Multiple query evaluation based on an enchanced genetic algorithm

Lynda Tamine-Lechani, Claude Chrisment, Mohand Boughanem

Information Processing & Management pp 215-231, 39(2) 2003

Multiple query evaluation based on an enhanced genetic algorithm

Lynda Tamine [1] , Claude Chrisment [2] , Mohand Boughanem [2]

[1] ISYCOM/ GRIMM Université de Toulouse II, 5 Allées A. Machado, 31058 Toulouse Cedex, France

[2] IRIT SIG Université de Toulouse III, 118 Route de Narbonne, 31062 Toulouse, France

tamine@univ-tlse2.fr, chrisme@irit.fr, bougha@irit.fr

Information Processing & Management pp 215-231, 39(2) 2003

Abstract

Recent studies suggest that significant improvement in information retrieval performance can be achieved by combining multiple representations of an information need. The paper presents a genetic approach that combines the results from multiple query evaluations. The genetic algorithm aims to optimise the overall relevance estimate by exploring different directions of the document space. We investigate ways to improve the effectiveness of the genetic exploration by combining appropriate techniques and heuristics known in genetic theory or in the IR field. Indeed, the approach uses a niching technique to solve the relevance multimodality problem, a relevance feedback technique to perform genetic transformations on query formulations and evolution heuristics in order to improve the convergence conditions of the genetic process.The effectiveness of the global approach is demonstrated by comparing the retrieval results obtained by both genetic multiple query evaluation and classical single query evaluation performed on a subset of TREC-4 using the Mercure IRS. Moreover, experimental results show the positive effect of the various techniques integrated to our genetic algorithm model.

Keywords: Information retrieval; Genetic algorithm; Relevance feedback

1. Introduction

Relevance estimation is critical in the process of attempting to satisfy the user query and is the main goal of modelling in IR. In traditional IR, a single retrieval algorithm is used in order to perform this task. However, it has been observed that retrieval effectiveness can often be improved significantly by using a number of different retrieval algorithms (Katzer & al, 1992) (McGill & al, 1979) (Lee, 1997). This is because different retrieval algorithms emphasize different document and query features when measuring the relevance and therefore retrieve different sets of documents. Since different algorithms can retrieve documents with different representations, the overall performance of the combined algorithm may be higher.

In this paper, we present a genetic method where different query formulation results are combined in order to estimate the overall relevance score. Genetic algorithms constitute an interesting category of modern heuristic search. Based on the powerful principle of *survival of the fittest*, genetic algorithms model the natural phenomena of genetic inheritance and Darwinian strife of survive. Genetic algorithms have been shown to be a powerful search mechanism and seem to be suitable in IR for the main following reasons (Tamine & Boughanem, 2000) :

- The document space represents a high dimensional space. As GAs have been shown to be powerful search mechanisms due to their robust nature and quick search capabilities, they seem to be suitable for information retrieval. Due to their inherent properties of implicit parallelism, a GA could perform the search in different regions of the document space simultaneously.

- Contrary to the classical retrieval models, the GA manipulates a population of queries rather than a single query. Each query may retrieve a subset of relevant documents that can be merged. We believe that this is more efficient than using a hill-climbing search based on a single query.

- The classical methods of query expansion manipulate each term independently of each other. Several experiments have however already shown that the terms occur in the documents by groups. The GA would contribute in this case to preserve useful information links representing a set of terms indexing the relevant documents.

- The classical methods of relevance feedback are not efficient when no relevant documents are retrieved with the initial query. In contrast, the probabilistic exploration induced by the GA allows the exploration of new zones in the document space independently from the initial query.

In the context of our study, a GA seems to be suitable for optimising the relevance score value by combining the evaluation results of a population of different query formulations. Each query selects a subset of relevant documents and we show that this is more efficient than using a hill-climbing search based on a single query. Furthermore, the GA proposed is improved using the following heuristics.

1- Niching technique

Despite no formal description, we believe that the relevance function is multimodal in the sense that relevant documents corresponding to the same user query may be located in different regions of the document space and therefore have some descriptors that are different. According to this assumption, we use the niching ecological technique (Chen, 1995) (Goldberg, 1994) in order to explore the document space by encouraging the reproduction of queries in different directions rather than reaching a unique optimal query when using a classical genetic exploration.

2- Application of enhanced operators

Relevance feedback is an effective technique commonly used in IR (Harman, 1992) (Kwok, 1995) (Robertson & al, 1995) (Boughanem & Soule-Dupuy, 1997). Rather than using blind genetic operators, we propose enhanced ones which aim to expand and reweight individual queries using the user's judgements.

3-Virtual individuals

In order to improve the convergence conditions of the GA, we propose the application of evolution heuristics. These are advanced genetic techniques which aim to integrate other individuals built on good retrieval criteria with the *natural* GA population.

The remainder of this paper is organised as follows. In section 2 we present an overview of multiple query evaluation techniques. Section 3 outlines our genetic retrieval algorithm. Section 4 presents experiments performed on documents issued from TREC followed by a discussion of the results. Section 5 concludes the paper.

2. Multiple query evaluation

2.1. General overview

The idea of combining multiple representations of either queries or texts, or of using different retrieval techniques in order to improve the retrieval performance has been suggested and

discussed with the name of "data fusion". Several works in this general area give theoretical rationales for combination techniques. The most common stems from the observation that different representations of the same query retrieve different documents (both relevant and non relevant) (see TREC track query, Buckley & Walz (1999)). This may be due to the fact that the process of representation is so uncertain that any one representation captures only a part of the user's need. Thus, the combination of multiple representations will address different aspects of the user need and then retrieve more relevant documents.

