QUERY OPTIMISATION USING AN IMPROVED GENETIC ALGORITHM

L.TAMINE M.BOUGHANEM
lechani@wissal.dz *bougha@irit.fr*

IRIT SIG Université de ToulouseIII 118 Route de Narbonne,
31062 Toulouse, France

Abstract

This paper presents an approach to intelligent information retrieval based on genetic heuristics. Recent search has shown that applying genetic models for query optimisation improve the retrieval effectiveness.
We investigate ways to improve this process by combining genetic heuristics and information retrieval techniques. More precisely, we propose to integrate relevance feedback techniques to perform the genetic operators and the speciation heuristic to solve the relevance multimodality problem. Experiments, with AP documents and queries issued from TREC, showed the effectiveness of our approach.

Keywords: Information Retrieval, Genetic Algorithm, Relevance Feedback

1. INTRODUCTION

With the proliferation of the World Wide Web and the widespread use of web search engines, the number of user profiles of IR systems grows and the information collections become wider and more heterogeneous. Therefore, it becomes more difficult to retrieve relevant information.
Several researches continue to deal with this problem. The technique commonly used is automatic query expansion and reweighting via relevance feedback [23] [22] [14] [15]. In situations where there is no relevance judgement, pseudo relevance feedback technique is used to expand the user query [7] [21].
Pseudo relevance feedback is based on the assumption that a set of the top retrieved documents are relevant. Thus, the retrieval performance depends on the quality of the initial search. Pseudo relevance feedback seems capable of both improving and hurting performance for different queries [20]. everal other studies in query improvement were based on hybrid techniques using neural approach [29] [23] [3].
Recently, there has been a growing interest in applying genetic algorithm to handle the process of IR. Genetic algorithms have been shown to be powerful search mechanism and seem to be suitable in IR for the main following reasons [27]:

– The document space represents a high dimensional space. As genetic algorithms 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. Thanks to their inherent properties of implicit parallelism, 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. But several experiments have 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 occurring in 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 exploring of new zones in the document space independently from the initial query.

This paper presents our GA approach to query optimisation. The GA model we propose is characterised by:

– Using a population of query niches [9] that explore several directions in the document space; we believe that this allows the exploration mechanism to retrieve documents with different descriptions in response to vague queries.

– Improving the genetic operators with relevance feedback techniques.
– Integrating virtual individuals in the population during the GA evolution.

Section 2 describes an overview of genetic algorithms in IR. Section 3 presents the query optimisation model; it describes the query optimisation process and presents the main characteristics of the GA. Finally, experiments performed on documents issued from TREC [28] program and discussions of the results are presented in the last section.

2. OVERVIEW OF GENETIC ALGORITHMS IN INFORMATION RETRIEVAL

Genetic algorithms [8] are stochastic optimisation methods based on principles of evolution and heredity. A GA maintains a population of potential solutions to a given optimisation problem. The population is renewed at each generation using both a fitness measure to evaluate the individuals and genetic transformations to reproduce the fittest ones.
The general theory of GA proves the main following properties [9]:

Implicit parallelism
When manipulating an n size population, GA explores simultaneously a number of directions running to n^3.

Resolution of exploration/exploitation dilemma
The genetic programming resolves efficiently the exploration/exploitation dilemma by allowing an exponentially increasing number of copies of the fitter individuals. Therefore, encouraging exploration in good directions.

2.1. An abstract GA

The abstract GA contains the following basic steps:

0. Set the initial population $Pop^{(0)}$
1. Compute the fitness of each individual
2. Select parent sets
3. Produce the children of the selected parent sets
4. Check the termination condition
   if true then output the best individual and stop
5. Reduce the extended population
6. Goto step 1.

The children of each generation are produced using selection, crossover and mutation operators [8.
Several studies have suggested the use of heuristics to improve the control on the genetic exploration. We note niching and speciation techniques [9] knowledge based operators and adaptive control methods [12] [25].

