Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 160 (2009) 2192–2205

FUZZY sets and systems

www.elsevier.com/locate/fss

Applying multi-objective evolutionary algorithms to the automatic learning of extended Boolean queries in fuzzy ordinal linguistic information retrieval systems[☆]

A.G. López-Herrera*, E. Herrera-Viedma, F. Herrera

Department of Computer Sciences and A.I., University of Granada, E-18071-Granada, Spain

Available online 26 February 2009

Abstract

The performance of information retrieval systems (IRSs) is usually measured using two different criteria, precision and recall. Precision is the ratio of the relevant documents retrieved by the IRS in response to a user's query to the total number of documents retrieved, whilst recall is the ratio of the number of relevant documents retrieved to the total number of relevant documents for the user's query that exist in the documentary database. In fuzzy ordinal linguistic IRSs (FOLIRSs), where extended Boolean queries are used, defining the user's queries in a manual way is usually a complex task. In this contribution, our interest is focused on the automatic learning of extended Boolean queries in FOLIRSs by means of multi-objective evolutionary algorithms considering both mentioned performance criteria. We present an analysis of two well-known general-purpose multi-objective evolutionary algorithms to learn extended Boolean queries in FOLIRSs. These evolutionary algorithms are the *non-dominated sorting genetic algorithm* (NSGA-II) and the *strength Pareto evolutionary algorithm* (SPEA2).
© 2009 Elsevier B.V. All rights reserved.

Keywords: Information retrieval systems; Genetic programming; Inductive query by example; Multi-objective evolutionary algorithms; Query learning

1. Introduction

Information retrieval (IR) may be defined as the problem of selecting documentary information from storage in response to searches provided by a user in form of queries [1,2]. Information retrieval systems (IRSs) deal with documentary databases containing textual, pictorial or vocal information. They process user queries to allow the user to access relevant information in an appropriate time interval.

Most of the commercial IRSs are based on the Boolean IR model [3], which presents some limitations. Due to this fact, some paradigms have been designed to extend this IR model and overcome its problems. One of the main drawbacks is that Boolean IRSs lack the ability to deal with imprecision and subjectivity, both of which are inherently

[☆] This work has been supported by the projects: FUZZY-LING, Cod. TIN2007-61079 and Proyecto de Excelencia de la Junta de Andalucia SAINFOWEB, Cod. 00602.

* Corresponding author. Tel.: +34 958 240466; fax: +34 958 243317.

E-mail addresses: lopez-herrera@decsai.ugr.es (A.G. López-Herrera), viedma@decsai.ugr.es (E. Herrera-Viedma), herrera@decsai.ugr.es (F. Herrera).

0165-0114/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2009.02.013

present in the IR process. This is the reason why fuzzy IR [4–6] has emerged as an active research area in the past decades.

Many fuzzy IR models assume numeric weights to characterize the contents of the desired documents [4,7–9]. However, it seems natural to characterize the contents of the desired documents by explicitly associating a linguistic weight to elements in a query, such as "important" or "very important", instead of a numerical value. So, some fuzzy linguistic IRS models [10–18] have been proposed using a *fuzzy linguistic approach* [19–21] to model the query weights. We call these models "fuzzy ordinal linguistic IRSs" (FOLIRSs).

FOLIRSs present a query language that allows the user to build fuzzy queries composed of statements of ordinal linguistic weighted terms joined by Boolean operators AND, OR and NOT. This query language allows the system to improve the retrieval activity solving the problems associated with the Boolean IR model. However, to build fuzzy ordinal linguistic queries (FOLQs) is not usually easy neither very intuitive [1]. This problem becomes a more serious issue if the users do not have previous experience with the model. A possible solution to overcome this problem is to build automatic aid tools to assist users to express their information needs by means of FOLQs. The *inductive query by example* (IQBE) paradigm [22], where a query describing the information contexts of a set of key documents provided by the user is automatically derived or learned, can be an useful paradigm to assist users to express FOLQs. The most well-known, in the context of fuzzy IR, IQBE approach is that of Kraft et. al. [23], which is based on *genetic programming* (GP) [24], called IQBE-GP, that derives fuzzy queries with numeric weights. As it is usual in the field [25] this approach is guided by a weighted fitness function combining the classical retrieval accuracy criteria, *precision* and *recall* [3]. We will use this approach of IQBE-GP together with a FOLIRS model in this paper.

Given that the retrieval performance of an IRS is usually measured in terms of precision and recall criteria, the optimization of any of its components, and concretely the automatic learning of queries, is a clear example of a multi-objective problem. Evolutionary algorithms (EAs) have been commonly used for IQBE purposes and their application in the area has been based on combining both criteria in a single scalar fitness function by means of a weighting scheme [25]. However, there is a kind of EAs specially designed for multi-objective problems, *multi-objective evolutionary algorithms* (MOEAs) which are able to obtain non-dominated solutions to the problem [26,27]. This characteristic of MOEAs, applied to IR—specially to the context of the IQBE paradigm, denoted as IQBE MOEA—allows one to derive a number of queries with a different precision–recall trade-off, which improves the aid possibilities to the users in the formulation of their FOLQs.

In the literature there are not many IQBE MOEAs to derive queries in IRSs, and besides, they are based on old-fashioned MOEAs. Three examples are:

- IQBE-MOGA-P [28] based on *multi-objective genetic algorithm* (MOGA) [29] and *genetic algorithm-programming* [30] to learn fuzzy queries with numeric weights.
- IQBE-SPEA [31] based on *strength Pareto evolutionary algorithm* (SPEA) [32] and GP to derive Boolean queries.
- IQBE-SPEA-GA [33] based on SPEA and *Genetic Algorithms* [34] to learn fuzzy multi-weighted queries with ordinal linguistic weights in a multi-granular FOLIRS [14].