Robertson (1977) also give an interesting analysis which suggests that each representation of a query is a source of evidence and can be used to improve prediction of probability of relevance.
McGill & Koll (1979) and Katzer & al (1982) find that different query formulations generated different documents. However, they noticed that there was a small overlap in the documents retrieved.
Turtle & Croft (1991) propose an inference network-based retrieval model which combines different document representations and various query formulations in a probabilistic framework. They demonstrate that combining the retrieval results of natural language and Boolean query formulations improves the effectiveness of IR. Belkin & al (1993) investigate the effect of progressively cumulating the evidence of various independently generated query representations of one type in a probabilistic-inference network retrieval system. Experiments carried out on a TREC collection show that an appropriate combination of different Boolean query formulations has a positive effect upon retrieval performance. Lee & al (1997) analyse the research results obtained in the data fusion theory literature and suggested a new rationale for evidence combination of different runs. They investigate different combining methods and show that using rank order of the documents retrieved gives better retrieval effectiveness than using similarity if the runs in the combination generate different rank-similarity curves.

Finally, it is concretely shown in the IR literature that significant improvements can be achieved by combining various sources of evidence. However open questions still remain. The most difficult is how to define the optimal strategy of combination. Belkin & al (1993) give an interesting analysis that is quoted below: " *All of the suggestions require some means of estimating the performance of a source of formulation in advance of its use, an inherently difficult problem which we have been enable to address in our current study* "
In this paper we address this question using a genetic approach. We suggest the combination of retrieval results of the best query formulations generated according to a relevance prediction function value and genetic enhanced operators. The whole process of query evaluation is based on both general genetic optimisation methodology and relevance feedback technique.

2.2. Multiple query evaluation based on genetic combination

Genetic techniques combining retrieval results have been proposed by several authors. Gordon (1988) adapt a GA to derive better descriptions of documents. Each document is assigned N descriptions represented by a set of indexing terms. Genetic operators and relevance judgement are applied to the descriptions in order to build the best document descriptions. The author show that the GA produces better document descriptions than the ones generated by the probabilistic model. Redescription improved the relative density of co-relevant documents by 39.74% after twenty generations and 56.61% after forty generations. Yang & Korfhage (1993) propose a GA for query optimisation by reweighing the query term indexing without query expansion. They

used a selection operator based on a stochastic sample, a blind crossover at two crossing points, and a classical mutation to renew the population of queries.
The experiments show that the queries converge to their relevant documents after six generations.
Kraft & al (1995) apply GA programming in order to improve tweighted boolean query formulations. Their first experiments show that GA programming is a viable method for deriving good queries. Horng & Yeh (2000) propose a novel approach to automatically retrieve keywords and then use genetic techniques to tune the keyword weights. The effectiveness of the approach is demonstrated by comparing the results obtained to those using a PAT-tree based approach.

In comparison with these studies, our genetic approach has two main advantages:

1- The genetic retrieval process deals with any retrieval model.
2- The GA takes advantage of domain techniques by using appropriate heuristics in order to improve the retrieval task.

3. Our Model: Multiple query evaluation based on an enhanced genetic algorithm

3.1. The genetic relevance optimisation process

The retrieval process, as shown in figure 1, is based on an iterative feedback evaluation of query niches. First, the initial query is submitted to the IRS and then the population of queries is organised into subpopulations, called niches, according to their evaluation results (retrieved documents). At each generation, the evaluation of each subpopulation of queries produces a list of documents. The merging process then produces the outcome list.
At this level, genetic techniques are applied in order to transform the query formulations in the direction of the fittest ones. The process is repeated for a fixed number of feedback iterations.

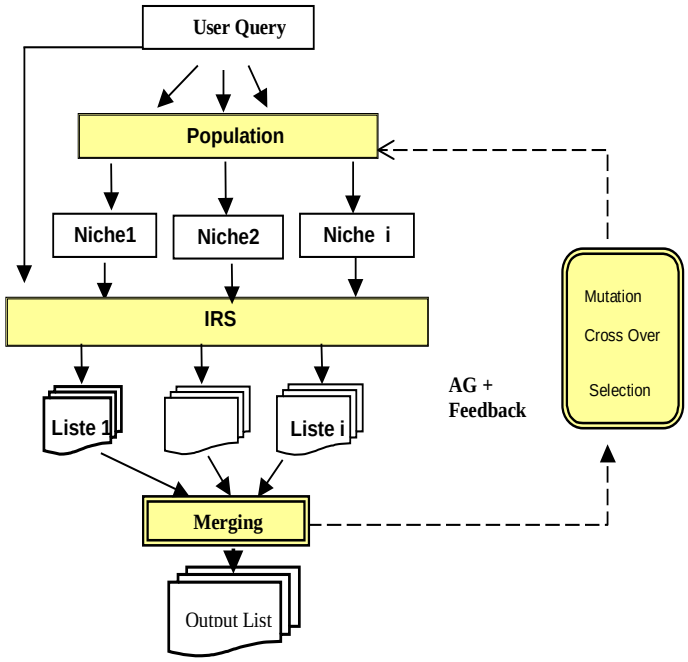

Figure 1 : The genetic retrieval process

3.2. The GA model

Our GA handles the process of relevance optimisation; thus it aims to reach optimal or near optimal query formulations which produce the best outcomes, according to the user query. Indeed, we aim to increase the overall level of recall that is reachable.