2.2. Related works in GA and IR

The development of scheme theory invented by Holland [16] and some theoretical studies in GA [1], have attracted scientists from several research areas. Some works and studies have been done in the IR area and we discuss a selection of these below.
Gordon [10] adopted 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 showed 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. Gordon exploited these results and defined a classification method [11] based on clustering the relevant documents for a specific query.
Yang & Korfhage proposed a GA to query optimisation by reweighting the document term indexing without query expansion [31]. 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 showed that the queries converge to their relevant documents after six generations.
Chan proposed a hybrid genetic and neural network based system called GANNET [6]. This system performs concept optimisation for user selected document using GA and uses the optimised concepts to perform concept exploration in a Hopfield net representing related concepts. The retrieving process is cyclic and is done in two stages. The first stage is the concept optimisation; the GA manipulates input documents and their associated keywords to generate an initial set of optimised concepts. The second stage is the concept exploration, the set of optimised concepts that are included in GA for the next concept optimisation. This process is repeated until there is no further improvement.
Kraft & al apply GA programming in order to improve the weighted Boolean query formulation [18]. The documents are viewed as a vector of index terms. A weighted Boolean query is represented as chromosome in Koza's genetic model [17]. The goal of the GA is to modify the query in order to improve the search performance in term of recall and precision. Their first experiments showed that the GA programming is viable method for deriving good queries.

3. THE QUERY OPTIMISATION MODEL

Our GA model handles the process of query optimisation; thus it aims to reach optimal or near optimal queries which produce the best outcomes of the system, according to the user query.
The main characteristics of our GA model are summarised in the following :

- ***Niching***
  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 at different regions of the document space and therefore have some different descriptors.
  According to this assumption, we use the niching ecological technique [9] in order to explore the document space by encouraging the reproduction queries in different directions rather than reaching a unique optimal query when using a classical genetic exploration.

- ***Restrictive application of enhanced operators***
  Relevance feedback is an effective technique commonly used in information retrieval [15] [19] [21] [3]. Rather than using blind genetic operators, we propose enhanced ones, which aim to expand and reweight individual queries using the relevance user's judgements.
  Furthermore, these operators are applied in the same niche in order to renew it and measure the goodness of the search direction it represents.

- ***Initial population***
  The initial population is not randomly constructed. It is composed of the user's query and the descriptors of the relevant documents retrieved at the initial search; if no relevant documents are retrieved at this stage, we process adhoc feedback.
  All the individual queries are initially grouped in the same niche.

Notations

T	Total number of stemmed terms automatically extracted from the documents
N	Total number of documents
t_i	ith term
n_i	Number of documents containing term t_i
d_j	jth document
tf_{ji}	frequency of t_i in d_j
d_{ji}	term weight of t_i in d_j
$Q_u^{(s)}$	query individual u at the generation (s) of the GA
$q_{ui}^{(s)}$	weight of the term i in $Q_u^{(s)}$
$pop^{(s)}$	Population at the generation (s) of the GA
$D_r^{(s)}$	set of relevant documents retrieved by $pop^{(s)}$
$D_{nr}^{(s)}$	set of non relevant documents retrieved by $pop^{(s)}$
$D_{QU}^{(s)}/L$	the L top documents retrieved by Q_u
$Rel_N^{(s)}(D)$	assumed local RSV of the document D in the niche N at the generation (s) of the GA
$Rel^{(s)}(D)$	assumed global RSV of the document D at the generation (s) of the GA
$RSV(Q_u,D)$	Retrieval Status Value of the document D when submitting the individual query $Q_u^{(s)}$
$Nb\_Niche^{(s)}$	number of the niches at the generation (s) of the GA
Niche-Size	size of a niche
Coniche limit	the min number of common documents retrieved by queries of the same niche
$N_j^{(s)}$	jth niche at the generation (s) of the GA
$\lvert N_j^{(s)}\rvert$	size of $N_j^{(s)}$
$Average\_Fit(N_j^{(s)})$	average fitness of $N_j^{(s)}$
J	Jaccard measure