Although these approaches have a great performance, there exist other more advanced MOEAs in the specialized literature [26] that have been never used before in the IQBE context. Those MOEAs can improve the performance of the existing IQBE MOEAs in the context of automatic derivation of queries (Boolean queries, numeric weighted queries or ordinal weighted queries) in IRSs.

In this work, an analysis of the performance of two well-known and successful MOEAs applied to the automatic learning of weighted queries, with linguistic weights, in the context of a FOLIRS model is presented. The MOEAs used are: the *non-dominated sorting genetic algorithm* (NSGA-II) [35] and the SPEA2 [36]. Both are adapted to use GP components and optimize both objectives (precision and recall) simultaneously, extending the Kraft et. al.'s IQBE-GP [23] proposal to work with ordinal linguistic weights into the multi-objective context.

The analysis of the proposal will include the use of the Cranfield and CACM collections [1,37], the evaluation of the Pareto fronts with C measure and hypervolume indicator, the analysis of the number of derived queries, and the comparison with the classical IQBE-GP proposal [23].

To do that, this paper is structured as follows. In Section 2 the IQBE paradigm and the FOLIRS model used in this paper are drawn. Section 3 describes the two MOEAs with GP components that are studied. In Section 4 the experimental framework and analysis are presented. Finally, some concluding remarks are pointed out in Section 5.

2. Preliminaries: IQBE and FOLIRS

In this section, we introduce the IQBE paradigm and the foundations of the FOLIRS model used in this paper, including the ordinal fuzzy linguistic approach, their components and procedure of evaluation.

2.1. The IQBE paradigm

The IQBE paradigm was proposed by Chen [22] as "a process in which searchers (users) provide documents (examples) and an algorithm induces (or learns) the key concepts of the examples with the purpose of finding other equally relevant documents". In this way, IQBE can be seen as a technique to assist users in the query building process by using automatic learning methods.

It works taking a set of relevant documents (and optionally non-relevant documents) provided by the user (they can be obtained from a preliminary query or from a browsing process through the documentary database) and applying an automatic learning process to generate a query that describes the user information needs (represented by the previous set of documents). The query that is obtained can be executed in other IRSs to obtain new relevant documents. In this way, it is not necessary for the user to interact with the IR process which is mandatory in other techniques for query refinement as the *relevance feedback* [37].

Several IQBE EAs for different IR models have been proposed and revised in [25]. The most used IQBE models are based on GP components, with queries being represented by expression syntax trees and the algorithms are articulated on the basis of the classic operators: cross, mutation and selection.

2.2. Fuzzy ordinal linguistic IRSs

In the following subsections we briefly present the components of the FOLIRS model used in this work, the ordinal fuzzy linguistic approach, the documentary database, the query subsystem, the matching subsystem and the evaluation model.

2.2.1. The ordinal fuzzy linguistic approach

The *ordinal fuzzy linguistic approach* is a fuzzy approximation technique appropriate to deal with qualitative aspects of problems [13]. It models linguistic information by means of ordinal linguistic labels supported by a *linguistic variable* [19–21]. A linguistic variable is defined by means of a syntactic rule and a semantic rule. In an ordinal fuzzy linguistic approach the syntactic rule is defined by considering a finite and totally ordered label set $\mathcal{S} = \{s_i\}, i \in \{0, \ldots, \mathcal{G}\}$ in the usual sense, i.e., $s_i \geqslant s_j$ if $i \geqslant j$, with odd cardinality (such as 7 or 9 labels), where the mid term represents an assessment of "approximately 0.5", and the rest of the terms being placed symmetrically around it. The semantics of the linguistic term set is established from the ordered structure of the term set by considering that each linguistic term for the pair $(s_i, s_{\mathcal{G}-i})$ is equally informative.

Example 1. A set with nine linguistic labels could be $\mathcal{S} = \{s_0 = Null(N),\ s_1 = Extremely\_Low(EL),\ s_2 = Very\_Low(VL),\ s_3 = Low(L),\ s_4 = Medium(M),\ s_5 = High(H),\ s_6 = Very\_High(VH),\ s_7 = Extremely\_High(EH),\ s_8 = Total(TO)\}$.

In any linguistic approach we need linguistic information management operators, such as

- *Minimization operator*: $MIN(s_a, s_b) = s_a$ *if* $a <= b$,
- *Maximization operator*: $MAX(s_a, s_b) = s_a$ *if* $a >= b$,
- *Negation operator*: $NEG(s_i) = s_j \,|\, j = \mathcal{G} - i$,
- *Aggregation operator*: Usually to combine ordinal linguistic information we can use aggregation operators based on symbolic computation, for example the LOWA operator [38], or the classical t-norm MIN and t-conorm MAX in the linguistic context.

2.2.2. The documentary database

This component stores the documents and the representation of their contents. Textual documents representation is typically based on index terms (that can be either single terms or sequences), which work as content identifiers for the documents. We assume that the database is built like in usual IRSs [1,2]. Therefore, IRS–user interaction is unnecessary because it is built automatically. The database stores a finite set of documents $\mathcal{D} = \{d_1, \ldots, d_m\}$, a finite set of index terms $\mathcal{T} = \{t_1, \ldots, t_l\}$, and the representation R_{d_j} of each document d_j characterized by a numeric indexing function $\mathcal{F} : \mathcal{D} \times \mathcal{T} \rightarrow [0, 1]$ such that $R_{d_j} = \sum_{i=1}^{l} \mathcal{F}(d_j, t_i)/t_i$ is the representation of d_j in fuzzy sets notation. $\mathcal{F}(d_j, t_i) = 0$ implies that the document d_j contents do not deal at all with the concept(s) represented by the index term t_i and $\mathcal{F}(d_j, t_i) = 1$ implies that the document d_j is perfectly represented by the concept(s) indicated by t_i. Using numeric values in (0,1), $\mathcal{F}$ can weight index terms according to their significance in describing the content of a document in order to improve the retrieval of documents.