3.2.1 Individual Encoding

In our approach, the genetic individual is a query formulation. Each gene corresponds to an indexing term or concept. Its value or *locus* is represented by a real value and defines the effectiveness of the term in the considered formulation. Each individual representing a query is of the form $Q_u\ (q_{u1}, q_{u2},...,q_{uT})$

T : Total number of stemmed terms automatically extracted from the documents
q_{ui} : weight of the term i in Q_u

Initially, a term weight can be computed by any query term weight scheme; it will then evolve through the generations. In our case, we used the following formula :

$$q_{ij}=\begin{cases}\dfrac{nq*qtf}{nq-qtf} & if\ \ (nq>qtf)\\ qtf & otherwise\end{cases}$$

qtf : query term frequency
nq : number of query terms

Note that the above encoding is quite flexible with respect to the length of the queries. Although each chromosome has a fixed length, the genes are interpreted (based on the value of the weight q_{ui} equal or different to 0) in such way that the individual phenotype (the query) has a variable length.

3.2.2. Population

The population is organised into several niches. A niche is a set of individual queries exploring a potential region of the document space. The theory of genetic niching technique (Goldberg, 1994) shows that the exploration process discovers relevant regions using different directions, that is we name *parallel and cooperative query search.* The niches structure are defined according to the *coniche* operator $\equiv_N$ defined as follows (Tamine, 2000):

$$(Q_u^{(s)} \equiv_N Q_v^{(s)}) \Leftrightarrow (\left|(Ds(Q_u^{(s)},L)) \cap (Ds(Q_v^{(s)},L)\right|) > Coniche\_Limit)$$

$Q_u^{(s)}$: individual query u at generation (s) of the GA
$Ds(Q_u^{(s)},L)$: the L top documents retrieved by $Q_u^{(s)}$
Coniche _ Limit : the min number of common documents retrieved by queries of the same niche

Thus, the queries belonging to the same niche select common documents. The size and the structure of the niches evolve at each generation due to both the retrieval process and genetic transformations.

3.2.3. Fitness function

The fitness function measures the effectiveness of a query to retrieve relevant documents at the top. It is computed using a formula built on the Guttman model:

$$Fitness(Q_u^{(s)}) = 1 + \frac{\sum_{dr \in Dr(s), dnr \in Dnr(s)} J(Q_u^{(s)}, dr) - J(Q_u^{(s)}, dnr)}{\sum_{dr \in Dr(s), dnr \in Dnr(s)} \left| J(Q_u^{(s)}, dr) - J(Q_u^{(s)}, dnr) \right|}$$

J : Jaccard measure defined as follows:

$$J(D_j, Q_u^{(s)}) = \frac{\sum_{i=1}^{T} q_{ui}^{(s)} d_{ji}}{\sum_{i=1}^{T} q_{ui}^{2} + \sum_{i=1}^{T} d_{ji}^{2} - \sum_{i=1}^{T} q_{ui}^{(s)}}$$

$Dr^{(s)}$: set of relevant documents retrieved at the generation(s) of the GA
$Dnr^{(s)}$: set of non relevant documents retrieved at the generation(s) of the GA
dr: relevant document
dnr : irrelevant document

The most favorable feature of the Guttman model function is that it is highly correlated with the standard goodness measure in information retrieval that is average precision (Bartell & al, 1998).

3.2.4. Genetic operators

The genetic operators defined in our approach are not classical ones as they are not based on the basic structure proposed in GA theory (Goldberg, 1994). They have been adopted to take advantage of techniques developed in IR. Thus, we qualify them as knowledge based operators. In addition they are restrictively applied to the niches in order to focus the search in the corresponding directions of the document space.

- Selection

The selection procedure is based on a variant of the usual roulette wheel selection (Goldberg, 1994). It consists essentially of assigning to every individual of the population a number of copies in the next generation, proportional to its relative fitness.

- Crossover

The crossover operator is applied to a pair of individuals that are selected in the same niche, according to the crossover probability *Pc* . We define a crossover based on term weight, with no crossing point. It allows the modifying of term weights according to their distribution in the relevant and in the non-relevant documents. Let us consider
$Q_u^{(s)}$ and $Q_v^{(s)}$ two individuals selected for crossover. The result is the new individual $Q_p^{(s)}$ defined as:

$$\{ Q_u^{(s)} (q_{u1}^{(s)}, q_{u2}^{(s)}, \ldots , q_{uT}^{(s)}),\ Q_v^{(s)} (q_{v1}^{(s)}, q_{v2}^{(s)}, \ldots , q_{vT}^{(s)}) \} \Rightarrow \{ Q_p^{(s+1)} (q_{p1}^{(s+1)}, q_{p2}^{(s+1)}, \ldots , q_{pT}^{(s+1)}) \}$$

We defined :

$$q_{pi}^{(s+1)} = Max(q_{ui}^{(s)}, q_{vi}^{(s)}) \text{ if } weight(t_i, Dr^{(s)}) \geq weight(t_i, Dnr^{(s)})$$

$$Min(q_{ui}^{(s)}, q_{vi}^{(s)}) \text{ otherwise}$$

$$weight(t_i, D) = \sum_{dj \in D} d_{ji} \ ,$$

d_{ji} : term weight of t_i in d_j
D : a set of documents

In other words, if the weight of term t_i in the set of relevant documents is higher than its weight in the set of non-relevant documents, this term is retained as significant and the highest weight among ($q_{ui}^{(s)}$, $q_{vi}^{(s)}$) is assigned to this term in the new query $Q_p^{(s+1)}$. Otherwise, the lowest weight is assigned to it in the new query.