3.1 The query optimisation process

The general query optimisation process is done as follows :

```
1. Submit the initial query and do the search
2. Judge the top thousand documents
3. Build the initial population
Repeat
4. For each niche of the population
      do the search
      build the local list of documents
   Endfor
5. Build a merged list
6. Renew the niches
7. Judge the top fifteen documents
8. Compute the fitness of each individual query
9. for each niche $N^{(s)}$ of the population
     Repeat
       parent1= Selection ($N^{(s)}$)
       parent2= Selection ($N^{(s)}$)
       Crossover (Pc , parent1, parent2,son)
       Mutation (Pm , son, sonmut)
       Add_Niche (sonmut,$N^{(s+1)}$)
     Until Niche_size ($N^{(s+1)}$) = Niche_size ($N^{(s)}$)
Until a fixed number of feedback iterations
```

3.2. Description of our GA

The following section presents the details of the GA processing the query optimisation

3.2.1. Individual, Niche and population

- Individual

In our approach, the genetic individual is a query. Each gene corresponds to an indexing term or concept. Its value or *locus* is represented by a real value and defines the importance of the term in the considered query. Each individual representing a query is of the form :

$$Q_u\ (q_{u1}, q_{u2},\ldots,q_{uT})$$

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 :

$$q_{ui}=\frac{(1+\log(tf_{ui}))*\log(\frac{N}{n_i})}{\sqrt{\sum_{k=1}^{T}((1+\log(tf_{uk}))*\log(\frac{N}{n_k}))^2}}$$

- Niche

A niche is a set of individual queries exploring in a potential region of the document space. The theory of genetic niching technique [9] shows that the exploration process discovers relevant regions using different directions, that is we name *parallel and cooperative query search.* We define the *coniche* operator,(i.e. queries belonging to the same niche) as following :

$$[\ Q_u^{(s)} \equiv_N Q_v^{(s)}\ ] \Leftrightarrow [(D_{Qu}^{(s)}/L) \cap (D_{Qv}^{(s)}/L) > Coniche\_Limit\ ] \quad (2)$$

The size and the structure (individual components) of the niche evolve at each generation due to both the retrieval process and genetic transformations. Furthermore, we can note that niches are not inevitably independent.

- Population

The population is renewed at each generation. It contains the whole niches build according to the expression (2) adding a virtual query wich represents the best terms retrieved at the corresponding feedback iteration, and the fittest individual query of the latter generation.

3.2.2. 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 [13] :

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

The most favourable feature of the Guttman model function is that it is highly correlated with the standard goodness measure in IR that is average precision [2].

3.2.3. Genetic operators

The genetic operators defined in our approach are not classical ones. They have been adapted to take advantage of techniques developed in IR. Thus, we qualify them as knowledge based operators. Adding to this, they are restrictively applied to the niches and so do the population size varying during the evolution of the GA.

- Selection

The selection procedure is based on a variant of the usual roulette wheel selection [9]. 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 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 modifying the 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 :

$$\begin{tikzcd}
Q_u^{(s)}(\, q_{u1}^{(s)}, q_{u2}^{(s)}, \ldots, q_{uT}^{(s)}) \arrow[dr, leftarrow] & & Q_v^{(s)}(\, q_{v1}^{(s)}, q_{v2}^{(s)}, \ldots, q_{vT}^{(s)}) \arrow[dl, leftarrow] \\
 & Q_p^{(s+1)}(\, q_{p1}^{(s+1)}, q_{p2}^{(s+1)}, \ldots, q_{pT}^{(s+1)}) &
\end{tikzcd}$$

$q_{pi}^{(s+1)} = \mathrm{Max}\ (q_{ui}^{(s)}, q_{vi}^{(s)})$
if importance $(t_i, D_r^{(s)}) \geq$ importance $(t_i, Dn_r^{(s)})$
Min $(q_{ui}^{(s)}, q_{vi}^{(s)})$ otherwise

We defined : $importance(t_i,D)=\sum_{dj\in D} d_{ji}$

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 [25]. Let us consider $Q_u^{(s)}$ as the selected individual query and $L^{(S)}$ as the set of terms from $D_r^{(S)}$ 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 $L^{(S)}$ terms and on the probability Pm . The $Lmut^{(s)}$ terms are sorted according to a score value calculated as follows :