2.2.3. The query subsystem

It allows users to formulate their information needs (queries) and presents the relevant documents which are retrieved by the system. To do this, each query is expressed as a combination of weighted index terms which are connected by the Boolean operators AND ($\wedge$), OR ($\vee$) and NOT ($\neg$). The query component allows users to weigh, using ordinal linguistic values, each query term according to a threshold semantics. By assigning ordinal linguistic weights in queries, users specify linguistic restrictions on the documents that the FOLIRS has to satisfy in the retrieval activity. By associating threshold weights [4,9,39] with terms in a query, the user is asking to see all the documents that are sufficiently related to the topics represented by such terms. As in [10], we use the linguistic variable *Importance* to express the ordinal linguistic weights associated with the query terms. Thus, we consider a set of ordinal linguistic values $\mathcal{S}$ to express the linguistic weights. Then, we define a linguistic weighted Boolean query as any legitimate Boolean expression whose atomic components (atoms) are tuples $\langle t_i, w_i \rangle$ belonging to the set, $\mathcal{T} \times \mathcal{S}$, $t_i \in \mathcal{T}$ and $w_i \in \mathcal{S}$ being an ordinal value of the linguistic variable *Importance*, modelling a threshold semantics. Accordingly, the set $\mathcal{Q}$ of the legitimate queries is defined by the following syntactic rules:

1. $\forall q = \langle t_i, w_i \rangle \in \mathcal{T} \times \mathcal{S} \rightarrow q \in \mathcal{Q}$.
2. $\forall q, p \in \mathcal{Q} \rightarrow q \wedge p \in \mathcal{Q}$.
3. $\forall q, p \in \mathcal{Q} \rightarrow q \vee p \in \mathcal{Q}$.
4. $\forall q \in \mathcal{Q} \rightarrow \neg q \in \mathcal{Q}$.
5. All legitimate queries $q \in \mathcal{Q}$ are only those obtained by applying rules 1–4, inclusive.

2.2.4. The matching subsystem

It evaluates the degrees (the retrieval status value) to which the document representation satisfies the requirements expressed in the query, and it retrieves the documents that are judged to be relevant. The evaluation subsystem is implemented by the matching or evaluation function $\mathcal{E}$, which assesses the relationship between $\mathcal{Q}$ and $\mathcal{D}$ by means of ordinal linguistic RSVs taken from the linguistic variable *"Relevance"*. Therefore, the goal of $\mathcal{E}$ consists of evaluating documents in terms of their relevance to an ordinal linguistic weighted query. $\mathcal{E}$ is defined by means of a constructive bottom-up evaluation process that acts in two steps:

1. Firstly, the documents are evaluated according to their relevance only to the weighted terms of the query. In this step, a partial relevance degree is assigned to each document with respect to every weighted term in the query.
2. Secondly, the documents are evaluated according to their relevance to the Boolean combination of the weighted terms (their partial relevance degree), and so on, working in a bottom-up fashion until the whole query is processed. In this step, a total ordinal linguistic relevance degree is assigned to each document. It is then used to rank the documents from the most relevant to the less relevant one. The final ordinal linguistic relevance degree is taken from the linguistic variable *Relevance*.

Thus, given any query $q \in \mathcal{Q}$, we define $\mathcal{E} : \mathcal{D} \times \mathcal{Q} \rightarrow \mathcal{S}$ according to the following four evaluation rules:

1. If $q = \langle t_i, w_i \rangle$ then

$$\mathcal{E}(d_j, q) = g(d_j, t_i, w_i).$$

where $g : \mathcal{D} \times \mathcal{T} \times \mathcal{S} \to \mathcal{S}$ is the linguistic matching function to model the threshold semantics defined in the following expression:

$$g(d_j, t_i, w_i) = \begin{cases} s_a & \text{if } s_a \geqslant w_i, \\ s_0 & \text{otherwise,} \end{cases}$$

where $s_a = Label(\mathcal{F}(d_j, t_i))$, $Label : [0, 1] \to \mathcal{S}$ is a function that assigns a label in $\mathcal{S}$ to a numeric value $r \in [0, 1]$ according to the expression: $Label(r) = s_i$ with $i = round(\mathcal{G} \cdot r)$, being $round(\cdot)$ the usual "round" operator.

2. If $q = \langle \wedge_{k=1} q_k \rangle$, $q_k \in \mathcal{Q}$, then

$$\mathcal{E}(d_j, q) = MIN(RSV^1_{d_j}, \ldots, RSV^k_{d_j}),$$

with $RSV^k_{d_j} = \mathcal{E}(d_j, q_k)\ \forall k$.

3. If $q = \langle \vee_{k=1} q_k \rangle$, $q_k \in \mathcal{Q}$, then

$$\mathcal{E}(d_j, q) = MAX(RSV^1_{d_j}, \ldots, RSV^k_{d_j}),$$

with $RSV^k_{d_j} = \mathcal{E}(d_j, q_k)\ \forall k$.

4. If q is negated then

$$\mathcal{E}(d_j, \neg q) = NEG(\mathcal{E}(d_j, q)).$$

When the evaluation subsystem finishes, the FOLIRS presents the retrieved documents arranged in ordinal linguistic relevance classes in decreasing order of $\mathcal{E}$ in such a way that the maximal number of classes is limited by the cardinality of the set of labels chosen for the linguistic variable *Relevance*.

2.2.5. Evaluation 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 user satisfaction [1]. Traditionally, the retrieval effectiveness is based on the document relevance with respect to the users needs. There are different criteria to measure this aspect, but *precision* (P) and *recall* (R) [3] are the most used. *Precision* is the ratio of the relevant documents retrieved by the IRS in response to a query and the total number of documents retrieved, whilst *recall* is the ratio between the number of relevant documents retrieved and the total number of relevant documents for the query that exist in the database [3]. The mathematical expression of each of them is

$$P = \frac{\mathcal{D}_{rr}}{\mathcal{D}_{tr}}; \quad R = \frac{\mathcal{D}_{rr}}{\mathcal{D}_{rt}},$$

where $\mathcal{D}_{rr}$ is the number of relevant documents retrieved, $\mathcal{D}_{tr}$ is the total number of documents retrieved and $\mathcal{D}_{rt}$ is the total number of relevant documents for the query which exist in the database. P and R are defined in [0, 1], being 1 the optimal value.