- Mutation

This consists essentially of exploring the terms occurring in the relevant documents in order to expand and/or reweight the query selected for the mutation. Let us consider $Q_u^{(s)}$ as the selected individual query and $Lmut^{(s)}$ as the set of terms from $Dr^{(s)}$ the relevant documents retrieved at the last generation of the GA. The mutation will alter genes of the selected individual on the basis of the $Lmut^{(s)}$ terms and on the probability *Pm* . The $Lmut^{(s)}$ terms are sorted according to a score value calculated as follows :

$$Score(t_i) = \frac{\sum_{dj \in Dr^{(s)}} d_{ji}}{\left\| Dr^{(s)} \right\|}$$

The mutation operation is done as follows:

1. For each term t_i in $Lmut^{(s)}$
2. If (random(p)<Pm) then
3. $q_{ui}^{(s)} = average(Q_i^{(s)})$
4. Endif
5. Endfor

where random(p) generates a random number p in the range [0..1]. The average function is computed as follows:

$$\text{average}(Q_u^{(s)}) = \frac{\sum_j^T q_{ui}^{(s)}}{nq_{ui}^{(s)}}$$

Where $nq_{ui}^{(s)}$ is the number of $q_{ui}^{(s)} \neq 0$ in $Q_u^{(s)}$.

3.2.5. Evolution heuristics

In order to improve the convergence conditions of the GA, we propose to add, at each generation, a niche composed of two virtual individual queries. The first one comes from applying the elitist reproduction strategy where the best individual of each generation is passed unaltered to the next generation. The second one is composed of a combination of the best terms retrieved at the current generation, according to the score formula:

$$Score(t_i) = \frac{\sum_{dj \in Dr^{(s)}} d_{ji}}{\left\| Dr^{(s)} \right\|}$$

Thus, we aim to reduce the time cost of reaching the relevant regions of the document space.

3.3. Merging method

At each generation of the GA, the system presents to the user a limited list of new documents. These documents are selected from all te retrieved documents by all the individual queries of the population, using a specific merging method. Indeed, we investigate two main methods for building the merged list according to two different rank formulas.

3.3.1. Full Merging

This merging method runs in two steps.

Step 1:

A ranked list of documents is obtained from each niche of the population by computing the following relevance measure:

$$\mathrm{Rel}_{Ni}^{(s)}(d_j)=\frac{1}{|N_i|}\sum_{Q_u^{(s)}\in Ni} RSV(Q_u^{(s)},d_j)$$

$RSV(Q_u^{(s)},d)$: RSV (Retrieval Status Value) of the document d for the query $Q_u^{(s)}$ at the generation (s) of the GA
N_i : ith niche at the current generation of the GA

Step 2:

The local lists of the documents corresponding to the different niches of the population are merged into a single list using the rank formula:

$$\mathrm{Rel}^{(s)}(d_j)=\sum_{i=1}^{Nb\_Niche(s)} Average\_Fit(N_i)*\mathrm{Rel}_{Ni}^{(s)}(dj)$$

$$Average\_Fit(N_i)=\frac{1}{|N_i|}\sum_{Q_u^{(s)}\in Ni} Fitness(Q_u^{(s)})$$

$Nb\_Niche^{(s)}$: number of niches at the generation s of the GA

The main feature of this relevance measure formula is the use of the fitness value of the niches in order to adjust the global ranking value of the output list of documents. Thus, ranking order given by the fittest niches considered more when building the outcome list of documents.

3.3.2. Selective merging

This method runs in a single step. Rather than considering the fittest niches, we consider in this case the fittest individual queries and perform a global merging of the corresponding documents retrieved using the rank formula:

$$\mathrm{Rel}_{(s)}(dj)=\sum_{Nj\in Pop^{(s)}}\sum_{Q_u^{(s)}\in N_j^{(s)}} Fitness(Q_u^{(s)**})*RSV(Q_u^{(s)},d_j)$$

$Pop^{(s)}$: population at the generation (s) of the GA
$Q_u^{(S)**}$: individual queries characterised by a fitness value higher than the average fitness of $Pop^{(s)}$

The main characteristic of this merging method is the use of the real fitness value of the fittest individual queries rather than the average fitness of the corresponding niches. Thus, we may reduce the error on the relevance assumption of the documents issued from their evaluation.

4. Experiments and results

The experiments have been carried out on a sub-collection of TREC-4 corpus. The documents we use are the AP88 newswire. We use 24 queries of TREC-4 (queries numbered 1-24). The experiments have been run using the Mercure IRS (Boughanem & Soule-Dupuy, 1997) that processes the spreading activation technique.

