. No mutation operator is considered.[3]

Generation of the initial population: All the individuals in the first population are randomly generated. A pool is created with all 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.

Fitness function: The following function is maximized:

$$F = \alpha \cdot P + \beta \cdot R$$

where precision P and recall R are computed as shown in Section 2.2, while α and β defined in $\mathscr{R}$ are the weighting factors.

3.2. A new approach to learn Boolean queries by means of multiobjective evolutionary algorithms

As is shown in Section 2.5, there are several kinds of multiobjective EAs. In first generation Pareto-based algorithms the elitism concept is lost. There is no way to assure the presence of the best solution since there is not an only best solution, but a set of them. To solve this, new multiobjective evolutionary models were designed. These models use an external population, where the nondominated solutions found are progressively stored.

With the aim of maintaining the elitism concept, we have considered SPEA (Zitzler & Thiele, 1999) as the multiobjective EA to be incorporated into the basic GP algorithm.[4] This algorithm introduces the elitism concept, explicitly maintaining an external population P_e. This population stores a fixed number of nondominated solutions which have been found since the start of the run.

Fig. 3 shows the scheme of the SPEA algorithm. In each generation, the new nondominated solutions found are compared with the solutions in the existing external population, storing the resulting nondominated solutions on the latter. Furthermore, SPEA uses these elitist solutions, together with those in the current population, in the genetic operations, in the hope to lead the population to good areas in the search space.

[2] Our implementation differs in this point as we consider a classical generational scheme where the intermediate population is created using tournament selection.

[3] We do use a mutation operator which changes a randomly selected term or operator by a random one, or a randomly selected subtree by a randomly generated one.

[4] The proposed multiobjective algorithm has common components with the basic GP algorithm described in the previous section: the coding and selection schemes, the genetic operators and the generation of the initial population. Their description will not be introduced again for the sake of clarity.

```
P <-- Generate the initial population
P_e <-- Non dominated solutions of P

For each generation
    Assign fitness values to elements in P and P_e
    P_aux <-- Tournament_Selection (P ∪ P_e)
    P <-- Cross_and_Mutation (P_aux)
    P_e <-- P_e ∪ Non_Dominated (P)
    P_e <-- Non_Dominated_and_Non_Repeated (P_e)
    If | P_e | > N_e
        Clustering Algorithm
    end if
end for
```

Fig. 3. SPEA's scheme.

Hence, the intermediate population is created from both the current population and the external population by means of tournament selection. This selection process involves randomly choosing a number of individuals of the population, the so called tournament size, with or without replacement, selecting the best individual of this group, and repeating the process until the number of selected individuals matches up with the population size. To perform the selection, there is a need to assign a fitness value to each individual of both populations. The fitness functions considered are:

- Elements of the elitist population:

$$S_i = \frac{n_i}{N+1} \tag{8}$$

  where n_i is the number of solutions in the current population dominated by the ith individual of the elitist population, and N is the size of the current population.
- Elements of the current population:

$$F_j = 1 + \sum_{i \in P_e \text{ and } i \text{ dominates } j} S_i \tag{9}$$

When the intermediate population is created, the genetic operators are used over the new individuals to get a new population of size N. Then, the nondominated solutions existing in the new population are copied to the elitist population P_e, removing dominated and duplicated solutions. Therefore, the new elitist population is composed of the best nondominated solutions found so far, including new and old elitist solutions.

To limit the growth of the elitist population, the size is restricted to N_e solutions using clustering techniques, selecting the solutions closer to the center of each cluster by means of the clustering algorithm shown in Fig. 4.

4. Experiments developed

As said, the experimental study has been developed using the *Cranfield* and *CACM* collections. *Cranfield is* composed of 1398 documents about Aeronautics while *CACM* contains 3204 documents published in the journal *Communications of the ACM* between 1958 and 1979. In both collections, the textual documents have been automatically indexed in the usual way[5] by first extracting the nonstop words and performing

[5] To do so, we use the classical Salton's SMART IRS (Salton, 1971).

```
Assign each element to a cluster
While |P_e | > N_e
        For each pair of clusters,
            Calculate the distance between clusters, like the average
            distance among all their elements
        end for
        Match  both clusters with minimum distance between them
end_while
Choose a solution of each cluster, that with minimum average
distance  to the other cluster elements
```

Fig. 4. SPEA's clustering algorithm.

a stemming process, thus obtaining a total number of 3857 and 7562 different indexing terms, respectively, and then considering the binary indexing to generate the term weights in the document representations.

Both collections have associated a large number of queries (225 in the Cranfield collection and 64 in the CACM collection). In our problem, each query generates a different experiment and our goal involves automatically deriving a set of queries that describes the information contents of the set of documents associated with it. Instead of working with the complete query set, we have selected a representative sample that allow us to study the behavior of our proposal.