We notice that the only way to know all the relevant documents existing for a query in the database (value used in the R measure) is to evaluate all documents. Due to this fact and taking into account that relevance is subjective, there are some classic documentary databases (TREC, CACM, Cranfield, ADI, CISI, etc.) available, each one with a set of queries for which the relevance judgments are known, so that they can be used to verify the new proposals in the field of IR [1,37]. In this contribution, we use the Cranfield and CACM collections.

3. Structure of the MOEAs with GP components

For the purpose of this research, two well-known and widely used MOEAs have been selected for performance evaluation: NSGA-II [35] and SPEA2 [36] (that improves the classical SPEA behaviour). A brief overview of these algorithms is given in this section. First, we introduce some basic definitions which are necessary in this research.

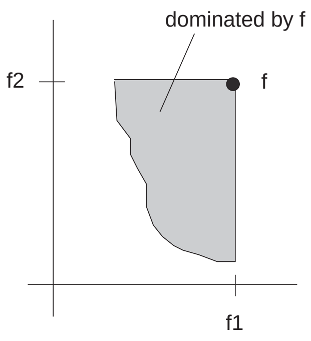

Fig. 1. Concept of dominance.

3.1. Objectives and evaluation of MOEAs

In multi-objective optimization problems, the definition of the *quality* concept is substantially more complex than in single-objective ones, since the processes of optimization imply several different objectives.

The key concepts to evaluate MOEAs are the *dominance relation* and the *Pareto sets*.

The algorithms presented in this paper assume the two classical criteria to evaluate IRSs, i.e., *precision* and *recall* [3] whose expressions are introduced in Section 2.2.5. The studied IQBE MOEA approaches assume that all objectives have to be maximized. The solutions are represented by objective vectors, which are compared according to the dominance relation defined below and displayed in Fig. 1.

Definition 1 (*Dominance relation*). Let $f, g \in \mathbb{R}^m$. Then f is said to dominate g, denoted as $f \succ g$, iff

1. $\forall i \in \{1, \ldots, m\} : f_i \geqslant g_i$,
2. $\exists j \in \{1, \ldots, m\} : f_j > g_j$.

Based on the concept of dominance, the Pareto set can be defined as follows.

Definition 2 (*Pareto set*). Let $F \subseteq \mathbb{R}^m$ be a set of vectors. Then the Pareto set F^* of F is defined as follows: F^* contains all vectors $g \in F$ which are not dominated by any vector $f \in F$, i.e.,

$$F^* := \{g \in F | \nexists f \in F : f \succ g\}.$$

Several quantitative measures based on the Pareto set concept have been proposed in the specialized literature [26,27,40,41]. Two of them are the C measure, and the hypervolume indicator.

Definition 3 (*Coverage of two sets [42]*). Let A, B be two sets of objective vectors. The function C maps the ordered pair (A, B) to the interval [0, 1]:

$$C(A, B) = \frac{|\{a \in A; \exists b \in B : b \succ a\}|}{|A|}$$

measures the ratio of individuals of the Pareto A that are dominated by individuals of the Pareto B. A value of 1 indicates that all individuals of the Pareto A are dominated by individuals of the Pareto B; on the other hand, a value of 0 indicates that none of the individuals of A is dominated by individuals of B. Note that always both directions have to be considered, since $C(A, B)$ is not necessarily equal to $1 - C(B, A)$.

Definition 4 (*Hypervolume [32]*). This indicator measures the hypervolume of that portion of the objective space that is dominated by a Pareto set A, and is to be maximized (see Fig. 2). In order to measure this quantity, the objective space must be bounded. If it is not, a bounding reference point that is dominated by all points should be used, as shown in Fig. 2. In our case, the reference point is (0, 0).

We will use both C and hypervolume in this study.

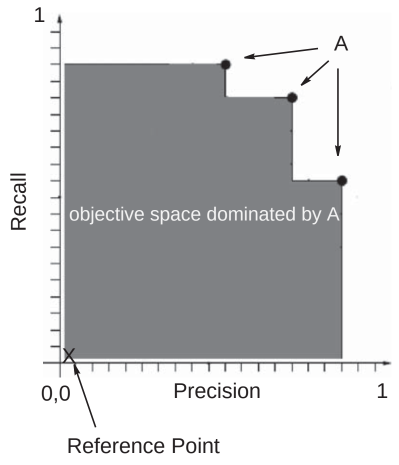

Fig. 2. The hypervolume indicator.

3.2. NSGA-II-GP

NSGA-II [35] is one of the most well-known and frequently used MOEAs for general multi-objective optimization in the literature. As in other EAs, first NSGA-II generates an initial population. Then an offspring population is generated from the current population by selection, crossover and mutation. The next population is constructed from the current and offspring populations. The generation of an offspring population and the construction of the next population are iterated until a stopping condition is satisfied. NSGA-II algorithm has two features, which make it a high-performance MOEA. One is the fitness evaluation of each solution based on Pareto ranking and a crowding measure, and the other is an elitist generation update procedure.