Prior experiments (Boughanem & al, 1999) allow us to evaluate the main parameters of the GA : crossover and mutation probability. The best performances were reached for these values: 0.7 and 0.07 respectively, which were then chosen for all the remaining experiments presented in this paper. The basic experimental conditions are the following:

- There are fifteen (15) judged documents as commonly used in relevance feedback works (Harman, 1992) (Salton & Buckley, 1990).
- The number of feedback iterations has been fixed at 5. Each feedback iteration corresponds to the judgment of the fifteen (15) documents selected from those retrieved by a new query generation of the GA.
- The niches are delimited by computing the common documents from the top fifty documents selected by each individual query.

4.1. Population parameters

The retrieval approach developed in this study is based on an iterative evaluation of query niches evolving through generations. As the population structure is an important characteristic of the approach, our first experiments aim to tune the population size and coniche limit parameters.

Table 1 presents the number of relevant documents retrieved at each feedback iteration and across feedback iterations, using various population sizes (2, 4, 6) and coniche limit values (3, 9, 15).

Experimental results in table 1 indicate that the parameters pop size and coniche limit play an important role in the effectiveness of the GA. It can be seen that the best values of cumulative relevant documents retrieved at the fifth feedback iteration (results highlighted) correspond to the following population size and coniche limit values (2, 3), (4, 9) and (6, 3). The results suggest that a large population size (greater than 4) induces the crowding of query niches number. As a niche traduces a retrieval direction, this would probably generate a noise that decrease the overall number of relevant documents.

In order to tune the population parameters we also compare the evaluation results measured by average precision (Avg Prec) and precision at 15 documents cutoff (Prec@15) for the different parameters values, as reported in table 2.

It can be seen as expected, that the best results are obtained for a population size varying from 2 to 4 (results highlighted). In order to make a good compromise between the cumulative number of relevant documents retrieved and the precision of the retrieval process at the first feedback iterations, we retain the pair value (4, 9) of population size and coniche limit values, for the remaining experiments.

Coniche limit	Iter1	Iter2	Iter3	Iter4	Iter5
			Pop size=2		
3	172(172)	113(285)	87(372)	80(452)	**70(522)**
9	172(172)	113(285)	87(372)	75(447)	71(518)
15	172(172)	113(285)	89(374)	69(443)	69(513)
Average number of niches	1	2	2	1	1
			Pop size=4		
3	180(180)	88(268)	93(361)	87(448)	61(509)
9	180(180)	88(268)	98(366)	75(442)	**78(520)**
15	180(180)	88(268)	97(365)	75(440)	57(497)
Average number of niches	1	2	3	2	2
			Pop size=6		
3	177(177)	105(282)	80(362)	61(423)	**68(491)**
9	177(177)	105(282)	78(360)	64(424)	56(480)
15	177(177)	105(282)	60(342)	68(410)	50(460)
Average number of niches	1	2	4	4	2

Table 1 : Effect of the population size and coniche limit values

The values in parentheses represent cumulative total number of documents retrieved.

Coniche limit	Iter1		Iter2		Iter3		Iter4		Iter5	
	Avg Prec	***Prec @ 15***	***Avg Prec***	***Prec @ 15***	***Avg Prec***	***Prec @ 15***	***Avg Prec***	***Prec @ 15***	***Avg Prec***	***Prec @ 15***
					Pop size=2					
3	0.20	0.47	**0.10**	**0.31**	0.07	0.24	0.05	0.22	0.03	0.19
9	0.20	0.47	**0.10**	**0.31**	0.07	0.24	0.05	0.20	0.03	0.19
15	0.20	0.47	**0.10**	**0.31**	0.07	0.24	0.05	0.19	0.04	0.19
					Pop size=4					
3	**0.21**	**0.51**	0.07	0.24	**0.06**	**0.25**	0.06	0.24	0.03	0.16
9	**0.21**	**0.51**	0.07	0.24	**0.06**	**0.27**	0.06	0.21	0.04	0.21
15	**0.21**	**0.51**	0.07	0.24	**0.06**	**0.26**	0.05	0.20	0.03	0.15
					Pop size=6					
3	0.22	0.49	0.09	0.29	0.05	0.22	0.04	0.16	0.03	0.18
9	0.22	0.49	0.9	0.29	0.05	0.21	0.04	0.17	0.03	0.15
15	0.22	0.49	0.09	0.29	0.04	0.16	0.04	0.18	0.03	0.13

Table 2 : Variation of the precision values

4.2. Genetic multiple query evaluation Vs single query evaluation

At this level, we address the question of how well our genetic combination performs relative to a single query evaluation. For this, we compare the performance results (recall-precision) from two distinct runs:

- the first one is based on a genetic combination of multiple query evaluation results as described above,
- the second one is based on a classic single query evaluation as performed in the Mercure IRS

Due to the multiple iteration aspect of the genetic evaluation and the use of relevance judgments, the results reported for the multiple query evaluation are based on a residual ranking evaluation. In our context, we measure at each iteration (corresponding to a GA generation), the performance retrieval after removing the documents retrieved and judged in the previous iterations. Let us consider that the iteration (i+1) of the GA is performed. In order to measure the effectiveness of the GA at this iteration, we compare the 15 top retrieved documents at that iteration to the single query evaluation results at the same iteration, wich corresponds to the list resulting from just keeping the next 15 documents at the iteration (i). This implies the basic question: what represents an *iteration* in the case of the single query evaluation process in order to make sense to the performance comparison? To answer this question, we consider that an iterative single query evaluation process may be based on the scanning of the overall initial output list, from the top to the bottom, using sub-lists presented to the user.This means that we analyze at each iteration, the following sub-list of documents (a sub-list is composed of 15 documents in the case of our experiments) ordered after the above list presented to the user according to the output list. Each sub-list represents the reference evaluation to compare with the genetic evaluation results obtained at the same related iteration.