The experimental environment considered is graphically shown in Fig. 5. The role of the user who provides documents will be played by the queries associated with the considered collection, and more exactly, by the relevance judgments associated with them. In this way, for example, if there are 29 relevant documents for query 1 in the Cranfield collection, this query will mimic a situation in which the user provides 29 documents related to his information need. Besides, the remaining, nonrelevant 1369 documents (1398—29) will be considered as nonrelevant documents provided to the IQBE process as well. We should remark that, opposite to relevance feedback techniques in which the collection query structures are considered and processed in the same way that documents, we only use the relevance judgments of the existing queries, as the IQBE process learns the query structures starting from scratch.

So, among the 225 queries associated to the Cranfield collection, we have selected a representative subset: on the one hand, those queries presenting 20 or more relevant documents have been taken into account; on the other hand, 10 queries with 15 or less relevant documents have also been chosen to test the performance of our approach with queries presenting a lesser number of relevant documents (representing a situation where the user provides a small number of documents to the IQBE process). The resulting 17 queries (numbers 1, 2, 3, 7, 8, 11, 19, 23, 26, 38, 39, 40, 47, 73, 157, 220 and 225) have 29, 25, 9, 6, 12, 8, 10, 33, 7, 11, 14, 13, 15, 21, 40, 20 and 25 relevant documents associated, respectively. On the other hand, 18 queries have been selected from the 64 associated to the CACM collection (numbers 4, 7, 9, 10, 14, 19, 24, 25, 26, 27, 40, 42, 43, 45, 58, 59, 60 and 61), those 13 presenting more than 20 relevant documents and five with less than 15 relevant documents (12, 28, 9, 35. 44, 11, 13, 51, 30, 29, 10, 21, 41, 26, 30, 43, 27 and 31 relevant documents, respectively). We have selected these queries in order to have enough chances to show the performance advantages of our multiobjective algorithm.

The experiments developed involve to run our multiobjective proposal as well as the Smith and Smith's one as comparison algorithm. Each algorithm has been run 10 times with different initializations for each selected query during the same fixed number of fitness function evaluations (100,000) in a 2.4 GHz Pentium IV computer with 1Gb of RAM.[6] The common parameter values considered are a maximum of 20 nodes for the trees,[7] 0.8 arid 0.2 for the crossover and mutation probabilities, respectively, 5 for the tournament

[6] The Smith & Smith's algorithm spends more or less 2:50 and 6:20 min when working with Cranfield arid CACM queries, respectively, whilst our proposal approximately takes 3 and 6:40 min.

[7] In practice, the maximum number of nodes is 19 since the expression trees implemented are binary arid therefore the number of nodes of a correct query tree has to be odd.

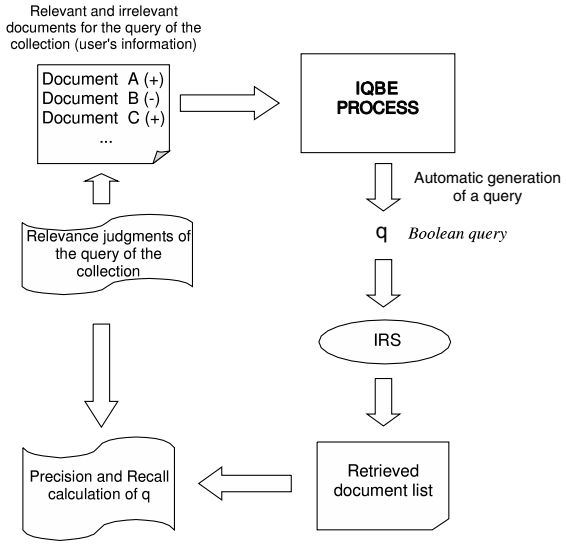

Fig. 5. Graphical representation of the IQBE experimental environment considered.

size arid a population size of M = 1600 queries. The high value of the latter parameter is because it is well known that GP requires large population sizes to achieve good performance. Apart from these parameters. Smith and Smith's algorithm uses a typical setting for the weights in the fitness function ($(\alpha, \beta) = (1.2, 0.8)$), and the size of the elitist population has been fixed to 100 in SPEA.

In Section 2.5, a set of metrics usually considered to measure the quality of the Pareto sets has been shown. Specifically, we have used three different metrics: M_2^* arid M_3^*, defined in the aforementioned section, and the number of nondominated solutions in the Pareto set.

The metric $\mathscr{M}_1$ has been given up since it cannot be used as we do not know the optimal Pareto fronts; furthermore, this metric does not consider the Pareto set distribution. Therefore, we have used $\mathscr{M}_2$ (that measures the distribution of nondominated solutions) and $\mathscr{M}_3$ (which measures the size of the area that contains the nondominated solutions). The reason of using $\mathscr{M}_2^*$ and $\mathscr{M}_3^*$ instead of $\mathscr{M}_2$ and $\mathscr{M}_3$ comes from the fact that we are interested in that Boolean queries learned are well distributed in the objective space, to be able to obtain several queries with different precision–recall trade-offs.

Notice that, since our problem is composed of just two objectives, $\mathscr{M}_3^*$ is equal to the distance among the objective vectors of the two outer solutions (hence, the maximum possible value is $\sqrt{2} = 1.4142$).