Each solution in the current population is evaluated in the following manner. First, Rank 1 is assigned to all non-dominated solutions in the current population. All solutions with Rank 1 are tentatively removed from the current population. Next, Rank 2 is assigned to all non-dominated solutions in the reduced current population. All solutions with Rank 2 are tentatively removed from the reduced current population. This procedure is iterated until all solutions are tentatively removed from the current population (i.e., until ranks are assigned to all solutions). As a result, a different rank is assigned to each solution. Solutions with smaller ranks are viewed as being better than those with larger ranks. Among solutions with the same rank, an additional criterion called a crowding measure is taken into account. The crowding measure for a solution calculates the distance between its adjacent solutions with the same rank in the objective space. Less crowded solutions with larger values of the crowding measure are viewed as being better than more crowded solutions with smaller values of the crowding measure.

A pair of parent solutions are selected from the current population by binary tournament selection based on the Pareto ranking and the crowding measure. When the next population is to be constructed, the current and offspring populations are combined into a merged population. Each solution in the merged population is evaluated in the same manner as in the selection phase of parent solutions using the Pareto ranking and the crowding measure. The next population is constructed by choosing a specified number (i.e., population size) of the best solutions from the merged population. Elitism is implemented in NSGA-II algorithm in this manner.

Considering the components previously defined and the descriptions of the authors in [35], NSGA-II consists of the next steps:

1. A combined population R_t is formed with the initial parent population P_t and offspring population Q_t (initially empty).
2. Generate all non-dominated fronts $F = (F_1, \ldots, F_m)$ of R_t.
3. Initialize $P_{t+1} = \emptyset$ and $i = 1$.
4. Repeat until the parent population is filled.
5. Calculate crowding-distance in F_i.

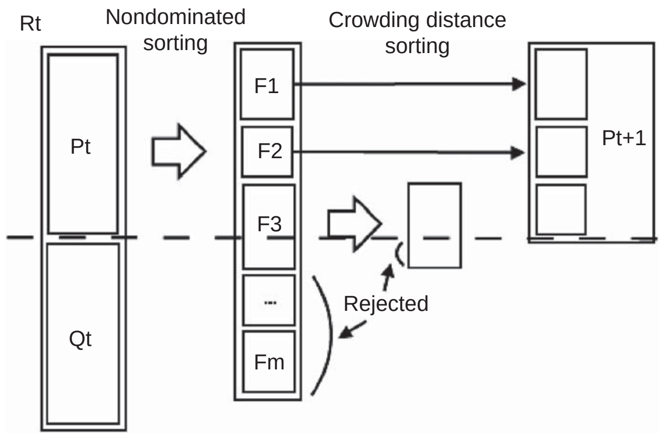

Fig. 3. NSGA-II procedure.

6. Include i-th non-dominated front in the parent population.
7. Check the next front for inclusion.
8. Sort in descending order using crowded-comparison operator.
9. Choose the first $(N - |P_{t+1}|)$ elements of F_i.
10. Use selection, crossover and mutation to create a new population Q_{t+1}.
11. Increment the generation counter.

The general NSGA-II procedure is drawn in Fig. 3.

In this paper NSGA-II will be adapted to use GP components. It will be denoted as IQBE-NSGA-II-GP.

3.3. SPEA2-GP

SPEA2 [36] was designed to overcome the problems of its predecessor for general multi-objective optimization, SPEA algorithm [32]. In contrast with SPEA, SPEA2: (1) incorporates a fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that it dominates and the number of individuals by which it is dominated; (2) uses the nearest neighbour density estimation technique which guides the search more efficiently; (3) has an enhanced archive truncation method which guarantees the preservation of boundary solutions. Next, we briefly describe the complete SPEA2 algorithm.

SPEA2 uses a fixed population and archive size. The population forms the current base of possible solutions, while the archive contains the current solutions. The archive is constructed and updated by copying all non-dominated individuals in both archive and population into a temporary archive. If the size of this temporary archive differs from the desired archive size, individuals are either removed or added as necessary. Individuals are added by selecting the best dominated individuals, while the removal process uses a heuristic clustering routine in the objective space. The motivation for this is that one would like to try to ensure that the archive contents represent distinct parts of the objective space. Finally, when selecting individuals for participating in the next generation all candidates are selected from the archive using a binary tournament selection scheme. Considering the components defined and the descriptions of the authors in [36], SPEA2 algorithm consists of the next steps:

Input:

- N (population size),
- $\overline{N}$ (external population size),
- T (maximum number of generations).

Output: A (non-dominated set).

1. Generate an initial population P_0 and create the empty external population $\overline{P_0} = \emptyset$.
2. Calculate fitness values of individuals in P_t and $\overline{P_t}$.

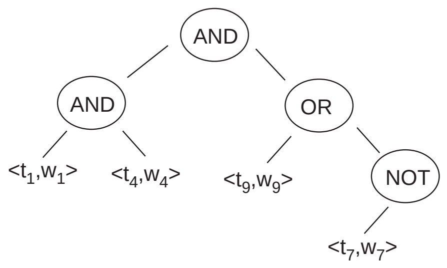

Fig. 4. An example of linguistic weighted Boolean query.

3. Copy all non-dominated individuals in $P_t \bigcup \overline{P}_t$ to $\overline{P}_{t+1}$. If $|\overline{P}_{t+1}| > \overline{N}$ apply truncation operator. If $|\overline{P}_{t+1}| < \overline{N}$ fill with dominated in $P_t \bigcup \overline{P_t}$.
4. If $t \geqslant T$, return A and stop.
5. Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill the mating pool.
6. Apply crossover and mutation operators to the mating pool and set P_{t+1} to the resulting population. Go to step 2 with $t = t + 1$.

In this paper SPEA2 will be adapted to use GP components. It will be denoted as IQBE-SPEA2-GP.

3.4. GP components for MOEAs

The two MOEAs-GP studied in this paper share the following components:

Codification scheme: Extended Boolean queries are encoded in expression syntax trees, whose terminal nodes are atoms (terms and linguistic weights $\langle t_i, w_i \rangle$) as defined in Section 2.2.3, and whose inner nodes are the Boolean operators AND, OR and NOT. Besides, each terminal node has an ordinal linguistic value associated, corresponding to the threshold semantics. Fig. 4 shows a graphical example of this kind of queries. Hence, the natural representation is to encode the query within a tree and to work with a GP algorithm [24] to evolve it, as done by previous approaches devoted to the derivation of Boolean queries [31,43] or extended Boolean queries (fuzzy queries with numerical weights) [23,44,45].