Table 3 present the details of the evaluation results (measured by average precision (Avg Prec), precision at 15 documents cutoff (Prec @ 15) and number of relevant documents retrieved (Rel. Doc)) of the two runs using the merging methods previously presented.

Single Query Evaluation					
	Iter1	Iter2	Iter3	Iter4	Iter5
Avg Prec	0.12	0.07	0.05	0.03	0.02
Prec @ 15	0.30	0.25	0.22	0.18	0.17
Rel. Doc	110(110)	92(203)	82(285)	65(351)	61(412)
Genetic Multiple Query Evaluation					
Full merging					
	Iter1	Iter2	Iter3	Iter4	Iter5
Avg Prec	0.21	0.04	0.07	0.05	0.03
Prec @ 15	0.5	0.18	0.20	0.20	0.19
Rel. Doc	180(180)	65(245)	86(331)	74(406)	69(475)
Selective merging					
	Iter1	Iter2	Iter3	Iter4	Iter5
Avg Prec	0.21	0.10	0.07	0.05	0.03
Prec @ 15	0.5	0.31	0.24	0.20	0.19
Rel. Doc	180(180)	88(266)	97(366)	75(442)	78(520)

Table 3: Retrieval performances

Table 4 provides a summary of the performance of our proposed approach measured by the improvement achieved compared to the single query evaluation method.

Genetic Multiple Query Evaluation					
Full Merging					
	Iter1	**Iter2**	**Iter3**	**Iter4**	**Iter5**
Avg Prec	75%	-43%	40%	67%	50%
Prec @ 15	67%	-28%	-9%	11%	12%
Rel. Doc	63%	20%	16%	15%	15%
Selective merging					
	Iter1	**Iter2**	**Iter3**	**Iter4**	**Iter5**
Avg Prec	75%	43%	40%	67%	50%
Prec @ 15	67%	24%	9%	11%	12%
Rel. Doc	63%	32%	28%	25%	26%

Table 4: Improvements of the genetic approach

As the tables illustrate, unless the second iteration, the genetic multiple query evaluation approach yields large improvements in average precision, precision at 15 documents cutoff and number of relevant documents, for both merging methods. The negative values in the second iteration may be due to the relative important improvement achieved at the first iteration. Indeed, no much more relevant documents are retrieved at this level. Furthermore, we note however that the improvements obtained by using the selective merging method are better than those obtained using the full one. In light of these results, it would seem that the query fitness value is more significant than the niches average fitness when merging the evaluation results. This might be due to the probable variation of the performances of the individual queries belonging to the same niche. Furthermore, the results suggest that we should perform a prior selection of the individual queries before merging the corresponding results.
According to these results, we choose the selective merging method to perform the remaining experiments. The details of each experiment testing the effects of the various aspects of the genetic combination method proposed are presented below.

4.3. Knowledge based operators vs. classical operators

Table 5 compares the results of the GA using both knowledge based and classical operators. The classical crossover is based on the classical GA crossover operator. Let us consider $Q_u^{(s)}$ and $Q_v^{(s)}$ two individuals selected for crossover with *c* as the crossing point and $Q_u^{(s+1)}$ and $Q_v^{(s+1)}$ as the new individuals resulting from the classical crossover. This operator is defined as follows.

$$Q_{p1}^{(s+1)}\left(q_{u1}^{(s+1)}, q_{u2}^{(s+1)}, \quad q_{vc}^{(s)}, q_{vc+1}^{(s)} \ldots, q_{vT}^{(s)}\right)$$

$$Q_{p2}^{(s+1)}\left(q_{v1}^{(s+1)}, q_{v2}^{(s+1)}, q_{uc}^{(s)}, q_{uc+1}^{(s)}, \ldots, q_{uT}^{(s)}\right)$$

$$if(c<i)then\begin{cases} q_{p1i}^{(s)}=q_{ui}^{(s)} \\ q_{p2i}^{(s)}=q_{vi}^{(s)} \end{cases} else\begin{cases} q_{p1i}^{(s)}=q_{vi}^{(s)} \\ q_{p2i}^{(s)}=q_{ui}^{(s)} \end{cases}$$

In classical mutation, the genes are mutated by modifying their weights arbitrarily.

	Iter1	**Iter2**	**Iter3**	**Iter4**	**Iter5**
Class Op *Knl. Op*	171 (171) 180(180)	79 (250) 88 (268)	65 (315) 97 (366)	65 (380) 75 (442)	68(449) 78 (520)
Improvement/ Cum_Doc	**5,2%**	**7,2%**	**16%**	**16%**	**15%**

Table 5 : Results for knowledge based operators vs. classical operators

We clearly notice that the knowledge-based operators are more effective than the classical ones. Indeed, both the number of relevant documents and the cumulative number of documents are much higher when applying enhanced operators rather than classical ones with an improvement of 15% at the fifth generation. This supports our intuition behind the interesting use of information retrieval techniques when performing genetic transformations on the individual queries.