Although the main aim of this paper is to get an IQBE algorithm generating several queries with a different precision–recall trade-off in a single run, we are going to establish a procedure to compare the performance of the proposed technique with that of the original Smith and Smith's proposal.

To do so, the best average solution in precision and recall is selected from the Pareto set derived by our multiobjective IQBE algorithm. This solution is obtained as shown below:

(1) 1000 pairs of random numbers are generated (w_{i1}, w_{i2}), $w_{i1} \in [0, 1]$, $w_{i2} = 1 - w_{i1}$.
(2) For each Boolean query S_j included in the Pareto set, with recall R_j and precision P_j, the next index is computed:

$$\text{Average}(S_j) = \frac{\sum_{i=1}^{1000} w_{i1} \cdot R_j + w_{i2} \cdot P_j}{1000} \tag{10}$$

(3) The solution that maximizes the Average value is selected.

5. Results and analysis of results

5.1. Analysis of the Pareto sets derived

Tables 1 and 2 show several statistics corresponding to our multiobjective proposal. These tables collects several data, about the composition of the 10 Pareto sets generated for each query, always showing the averaged value and its standard deviation. From left to right, the columns contain the number of nondominated solutions obtained (#*p*), equal to the number of different objective vectors (i.e., precision–recall pairs) existing among them, and the values of the two multiobjective EA metrics selected, $\mathscr{M}_2^*$ and $\mathscr{M}_3^*$ all of them followed by their respective standard deviation values.

The main aim of this paper has been clearly fulfilled since the Pareto fronts obtained are very well distributed, as demonstrated by the high values in the $\mathscr{M}_2^*$ and $\mathscr{M}_3^*$ metrics. So, we can see that all runs generate a proportional number of Boolean queries with different precision–recall trade-offs according to the number of relevant documents associated with them (for those cases where a larger number of relevant documents are provided, a larger number of different queries are obtained in the Pareto sets); and that standard deviation values are around 0.4 and 0.6 in the Cranfield and CACM collections, respectively. The values of the $\mathscr{M}_2^*$ and $\mathscr{M}_3^*$ metrics are very appropriate as well, emphasizing the values of the latter, very closer to 1.4142, the maximum possible value. This shows us how the Pareto fronts generated cover a wide area in the space.

More specifically, the experiments generated from queries 157 of the Cranfield collection and 25 of the CACM collection are those deriving the Pareto sets with the best average values. So, query 157 generates a Pareto front with around 24 different solutions, and obtains a value of 10.56 for the distribution of these solutions over it ($\mathscr{M}_2^*$) and a value of 1.29 for the ($\mathscr{M}_3^*$) metric. Similarly, query 25 of CACM has the following values associated: 29.7 for the number of nondominated solutions in the Pareto front, and 12.9 and 1.316 for $\mathscr{M}_2^*$ and $\mathscr{M}_3^*$ metrics, respectively.

As an example, Figs. 6 and 7 graphically show the Pareto fronts obtained for queries 157 of Cranfield and 25 of CACM, respectively, representing the recall values in the *X*-axis and the precision ones on the

Table 1
Statistics of the Pareto sets obtained by the proposed SPEA-GP IQBE algorithm on the Cranfield collection

#*q*	#*p*	$\sigma_{\#p}$	M_2^*	$\sigma_{M_2^*}$	M_3^*	$\sigma_{M_3^*}$
Cranfield						
1	15.800	0.675	6.909	0.291	1.237	0.007
2	11.700	0.736	5.188	0.314	1.190	0.014
3	2.000	0.141	0.950	0.111	0.649	0.088
7	2.600	0.155	1.300	0.077	1.009	0.023
8	7.700	0.375	3.599	0.162	1.209	0.012
11	3.200	0.190	1.600	0.095	1.040	0.028
19	2.100	0.170	1.000	0.122	0.645	0.080
23	18.400	0.551	8.038	0.227	1.235	0.012
26	5.400	0.155	2.700	0.077	1.269	0.012
38	4.700	0.318	2.333	0.147	1.040	0.031
39	7.800	0.276	3.584	0.125	1.196	0.010
40	5.900	0.411	2.764	0.213	1.093	0.021
47	6.100	0.330	2.940	0.157	1.117	0.016
73	11.100	0.499	5.049	0.203	1.192	0.012
157	*24.400*	0.738	*10.559*	0.319	*1.290*	0.004
220	9.200	0.237	4.191	0.104	1.137	0.011
225	14.100	0.640	6.180	0.276	1.238	0.006

Table 2
Statistics of the Pareto sets obtained by the proposed SPEA-GP IQBE algorithm on the CACM collection