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

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
```

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 $n_{qui}^{(S)}$ is the number of $q_{ui}^{(s)} \neq 0$ in $Q_u^{(s)}$.

3.2.4. 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 the whole ones retrieved by all the individual queries of the population, using a specific merging.
The merging method we propose runs in two steps. In the first step, a ranked list of documents is obtained from each niche of the population by computing the following relevance measure :

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

In the second step, the local lists of documents corresponding to the different niches of the population are merged into a unique list using the rank formula :

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

$$Average\_Fit(N_j^{(s)})=\frac{1}{\left|N_j^{(s)}\right|}\sum_{Q_u^{(s)}\in N_j^{(s)}} Fitness(Q_u^{(s)})$$

The main feature of the 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 is more considered when building the outcome list of documents.

4. EXPERIMENTS AND RESULTS

4.1. Experiments

The experiments were carried out on AP Documents with15 queries issued from TREC program. They were run using the Mercure IR system [3] that process the search using the spreading activation technique.
The main goal of these experiments was to evaluate the effectiveness of our GA model for IR. More precisely, we measure the effects of the initial size of population, coniche limit value and knowledge based operators comparing with blind ones.
The basic experimental conditions are the following :

- There are fifteen (15) judged documents as commonly used in relevance feedback works [15] [24].
- The number of feedback iterations has been fixed at 5. Each feedback iteration corresponds to the judgement of the fifteen(15) documents selected from those retrieved by a new query generation of the GA.
- The values of the crossover and mutation probability are respectively fixed at 0.7 and 0.3. The values are chosen after prior experiments [4].
- The niches are delimited by computing the common documents on the top fifty selected by each individual query.
- The coniche limit value is fixed as a proportion of the number of judged documents.

4.2. Evaluation method

Because of the multiple iteration aspect of the search and the use of relevance judgement, the results reported in the paper are based on a residual ranking evaluation [5]. This method is used to evaluate the effectiveness of relevance feedback methods. In this method, all the documents previously judged are removed from the document rankings produced by both the initial query, which corresponds to iteration 0 in our algorithm, and the feedback query, which corresponds to iteration 1 in our algorithm. Precision and recall are computed for these and then for both residual lists of documents. In the case of multiple iteration, the comparison is done in the same way between the residual documents retrieved at iteration (i) to the residual document retrieved at iteration (i+1). This tells us how much we gained by doing the next iteration of the GA.

4.3. Results and discussion

4.3.1. *Effects of the GA parameters :*

Initial population size, coniche limit

It is well known in GA literature [8][9] that the population size has an important effect on the results of the genetic optimisation process.
In the case of our approach, the population size of the individual queries increases from an initial value, according to the coniche limit value fixed in the definition of the coniche operator. Therefore, the first experiment has been performed by varying both initial population size and coniche limit values.

Iter	*1*	*2*	*3*	*4*	*5*	*6*
P_S=2						
C_L						
3	94(94)	62(156)	40(196)	59(255)	48(303)	**45(348)**
9	91(91)	60(151)	40(191)	56(247)	42(289)	48(337)
15	91(91)	60(151)	40(191)	51(242)	53(295)	44(339)
P_S =4						
C_ L						
3	100(100)	43(143)	48(192)	45(237)	38(275)	**40(315)**
9	100(100)	43(143)	50(193)	45(238)	38(276)	32(308)
15	100(100)	43(143)	50(193)	40(233)	45(278)	35(314)
P_S =6						
C_L						
3	102(102)	40(142)	40(182)	44(226)	37(263)	31(291)
9	102(102)	40(142)	42(184)	43(227)	41(268)	**28(296)**
15	102(102)	40(142)	43(185)	45(230)	48(278)	33(311)

Table 1: ***Effects of the GA parameters initial population size and coniche limit***

Values represent number of relevant documents retrieved at iteration and the values in parentheses represent cumulative total number of documents retrieved.