4.4. Virtual individuals

The artificial niche integrated to the population is composed of the fittest individual and the best concepts of the latter generation. Comparative experiments have been carried out in order to evaluate the independent effects of each virtual individual (cf. 3.2.5) and combined effects on the retrieval results and then plot histogram aspresented in figure 2. A bar indicates the percentage difference between the application of each virtual query and then both in comparison with the baseline that indicates the non integration of the artificial niche.
We clearly notice that each virtual individual has a positive effect on the search results with respectively 16% and 23% of improvement at the fifth generation of the GA.

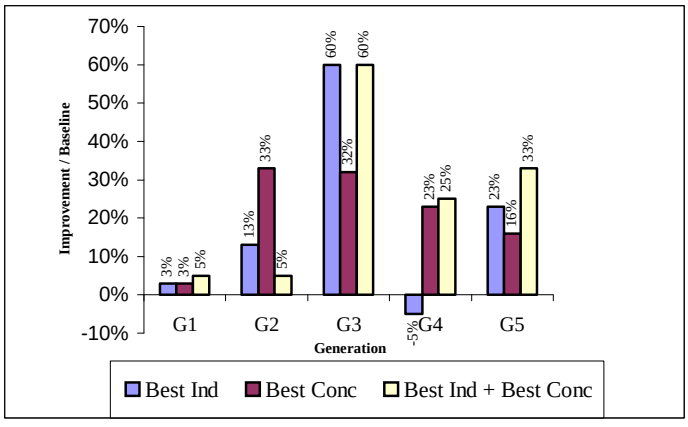

Figure 2 : Effect of the virtual individual queries

Furthermore, we notice that the effects of each individual query are not linearly added. For example, the results show that at the second generation, the independent application of these virtual queries are 33% and 13% but the improvement due to the combined application of the

virtual queries is only 5%. This may be justified by the impact of the merging method. Indeed, the rank order based on the fitness values of the query individuals are not necessary correlated to the relevance value of the corresponding retrieved documents.

4.5. Niching technique

The main goal of using niching technique is to reach different optima for a specific optimisation problem. In the context of our study, niching would allow the recall of relevant documents with quite different descriptors. In order to evaluate its precise effect on the search results, we have organised the query collection test into bins. Indeed, using Jaccard similarity measure for document selection, each query bin correponds to a document region. It is characterised by a corresponding average similarity value between relevant documents in fixed intervals:[20 25[, [25 30[, [30 35[. After this, we have plotted the histogram presented in figure 3. The x-axis represents the document bins and the y-axis represents the cumulative number of relevant documents retrieved at the fifth generation of the GA.

It can be seen that a niching technique improves the results for the first and the second bin with respectively 42% and 45% compared to the baseline. In contrast, the performances decrease in the case of the third bin. This might be due to the fact that because of the relative quite large distance between relevant documents ([30 35]), the convergence of the GA becomes slow. This suggests the use of more suitable combinations between the coniche operator definition and prior user relevance judgements.

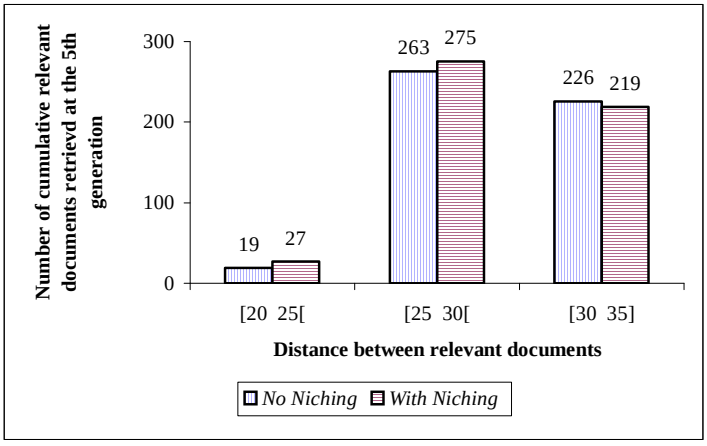

Figure 3: Effect of the niching technique

5. Conclusions

We propose in this paper a genetic approach for combination of multiple query retrieval results. The GA handles the process of relevance optimisation. It is adapted to the retrieval task context by using knowledge-based operators to guide the retrieval process, a niching technique to encourage exploration in different directions of the document space and virtual individuals to reduce the cost convergence. The results presented above prove the effectiveness of our approach.

Prior experiments have been performed on TREC 6 comparing genetic multiple query evaluation and single pass searches equivalent to a Rocchio model type search (Boughanem & al, 2000). The results have shown that the genetic approach is more effective particularly in improving recall. This is due to the robust capacity of the genetic model to reach relevant documents in different directions of the document space due to a population of query formulations, rather than a single one. Furthermore, we have showed in this paper that adding a niching technique improves the exploration effectiveness by organizing the population into niches exploring potential relevant directions of the document space.
The results presented above show also that the other heuristics integrated with the GA improve the results in few generations.

Our next goal in the near future is to integrate various advanced techniques. Our prospects concern mainly three points. The first one relates to the revision of the application of the genetic operators. More precisely, we direct ourselves towards a restrictive application of the crossing and exploitation of the principle of self-adapting mutation by controlling the probability mutation during the search (Bäck 1995), (Bäck & Schutz 1996). This adaptation will be controlled by the distribution of the terms in the relevant documents and, by the density of the areas of relevance determined by the niches of the queries.The second point would consist in exploiting the technique of clearing proposed by (Petrowski 1997. This technique leads to the safeguarding of the best queries of each niche and thus to secure possible effects of dispersion in search space due to the application of the genetic operators in one hand, and fusion in the second hand. Lastly, it would be interesting to exploit the theory of subjective optimisation (Venturini & Slimane 1997) in order to put the genetic information retrieval search within a more adapted framework to the co-operation between the user and the IRS.