#q	#p	$\sigma_{\#p}$	M_2^*	$\sigma_{M_2^*}$	M_3^*	$\sigma_{M_3^*}$
CACM						
4	6.300	0.285	3.077	0.142	1.218	0.015
7	13.800	0.486	6.153	0.198	1.239	0.011
9	5.400	0.473	2.700	0.237	1.165	0.053
10	15.700	1.030	6.845	0.447	1.228	0.024
14	12.600	0.429	5.214	0.171	1.231	0.028
19	8.000	0.200	3.640	0.116	1.268	0.008
24	7.500	0.158	3.507	0.074	1.233	0.010
25	*29.700*	0.694	*12.900*	0.299	1.316	0.007
26	14.700	0.511	6.471	0.229	1.230	0.011
27	15.700	0.694	6.875	0.324	1.220	0.010
40	4.900	0.298	2.450	0.149	1.153	0.023
42	13.500	0.292	6.086	0.150	1.285	0.006
43	22.800	0.834	9.965	0.361	1.291	0.008
45	13.200	0.629	5.933	0.260	1.260	0.007
58	16.600	0.751	7.414	0.344	*1.337*	0.004
59	21.400	0.951	9.375	0.408	1.282	0.011
60	12.900	0.591	5.738	0.226	1.187	0.012
61	15.500	0.552	6.841	0.238	1.258	0.011

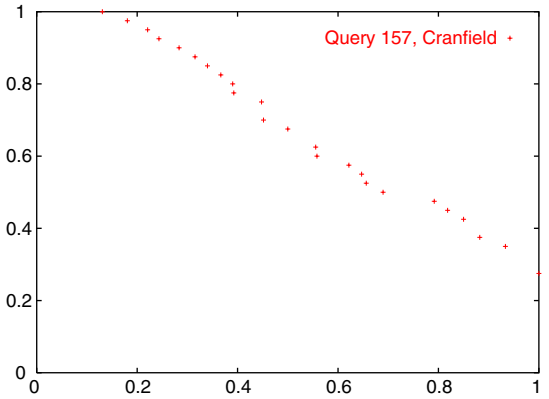

Fig. 6. Pareto front obtained for query 157 of Cranfield.

Y-axis. As done in Zitzler et al. (2000), the Pareto sets obtained in the 10 runs performed for each query were put together, and the dominated solutions where removed from the unified set before plotting the curves.

The problem found is that the number of solutions presenting different precision–recall values (different objective value arrays) can be a little bit low with respect to the size of the elitist population. The main reason of this behavior is found in the way we measure the similarity between a pair of solutions (queries).

Two solutions can be equal in the objective space or in the decision space. In IR, if we work in the objective space, two solutions will be equal when their precision and recall values are the same, regardless of its structure. However, if we work in the decision space, two solutions will be equal when their structures (i.e., query compositions) coincide.

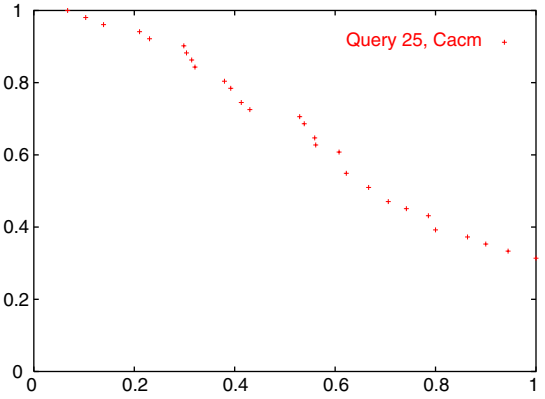

Fig. 7. Pareto front obtained for query 25 of CACM.

In this proposal, we have decided to work in the objective space, supported by the recommendation of the SPEA algorithm's authors (Zitzler & Thiele, 1999) of using the clustering algorithm in this space. Nevertheless, we notice that, with this criterion, several optimal solutions were eliminated. The only difference between these solutions and those of the elitist population is the query composition. This considerably reduces the final number of individuals in the elitist population.

To solve this, we decided to work in the decision space, utilizing the *edit* or *Levenshtein distance* (Levenshtein, 1996) to measure the similarity between expression trees. Although this measure increased the number of solutions, the run time also increased.[8] Finally, we decided to choose the original option, leaving the search for new similarity functions between query expressions for future works.

5.2. Analysis of the "Best" queries derived

Before developing this new analysis, we should again remark that, though the obtained results can be used for comparing our algorithm with the basic one, the fundamental SPEA-GP aim is not to obtain the best individual query but we are trying to find a set of queries with a different precision–recall trade-off.

The results obtained by the basic algorithm on the Cranfield and the CACM collections are shown in Tables 3 and 4, respectively, In both tables, $\#q$ stands for the corresponding query number, Sz for the average of the generated queries size and σ_{Sz} for its standard deviation, F and σ_F for the average and standard deviation of the fitness value, respectively, P and R for the average of the precision and recall values (respectively, σ_R and σ_P for their standard deviations), $\#rt$ for the number of documents retrieved by the query, and $\#rr$ for the number of relevant documents retrieved, both with their standard deviations, $\sigma_{\#rt}$ and $\sigma_{\#rr}$.