Crossover operator: Subtrees are randomly selected and crossed over in two randomly selected queries, as drawn in Fig. 5.

Mutation operator: A randomly selected term, weight or operator is changed in a randomly selected tree. An example is shown in Fig. 6.

Initial population: All individuals of the first generation are generated in a random way. The population is created including all the terms in the relevant documents provided by the user. Those that appear in more relevant documents will have greater probability of being selected.

4. Experimental study

This section describes the used experimental framework and analysis.

4.1. Experimental framework

The experimental study has been developed using the Cranfield and CACM collections [1,37]:

- Crandfield is composed of 1398 documents about Aeronautics.
- CACM contains 3204 document published in the journal *Communications of the ACM* between 1958 and 1979.

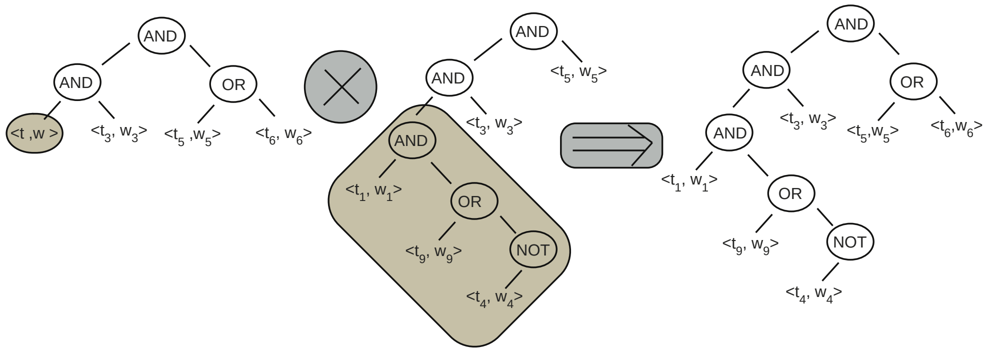

Fig. 5. An example of crossover on two queries.

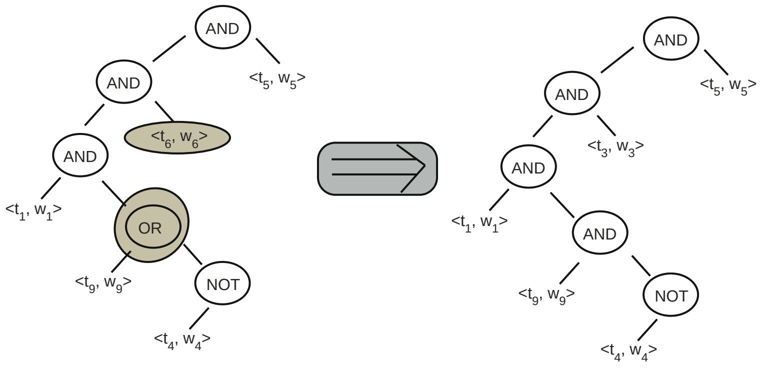

Fig. 6. An example of mutation.

In both collections, the textual documents have been automatically indexed in the usual way[1] by first extracting the non-stop words and performing a stemming process, thus obtaining a total number of 3857 and 7562 different indexing terms, respectively. Then a $tf \cdot idf$ scheme [37] has been used to generate the term weights in the document representations. Both collections have a number of test queries associated (225 in the Cranfield collection and 64 in the CACM collection). In our problem, each test 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 test query set, we have selected a representative sample that allows to study the behaviour of the studied IQBE MOEA-GP approaches.

The role of the user who provides documents will be played by the test queries associated with the considered collection, and more exactly, by the relevance judgements associated with them. In this way, for example, if there are 29 relevant documents for test query #1 in the Cranfield collection, this test query will mimic a situation in which the user provides 29 documents related to his information needs. Besides, the remaining 1369 documents (1398–29) will be considered as non-relevant documents for the IQBE process.

The studied IQBE MOEA-GP approaches generate a set of queries from the relevant and the non-relevant documents sets. To do so, it is necessary to consider sufficiently representative number of positive examples (relevant documents), so test queries with more relevant documents associated have been selected:

- Among the 225 test queries associated with the Cranfield collection, those test queries presenting 20 or more relevant documents have been taken into account. The seven resulting test queries (number #1, #2, #23, #73, #157, #220 and #225) have 29, 25, 33, 21, 40, 20 and 25 relevant documents associated, respectively.

[1] To do so, we use the classical Salton's SMART IRS [46].

Table 1
Common parameters.

Parameter	Value
Cross probability	0.8
Mutation probability	0.2
Nodes per query	19 maximum
Population size (M)	800
Elite population	200
Other parameters	Term set $\mathcal{S}$ with cardinality $\mathcal{G}+1=9$

Table 2
Average results of the C measure for each test query and the two studied IQBE MOEAs-GP on Cranfield and CACM collections.

Cranfield			CACM		
Query	$C(X,Y)$	$C(Y,X)$	Query	$C(X,Y)$	$C(Y,X)$
#1	**0.082**	0.976	#10	**0.148**	1.000
#2	**0.111**	0.974	#14	**0.352**	1.000
#23	**0.047**	0.978	#25	**0.113**	0.975
#73	**0.228**	0.959	#26	**0.163**	1.000
#157	**0.028**	0.972	#43	**0.232**	0.996
#220	**0.211**	0.959	#58	**0.322**	1.000
#225	**0.174**	0.943	#59	**0.143**	1.000
			#61	**0.175**	0.967

X = IQBE-NSGA-II-GP, Y = IQBE-SPEA2-GP.