References

B.T. Bartell, G.W Cortell, R.K Belew (1998). Optimising similarity using multiquery relevance Feedback, Journal of the American Society for Information Science 49(8)

T.. Bäck (1995). Evolutionary Algorithms in theory and practice. *New-York Oxford University Press*, 1995.

T. Bäck and M. Schutz (1997). Intelligent mutation rate control in canonical genetic algorithms. *W. Ras and Michalewicz Editors : Foundation of Intelligent Systems 9th International Symposium, ISMIS'96*, page 158-167, Springer, Berlin, 1996

N. J. Belkin, C. Cool, W. Bruce Croft, J. P. Callan (1993). Effect of multiple query representations on information retrieval system performance. In Proceedings of ACM SIGIR, Conference on Research and Development in Information Retrieval , pp 339-346, Pittsburgh

C. Buckley & J. Walz. The TREC8 query track. In Proceedings of the eighth Text Retrieval Conference (TREC-8) , E. Voorhees and D. Harman Editors NIST 500-246

M. Boughanem & C. Soulé Dupuy (1997). Query modification based on relevance backpropagation, In Proceedings of the 5th International Conference on Computer Assisted Information Searching on Internet (RIAO'97), Montreal pp 469-487

M. Boughanem, C. Chrisment & L.Tamine (1999): Genetic Approach to Query Space Exploration. Information Retrieval Journal volume 1 N°3 , pp175-192

M. Boughanem, C. Chrisment, J. Mothe, C. Soule-Dupuy & L. Tamine (2000). Chapter in Connectionist and Genetic Approaches to perform IR, Soft Computing, Techniques and Application, Crestani & Pasi Eds, pp 173-196

H.Chen Machine learning for Information Retrieval (1995). Neural networks, symbolic learning and genetic algorithms, Journal of the American Society for Information Science 46(3), pp 194-216

Goldberg D.E (1994). Algorithmes génétiques, exploration, optimisation et apprentissage automatique, Edition Addison Wesley

M. Gordon (1988). Probabilistic and genetic algorithms for document retrieval, Communications of the ACM pp 1208-1218

D. Harman (1992). Relevance feedback revisited: In Proceedings of ACM SIGIR, Conference on Research and Development in Information Retrieval, pp 1-10

Horng J.T & Yeh C.C (2000). Applying genetic algorithms to query optimisation in document retrieval, In Information Processing and Management 36(2000) pp 737-759

Katzer , M.J. McGill, J.A. Tessier, W. Frakes and P. DasGupta (1982). A study of the overlap among document representations. Information Technology : Research and Development, 1 (4) : pp 261-274

Kraft DH, Petry FE, Buckles BP and Sadisavan T (1995). Applying genetic algorithms to information retrieval system via relevance feedback, In Bosc and Kacprzyk J Eds, Fuzziness in Database Management Systems Studies in Fuzziness Series, Physica Verlag, Heidelberg, Germany pp 330-344

K.L Kwok (1995). A Network approach to probabilistic information retrieval, ACM Transactions on Information Systems, Vol 13 N3 pp 324 – 353

J.H. Lee (1997). Analyse of multiple evidence combination , In Proceedings of ACM SIGIR, Conference on Research and development in Information Retrieval pp 267-275

MCGill, Koll & Norreeault (1979). An evaluation of factors affecting document ranking by IR systems, Syracuse, Syracuse university school of information studies

A. Petrowski. (1997). A new selection operator dedicated to speciation. International Conference on Genetic Algorithms *ICGA*, july 19-23, 1997

S.E Robertson (1977). The probability ranking principle in IR, Journal of documentation 33 (4), pp 294 – 304

S.E. Robertson, S.Walker, S.Jones, M.M Hancock Beaulieu and M. Gatford (1995) : Okapi at TREC3, In Proceedings of the third Text Retrieval Conference (TREC-3)

G. Salton, C. Buckley (1990). Improving retrieval performances by relevance feedback, Journal of the American Society for Information Science, Vol 41, N°4, pp 288-297

L. Tamine (2000). Optimisation de requêtes dans système de recherche d'information, approche basée sur l'exploitation de techniques avancées de l'algorithmique génétique. Doctorat thesis, University Paul Sabatier, Toulouse, France

L.Tamine & M. Boughanem (2000). Query optimisation using an improved genetic algorithm, Conference on Information knowledge and Management (CIKM) , Washington 6-11 November 2000, pp 368-373

Turtle H. & Croft W. B. (1991). Evaluation of an inference network-based retrieval model ACM transactions on information systems, 9, 3 : 187 - 222

G. Venturini, M. Slimane (1997), F. Morin and Asselin Beauville. On using interactive genetic algorithms for knowledge discovery. *International Conference on Genetic Algorithms, ICGA'97* july 19-23, 1997.

J. J Yang & R. R Korfhage (1993). Query optimisation in information retrieval using genetic Algorithms, in Proceedings of the fifth International Conference on Genetic Algorithm (ICGA), pp 603-611, Urbana, IL