Tables 5 and 6 show the results obtained by our multiobjective proposal. These results come from the best average queries derived, chosen by means of the procedure described in Section 4. Notice that, in both tables, the best results are shown in boldface, fn view of these results, the performance of our proposal is very significant. On the one hand, the fitness value provided by our multiobjective proposal improves Smith and Smith's results in all the queries of the Cranfield collection,[9] and in 16 of the 18 queries of the CACM collection. In both collections, the average precision value is slightly reduced, while the average recall value

[8] On the Cranfield collection, the genotypic approach spends more or less 3 min whilst the phenotypic variant takes around 30 min.

[9] Notice that the same weight values for the fitness function considered by the Smith and Smith's approach have been used to compute this value with the query generated from our algorithm.

Table 3
Results obtained by the basic Smith and Smith's IQBE algorithm on the Cranfield collection

#q	Sz	σ_{Sz}	F	σ_F	P	σ_P	R	σ_R	#rr	$\sigma_{\#rr}$	#rt	$\sigma_{\#rt}$
Cranfield												
1	18.800	0.600	1.385	0.055	1.000	0.000	0.231	0.069	6.700	2.002	6.700	2.002
2	18.800	0.600	1.507	0.144	1.000	0.000	0.384	0.180	9.600	4.499	9.600	4.499
3	18.800	0.190	1.884	0.013	1.000	0.000	0.855	0.016	7.700	0.145	7.700	0.145
7	18.800	0.190	1.533	0.021	1.000	0.000	0.417	0.026	2.500	0.158	2.500	0.158
8	18.600	0.253	1.327	0.024	1.000	0.000	0.158	0.030	1.900	0.359	1.900	0.359
11	18.800	0.190	1.580	0.031	1.000	0.000	0.475	0.039	3.800	0.310	3.800	0.310
19	19.000	0.000	1.768	0.038	1.000	0.000	0.710	0.048	7.100	0.478	7.100	0.478
23	19.000	0.000	1.385	0.031	1.000	0.000	0.197	0.039	6.500	1.285	6.500	1.285
26	18.800	0.190	1.383	0.024	1.000	0.000	0.228	0.030	1.600	0.210	1.600	0.210
38	19.000	0.000	1.571	0.028	1.000	0.000	0.464	0.035	5.100	0.386	5.100	0.386
39	19.000	0.000	1.417	0.018	1.000	0.000	0.271	0.022	3.800	0.310	3.800	0.310
40	19.000	0.000	1.588	0.023	1.000	0.000	0.485	0.029	6.300	0.375	6.300	0.375
47	18.800	0.190	1.397	0.017	1.000	0.000	0.247	0.021	3.700	0.318	3.700	0.318
73	18.800	0.600	1.451	0.080	1.000	0.000	0.314	0.100	6.600	2.107	6.600	2.107
157	19.000	0.000	1.330	0.030	1.000	0.000	0.163	0.037	6.500	1.500	6.500	1.500
220	19.000	0.000	1.521	0.078	1.000	0.000	0.390	0.097	7.800	1.939	7.800	1.939
225	19.000	0.000	1.501	0.041	1.000	0.000	0.376	0.051	9.400	1.281	9.400	1.281

Table 4
Results obtained by the basic Smith and Smith's IQBE algorithm on the CACM collection

#q	Sz	σ_{Sz}	F	σ_F	P	σ_P	R	σ_R	#rr	$\sigma_{\#rr}$	#rt	$\sigma_{\#rt}$
CACM												
4	18.800	0.190	1.420	0.025	1.000	0.000	0.275	0.031	3.300	0.375	3.300	0.375
7	19.000	0.000	1.414	0.021	1.000	0.000	0.268	0.026	7.500	0.738	7.500	0.738
9	19.000	0.000	1.422	0.049	1.000	0.000	0.278	0.061	2.500	0.552	2.500	0.552
10	19.000	0.600	*1.399*	0.114	1.000	0.000	0.249	0.143	8.700	5.001	8.700	5.001
14	19.000	0.000	1.472	0.018	0.604	0.018	0.934	0.007	41.100	0.300	68.100	2.587
19	18.600	0.379	1.353	0.048	1.000	0.000	0.191	0.059	2.100	0.655	2.100	0.655
24	18.800	0.190	1.440	0.027	1.000	0.000	0.300	0.033	3.900	0.435	3.900	0.435
25	19.000	0.000	1.349	0.048	1.000	0.000	0.186	0.060	9.500	3.041	9.500	3.041
26	19.000	0.000	1.468	0.032	0.994	0.005	0.343	0.044	10.300	1.327	10.400	1.380
27	18.600	0.379	1.327	0.016	1.000	0.000	0.159	0.020	4.600	0.586	4.600	0.586
40	18.200	0.419	1.472	0.023	1.000	0.000	0.340	0.029	3.400	0.290	3.400	0.290
42	18.800	0.190	1.322	0.022	1.000	0.000	0.152	0.028	3.200	0.580	3.200	0.580
43	19.000	0.000	1.389	0.073	1.000	0.000	0.237	0.091	9.700	3.761	9.700	3.761
45	18.800	0.188	1.311	0.014	1.000	0.000	0.138	0.017	3.600	0.452	3.600	0.452
58	18.800	0.190	*1.387*	0.027	1.000	0.000	0.233	0.034	7.000	1.010	7.000	1.010
59	19.000	0.000	1.409	0.067	0.976	0.055	0.298	0.090	12.800	3.868	13.300	4.605
60	19.000	0.000	1.487	0.020	1.000	0.000	0.359	0.024	9.700	0.664	9.700	0.664
61	19.000	0.000	1.394	0.070	1.000	0.000	0.242	0.088	7.500	2.729	7.500	2.729