- On the other hand, CACM provides 64 test queries, of which those that have 30 or more relevant documents have been selected. The eight resulting test queries (number #10, #14, #25, #26, #43, #58, #59, #61) have 35, 44, 51, 30, 41, 30, 43 and 31 relevant documents associated, respectively.

The studied IQBE MOEAs-GP in this contribution have been run 30 times for each test query and collection (a total of 900 runs) with different initializations for each selected test query. The number of chromosome evaluations is 50,000 per run. We use a 3.4 GHz Pentium IV computer with 1 Gb of RAM. The common parameter values considered are presented in Table 1.

4.2. Results and analysis

From each run a Pareto set is obtained and a later filtration process is performed in the decision space in order to remove those queries that are identical to another. The two filtered Pareto sets obtained by each run and test query are compared with the performance C measure and the hypervolume indicator.

In Table 2 we present the average results for the 30 runs of the C measure for each pair of IQBE MOEAs-GP and test query for the Cranfield and CACM collections, where values in bold denote the best results.

In Table 3 we present the average results for the on 30 runs of the hypervolume indicator for each pair of IQBE MOEAs-GP and test query for the Cranfield and CACM collections, where values in bold denote the best results. Note that *HV(X)* denotes the hypervolume indicator of the Pareto set achieved by the IQBE MOEA-GP *X*.

The experimental results show that IQBE-NSGA-II-GP, is the IQBE MOEA-GP approach that achieves a better performance, i.e., it achieves better non-dominated solutions sets (view values in bold in Tables 2 and 3) in the process of learning FOLQs than IQBE-SPEA2-GP.

In Table 4 we present the average number of different queries in the decision space for the 30 runs achieved by each IQBE MOEAs-GP for each test query for the Cranfield and CACM collections, where values in bold denote the best results. These results show that IQBE-NSGA-II-GP, is the IQBE MOEA-GP approach that achieves more different queries, in both decision and objective space, for a unique run.

Table 3
Average results of the hypervolume indicator for each test query and the two studied IQBE MOEAs-GP on Cranfield and CACM collections.

Cranfield			CACM		
Query	$HV(X)$	$HV(Y)$	Query	$H(X)$	$H(Y)$
#1	**0.425**	0.356	#10	**0.321**	0.273
#2	**0.471**	0.390	#14	**0.267**	0.216
#23	**0.379**	0.308	#25	**0.245**	0.204
#73	**0.467**	0.376	#26	**0.372**	0.317
#157	**0.331**	0.272	#43	**0.285**	0.239
#220	**0.480**	0.392	#58	**0.345**	0.293
#225	**0.431**	0.359	#59	**0.274**	0.229
			#61	**0.364**	0.309

X = IQBE-NSGA-II-GP, Y = IQBE-SPEA2-GP.

Table 4
Average number of different queries for each test query and the two studied IQBE MOEAs-GP on Cranfield and CACM collections.

Cranfield			CACM		
Query	X	Y	Query	X	Y
#1	**232.77**	11.70	#10	**340.93**	12.50
#2	**333.67**	10.87	#14	**500.10**	12.27
#23	**307.43**	9.43	#25	**478.33**	14.40
#73	**456.50**	9.20	#26	**323.23**	11.53
#157	**279.97**	10.77	#43	**417.23**	13.80
#220	**390.53**	10.17	#58	**284.73**	10.73
#225	**312.53**	9.90	#59	**355.90**	14.00
			#61	**320.70**	12.37

X = IQBE-NSGA-II-GP, Y = IQBE-SPEA2-GP.

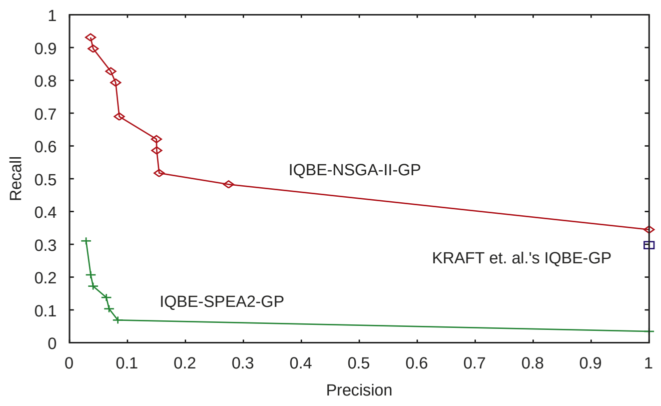

Fig. 7. Pareto sets achieved by IQBE-NSGA-II-GP, IQBE-SPEA2-GP and Kraft et. al.'s IQBE-GP for the test query #1 on Cranfield collection.

In Fig. 7 we present the queries achieved[2] in a unique run, by IQBE-NSGA-II-GP, IQBE-SPEA2-GP and the Kraft et. al.'s IQBE-GP proposal, adapted to use ordinal linguistic weights in the objective space. In it, we can see as IQBE-NSGA-II-GP gets a good set of solutions, with a good precision–recall trade-off. It also improves the performance of the Kraft et. al.'s IQBE-GP proposal.

[2] The results correspond to the test query #1 of Cranfield collection.

5. Conclusions

In this contribution, an analysis of performance, in the FOLIRSs context, of two of the most currently used MOEAs in the specialized literature has been done. The studied MOEAs have been applied on the automatic learning of FOLQs adapted to use GP components. All of them extend the Kraft et al.'s IQBE-GP proposal [23] to work with ordinal linguistic weights in the multi-objective context.

The experimental results show that NSGA-II, with GP (IQBE-NSGA-II-GP) achieves better results than IQBE-SPEA2-GP.