is significantly increased in the most of the cases, being these variations more pronounced in the CACM collection. In the same way, the number of relevant documents retrieved is also significantly increased. It seems that the diversity induced by the Pareto-based selection and the use of an elitist population make SPEA-GP converge to better space zones.

This way, we can conclude that our multiobjective IQBE approach is not only a good way to derive different queries with several precision–recall trade-offs but also to obtain individual queries with high retrieval accuracy.

Table 5
Results obtained by the our multiobjective proposal, SPEA-GP algorithm on the Cranfield collection

#q	Sz	σ_{Sz}	F	σ_F	P	σ_P	R	σ_R	#rr	$\sigma_{\#rr}$	#rt	$\sigma_{\#rt}$
Cranfield												
1	19.000	0.000	*1.515*	0.041	0.955	0.057	0.462	0.068	13.400	1.960	14.200	2.960
2	19.000	0.000	*1.529*	0.073	0.858	0.124	0.624	0.136	15.600	3.412	19.200	7.318
3	17.200	1.112	*1.905*	0.013	0.980	0.013	0.911	0.021	8.200	0.190	8.400	0.290
7	18.400	0.290	*1.780*	0.023	0.983	0.016	0.750	0.026	4.500	0.158	4.600	0.210
8	18.600	0.253	*1.533*	0.021	1.000	0.000	0.417	0.026	5.000	0.316	5.000	0.316
11	18.600	0.253	*1.770*	0.019	0.975	0.016	0.750	0.031	6.000	0.245	6.200	0.340
19	18.800	0.188	*1.896*	0.019	0.973	0.017	0.910	0.017	9.100	0.170	9.400	0.322
23	19.000	0.000	*1.464*	0.063	0.937	0.116	0.424	0.144	14.000	4.754	16.000	8.866
26	17.600	0.569	*1.462*	0.017	0.980	0.019	0.357	0.030	2.300	0.318	2.600	0.290
38	19.000	0.000	*1.719*	0.024	0.990	0.009	0.664	0.034	7.300	0.375	7.400	0.429
39	18.200	0.759	*1.570*	0.019	0.980	0.013	0.493	0.033	6.900	0.457	7.100	0.556
40	19.000	0.000	*1.635*	0.028	0.942	0.030	0.631	0.028	8.200	0.369	8.900	0.670
47	19.000	0.000	*1.691*	0.024	0.938	0.022	0.707	0.019	10.600	0.290	11.400	0.514
73	19.000	0.000	*1.570*	0.067	0.959	0.041	0.524	0.067	11.000	1.414	11.500	1.628
157	19.000	0.000	*1.359*	0.091	0.896	0.189	0.355	0.172	14.200	6.867	20.700	23.469
220	18.800	0.600	*1.610*	0.051	0.945	0.074	0.595	0.082	11.900	1.640	12.800	2.821
225	19.000	0.000	*1.508*	0.035	0.992	0.023	0.396	0.052	9.900	1.300	10.000	1.483

Table 6
Results obtained by the our multiobjective proposal, SPEA-GP algorithm on the CACM collection

#q	Sz	σ_{Sz}	F	σ_F	P	σ_P	R	σ_R	#rr	$\sigma_{\#rr}$	#rt	$\sigma_{\#rt}$
CACM												
4	19.000	0.000	*1.577*	0.030	0.920	0.036	0.592	0.034	7.100	0.411	8.000	0.787
7	18.800	0.190	*1.561*	0.016	0.913	0.019	0.582	0.033	16.300	0.928	18.100	1.315
9	18.000	0.583	*1.538*	0.041	0.941	0.032	0.511	0.079	4.600	0.710	5.200	1.028
10	19.000	0.000	1.330	0.064	0.524	0.081	0.877	0.069	30.700	2.410	60.900	15.248
14	18.800	0.600	*1.474*	0.026	0.601	0.022	0.941	0.011	41.400	0.490	69.000	2.864
19	19.000	0.000	*1.491*	0.014	1.000	0.000	0.364	0.018	4.000	0.200	4.000	0.200
24	18.600	0.379	*1.566*	0.015	0.982	0.017	0.485	0.029	6.300	0.375	6.500	0.534
25	18.600	0.800	*1.383*	0.047	0.970	0.054	0.273	0.056	13.900	2.879	14.500	3.801
26	19.000	0.000	*1.558*	0.020	0.947	0.032	0.527	0.043	15.800	1.279	17.500	2.491
27	19.000	0.000	*1.488*	0.016	0.994	0.006	0.369	0.024	10.700	0.708	10.800	0.772
40	17.200	1.147	*1.639*	0.033	0.953	0.030	0.620	0.046	6.200	0.465	6.700	0.708
43	19.000	0.000	*1.407*	0.095	0.875	0.134	0.446	0.093	18.300	3.796	22.300	9.263
42	18.800	0.190	*1.450*	0.011	0.980	0.019	0.343	0.028	7.200	0.580	7.500	0.840
45	19.000	0.000	*1.480*	0.013	0.930	0.029	0.454	0.044	11.800	1.14	13.200	1.719
58	18.800	0.190	1.356	0.008	0.990	0.009	0.210	0.013	6.300	0.401	6.400	0.473
59	19.000	0.000	*1.463*	0.061	0.817	0.075	0.602	0.064	25.900	2.737	32.200	5.980
60	19.000	0.000	*1.598*	0.017	0.988	0.007	0.515	0.025	13.900	0.670	14.100	0.741
61	19.000	0.000	*1.466*	0.077	0.931	0.109	0.435	0.087	13.500	3.008	15.100	5.394

6. Concluding remarks

The automatic derivation of Boolean queries has been considered by incorporating a second generation multiobjective evolutionary approach, SPEA, to an existing GP-based IQBE proposal. The proposed approach has performed appropriately in 35 queries, 17 of the well known Cranfield collection, and 18 of the CACM collection, in terms of absolute retrieval performance arid of the quality of the obtained Pareto sets, allowing us to derive a set of queries with different precision–recall trade-offs.

In our opinion, many different future works arise from this preliminary study. Firstly, we will search for new functions to measure the similarity between expression trees with the purpose of being able to work in the decision space. On the other hand, preference information of the user on the kind of queries to be derived can be included in the Pareto-based selection scheme in the form of a goal vector whose values are adapted during the evolutionary process (Fonseca & Fleming, 1993). On the other hand, the proposed IQBE algorithm can be extended to other kinds of IRSs based on complex query languages such as extended Boolean (fuzzy) IRSs (Herrera-Viedma, 2001; Herrera-Viedma, Cordón, Luque, López, & Muñoz, 2003) (several preliminaries works on this topic can be found in Cordón, Herrera-Viedma, Luque, et al., 2003; Cordón et al., 2004). Moreover, a training-test validation procedure can be considered to test the real application of the proposed IQBE algorithm.

Acknowledgement

This research has been supported by CICYT under projects TIC2003-07977 and TIC2003-00877 with FEDER fundings.

References

Bäck, T., Fogel, D., & Michalewicz, Z. (Eds.). (1997). *Handbook of evolutionary computation*. IOP Publishing and Oxford University Press.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison.

Bordogna, G., Carrara, P., & Pasi, G. (1995). Fuzzy approaches to extend Boolean information retrieval. In P. Bosc & J. Kacprzyk (Eds.), *Fuzziness in database management systems* (pp. 231–274). Springer-Verlag.

Boughanem, M., Chrisment, C., & Tamine, L. (1999). Genetic approach to query space exploration. *Information Retrieval, 1*, 175–192.

Boughanem, M., Chrisment, C., & Tamine, L. (2002). On using genetic algorithms for multimodal relevance optimization in information retrieval. *Journal of the American Society for Information Science and Technology, 53*(11), 934–942.

Boughanem, M., Chrisment, C., & Tamine, L. (2003). Multiple query evaluation based on an enhanced genetic algorithm. *Information Processing & Management, 39*, 215–231.

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective decision making theory and methodology*. North-Holland.

Chen, H., Shankaranarayanan, G., She, L., & Iyer, A. (1998). A machine learning approach to Inductive Query by Examples: An experiment using relevance feedback, IDS, genetic algorithms, and simulated annealing. *Journal of the American Society for Information Science, 49*(8), 693–705.

Chen, Y., & Shahabi, C. (2001). Automatically improving the accuracy of user profiles with genetic algorithm. In *Proceedings of international conference on artificial intelligence and soft computing*, Cancun, Mexico.

Coello, C. A., Van Veldhuizen, D. A., & Lamant, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers.

Cordón, O., Herrera-Viedma, E., López-Pujalte, C., Luque, M., & Zarco, C. (2003). A review of the application of evolutionary computation to information retrieval. *International Journal of Approximate Reasoning, 34*, 241–264.

Cordón, O., Herrera-Viedma, E., Luque, M., Moya, F., & Zarco, C. (2003). Analyzing the performance of a multiobjective GA-P algorithm for learning fuzzy queries in a machine learning environment. In *Lecture notes in artificial intelligence, vol. 2715. Proceedings of the 10th IFSA world congress*, Istanbul, Turkey (pp. 611–615).

Cordón, O., Moya, F., & Zarco, C. (2000). A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system. *Mathware & Soft Computing, 7*(2–3), 309–322.

Cordón, O., Moya, F., & Zarco, C. (2002). A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. *Soft Computing, 6*(5), 308–319.