References

[1] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, Reading, MA, 1999.
[2] G. Salton, M. McGill, An Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.
[3] C. van Rijsbergen, Information Retrieval, Butterworths, London, 1979.
[4] D. Buell, D. Kraft, A model for a weighted retrieval system, Journal of the American Society for Information Science 32 (1981) 211–216.
[5] S. Miyamoto, Fuzzy Sets in Information Retrieval and Cluster Analysis, Kluwer Academic Publishers, Dordrecht, 1990.
[6] V. Tahani, A fuzzy model of document retrieval systems, Information Processing & Management 12 (1976) 177–187.
[7] A. Bookstein, Fuzzy request: an approach to weighted Boolean searches, Journal of the American Society for Information Science and Technology 31 (4) (1980) 240–247.
[8] G. Bordogna, P. Carrara, G. Pasi, Query term weights as constraints in fuzzy information retrieval, Information Processing & Management 27 (1) (1991) 15–26.
[9] D. Buell, D. Kraft, Threshold values and Boolean retrieval systems, Information Processing & Management 17 (1981) 127–136.
[10] G. Bordogna, G. Pasi, A fuzzy linguistic approach generalizing Boolean information retrieval: a model and its evaluation, Journal of the American Society for Information Science 44 (2) (1993) 70–82.
[11] G. Bordogna, G. Pasi, An ordinal information retrieval model, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (2001) 63–76.
[12] E. Herrera-Viedma, An information retrieval system with ordinal linguistic weighted queries based on two weighting elements, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (2001) 77–88.
[13] E. Herrera-Viedma, Modelling 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) (2001) 460–475.
[14] E. Herrera-Viedma, O. Cordón, M. Luque, A. López, A. Muñoz, A model of fuzzy linguistic IRS based on multi-granular linguistic information, International Journal of Approximate Reasoning 34 (2003) 221–239.
[15] E. Herrera-Viedma, A. López-Herrera, A model of information retrieval system with unbalanced fuzzy linguistic information, International Journal of Intelligent Systems 22 (11) (2007) 1197–1214.
[16] E. Herrera-Viedma, A. López-Herrera, M. Luque, C. Porcel, A fuzzy linguistic IRS model based on a 2-tuple fuzzy linguistic approach, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 15 (2) (2007) 225–250.
[17] E. Herrera-Viedma, A. López-Herrera, C. Porcel, Tuning the matching function for a threshold weighting semantics in a linguistic information retrieval system, International Journal of Intelligent Systems 20 (2005) 921–937.
[18] D. Kraft, G. Bordogna, G. Pasi, An extended fuzzy linguistic approach to generalize Boolean information retrieval, Information Sciences 2 (1994) 119–134.
[19] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part I, Information Sciences 8 (1975) 199–249.
[20] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part II, Information Sciences 8 (1975) 301–357.
[21] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part III, Information Sciences 9 (1975) 43–80.
[22] H. Chen, G. Shankaranarayanan, L. She, A. Iyer, A machine learning approach to inductive query by example: an experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing, Journal of the American Society for Information Science 49 (8) (1998) 693–705.
[23] D. Kraft, F. Petry, B. Buckles, T. Sadasivan, Genetic algorithms and fuzzy logic systems, in: E. Sanchez, T. Shibata, L.A. Zadeh (Eds.), Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback, World Scientific, Singapore, 1997, pp. 155–173.
[24] J. Koza, Genetic Programming. On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, 1992.
[25] O. Cordón, E. Herrera-Viedma, C. López-Pujalte, M. Luque, C. Zarco, A review on the application of evolutionary computation to information retrieval, International Journal of Approximate Reasoning 34 (2003) 241–264.
[26] C. Coello, G. Lamont, D. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd ed., Springer, Berlin, 2007.
[27] K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Wiley, New York, 2001.
[28] O. Cordón, F. Moya, C. Zarco, Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments, in: Proc. IEEE Internat. Conf. Fuzzy Systems (FUZZ-IEEE 2004), Budapest, Hungary, 2004, pp. 571–576.
[29] C. Fonseca, P. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in: Proc. 5th Internat. Genetic Algorithms (ICGA'93), San Mateo, CA, 1993, pp. 416–423.
[30] L. Howard, D. D'Angelo, The GA-P: a genetic algorithm and genetic programming hybrid, IEEE Expert (1995) 11–15.
[31] O. Cordón, E. Herrera-Viedma, M. Luque, Improving the learning of Boolean queries by means a multiobjective IQBE evolutionary algorithm, Information Processing & Management 42 (3) (2006) 615–632.
[32] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.

[33] O. Cordón, E. Herrera-Viedma, M. Luque, Multiobjective genetic algorithm for linguistic persistent query learning in text retrieval, in: Y. Jin (Ed.), Multi-objective Machine Learning, Studies in Computational Intelligence Series, Springer, Berlin, 2006, pp. 601–627.
[34] Z. Michalewicz, Genetic Algorityms + Data Structures = Evolution Programs, Springer, New York, 1996.
[35] K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[36] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, T. Fogarty (Eds.), Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95–100.
[37] G. Salton, Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley, Reading, MA, 1989.
[38] F. Herrera, E. Herrera-Viedma, J. Verdegay, Direct approach processes in group decision making using linguistic OWA operators, Fuzzy Sets and Systems 79 (1996) 175–190.
[39] D. Kraft, D. Buell, Fuzzy sets and generalized Boolean retrieval systems, International Journal of Man-Machine Studies 19 (1983) 45–56.
[40] J. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, revised version, February 2006.
[41] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2) (2000) 173–271.
[42] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms a comparative case study, in: A.E. Eiben, T. Bäck, M. Schoenauer, H.P. Schwefel (Eds.) Proc. 5th Internat. Conf. Parallel Problem Solving from Nature (PPSN-V), Springer, Berlin, Germany, 1998, pp. 292–301.
[43] 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.
[44] O. Cordón, F. Moya, C. Zarco, A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system, Mathware and Soft Computing 7 (2–3) (2000) 309–322.
[45] 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 (2002) 308–319.
[46] G. Salton, The SMART Retrieval System, Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1997.