Cordón, O., Moya, F., & Zarco, C. (2004). Automatic learning of multiple extended Boolean queries by multiobjective GA-P algorithms. In V. Loia, M. Nikravesh, & L. A. Zatleh (Eds.), *Fuzzy logic and the internet* (pp. 40–47). Springer.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.

Fan, W., Gordon, M., & Pathak, P. (2004). A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing & Management, 40*(4), 587–602.

Fernández-Villacanas, J., & Shackleton, M. (2003). Investigation of the importance of the genotype–phenotype mapping in information retrieval. *Future Generation Computer Systems, 19*(1), 55–68.

Fogel, D. (1991). *System identification through simulated evolution: A machine learning approach*. USA: Ginn Press.

Fonseca, C., & Fleming, P. (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In *Proceedings of the fifth international conference on genetic algorithms* (pp. 416–423). San Mateo, CA: Morgan Kaufmann.

Gordon, M. (1988). Probabilistic and genetic algorithms for document retrieval. *Communications of the ACM, 31*(10), 1208–1218.

Gordon, M. (1991). User-based document clustering by redescribing subject description with a genetic algorithm. *Journal of the American Society for Information Science, 42*(5), 311–322.

Herrera-Viedma, E. (2001). Modeling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach. *Journal of the American Society for Information Science and Technology, 52*(6), 460–475.

Herrera-Viedma, E., Cordón, O., Luque, M., López, A. G., & Muñoz, A. M. (2003). A model of fuzzy linguistic IRS based on multi-granular linguistic information. *International Journal of Approximate Reasoning, 34*, 221–239.

Horng, J., & Yeh, C. (2000). Applying genetic algorithms to query optimization in document retrieval. *Information Processing & Management, 36*, 737–759.

Koza, J. (1992). *Genetic programming. On the programming of computers by means of natural selection*. The MIT Press.

Kraft, D., Petry, F., Buckles, B., & Sadasivan, T. (1997). Genetic algorithms for query optimization in information retrieval: Relevance feedback. In E. Sanchez, T. Shibata, & L. Zadeh (Eds.), *Genetic algorithms and fuzzy logic systems* (pp. 155–173). World Scientific.

Larsen, H., Marín, N., Martín-Bautista, M. J., & Vila, M. A. (2000). Using genetic feature selection for optimizing user profiles. *Mathware & Soft Computing, 7*(2–3), 275–286.

Levenshtein, V. I. (1996). Binary codes of correcting deletions, insertions and reversal. *Soviet Physics Doklady, 6*, 705–710.

López-Pujalte, C., Guerrero, V., & Moya, F. (2002). A test of genetic algorithms in relevance feedback. *Information Processing & Management, 38*, 793–805.

López-Pujalte, C., Guerrero, V., & Moya, F. (2003). Order-based fitness functions for genetic algorithms applied to relevance feedback. *Journal of the American Society for Information Science and Technology, 54*(2), 152–160.

Martín-Bautista, M., Larsen, H., & Vila, M. (1999). A fuzzy genetic algorithm approach to an adaptive information retrieval agent. *Journal of the American Society for Information Science, 50*(9), 760–771.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. Springer-Verlag.

Robertson, A., & Willet, P. (1994). Generation of equifrequent groups of words using a genetic algorithm. *Journal of Documentation, 50*(3), 213–232.

Robertson, A., & Willet, P. (1996). An upperbound to the performance for ranked-output searching: Optimal weighting of query terms using a genetic algorithm. *Journal of Documentation, 52*(4), 405–420.

Rodríguez-Vazquez, K., Fonseca, C. M., & Fleming, P. J. (1997). Multiobjective genetic programming: A nonlinear system identification application. In *Late breaking papers at the genetic programming 1997 conference* (pp. 207–212).

Salton, G. (1971). *The smart retrieval system. Experiments in automatic document processing*. Englewood Cliffs: Prentice-Hall.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval*. McGraw-Hill.

Sanchez, E., Miyano, H., & Bracket, J. (1995). Optimization of fuzzy queries with genetic algorithms. Application to a data base of patents in biomedical engineering. In *Proceedings of VI IFSA congress*, Sao-Paulo, Brazil (pp. 293–296).

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic algorithms and their applications. Proceedings of the first international conference on genetic algorithms* (pp. 93–100). Lawrence Erlbaum.

Schwefel, H.-P. (1995). Evolution and optimum seeking. *Sixth generation computer technology series*. John Wiley and Sons.

Smith, M., & Smith, M. (1997). The use of genetic programming to build Boolean queries for text retrieval through relevance feedback. *Journal of Information Science, 23*(6), 423–431.

Van Rijsbergen, C. (1979). *Information retrieval* (2nd ed.). Butterworth.

Vrajitoru, D. (1998). Crossover improvement for the genetic algorithm in information retrieval. *Information Processing & Management, 34*(4), 405–415.

Yang, J., & Korfhage, R. (1994). Query modifications using genetic algorithms in vector space models. *International Journal of Expert Systems, 7*(2), 165–191.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation, 8*(2), 173–195.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.