Table 1 lists the number of relevant documents retrieved at each feedback iteration and the cumulative total number of relevant documents retrieved by that point for initial population size ∈ {2, 4, 6 } and coniche limit ∈ {3, 9, 15 }. The table shows also the variation of the population size and the number of population niches during the evolution of the GA.
We notice that for the same initial population size, the coniche limit value has not a great effect on the results. We think that because of the limited number of judged documents, the structure of the niche does not vary widely and so it recalls quite the similar documents at the top list.
In contrast, it can be seen that the initial population size is a determinant factor in the retrieval process. We particularly notice that the retrieval performances decrease for increasing initial population sizes. This supports the idea that a large population size induces the multiplicity of retrieval directions due to the variation of the number of niches.
The setting highlighted show that the best results are obtained for initial pop size equal to *2* and coniche limit equal to *3*. We retained these values for our algorithm for the remaining experiments.

Iter	*1*	*2*	*3*	*4*	*5*	*6*
With GA	94(94)	62(156)	40(196)	59(255)	48(303)	**45(348)**
NO GA	71(71)	78(149)	50(199)	31(230)	38(268)	33(303)

Table 2 *:* ***Results for GA vs. No GA retrieval process***

Table 2 compares the results using no GA and using GA for different iterations. We notice that with GA, the total number of relevant documents after *6* iterations is much higher than using no GA. More precisely, in order to show the effects of GA processing on the system outcomes at each generation, we plot histogram presented in figure 1. A bar above the x-axis indicates that the GA processing outcomes a greater number of relevant documents at the corresponding generation. In contrast, a bar below the x-axis indicates that No GA processing outcomes a greater number of relevant documents at the corresponding feedback iteration.

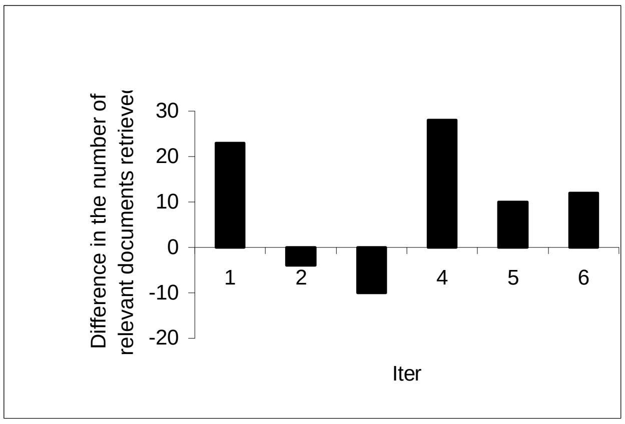

Figure 1 : ***Comparison of the effects of GA and NoGA on the system recall***

We notice that for the first iteration corresponding to the processing of the initial population in the case of GA, the bar is high and above the x-axis. In one hand, this shows us the positive effect of the method performed to construct the initial population. In other hand, this allows us to explain the negative change in the two succeeded iterations (2 and 3). Indeed, as an important number of relevant documents are retrieved at the first iteration, the followed iterations did not perform better. However, the GA processing recall a greater number of relevant documents than No GA processing at all the other succeeded iterations.

4.3.2.Effects of the knowledge based operators

Table 3 compares the results of the GA using the knowledge based operators and the blind ones.

Iter	*1*	*2*	*3*	*4*	*5*	*6*
Knl. Op	94(94)	62(156)	40(196)	59(255)	48(303)	45(348)
Bld. Op	91(91)	43(139)	36(169)	40(209)	27(235)	30(266)

Table 3 : ***Results for knowledge based operators vs. blind operators***

We clearly notice that the knowledge-based operators are more effective than the blind ones. This supports our intuition behind the interesting use of information retrieval techniques when performing the genetic transformations on the individual queries.

4.3.3. Effects of the merging method

The merging of the whole documents selected by all the individual queries of the population is an important operation in our GA. Indeed, despite a good recall value for the union of the local lists corresponding to the niches, the merging method could decrease precision in the top rank outcome list. In order to check this, we performed an experiment using a classical merging method based on the average RSV at both first and second step of the method defined above.

Iter	*1*	*2*	*3*	*4*	*5*	*6*
Mrg. based on fit. val	94(94)	62(156)	40(196)	59(255)	48(303)	45(348)
Mrg based on rank. val	91(91)	71(162)	50(212)	41(253)	47(300)	40(339)

Table 4. : ***Effect of the merging method***

Table 4 compares the results of the GA merging methods using at the second step formula based respectively on fitness value and rank value. The table shows that the merging formula based on fitness value produces fairly better results than one based on average rank value. The results might be improved by using more suitable ranking formula, probably by revising the contribution of the fitness value of an individual query when computing the ranking of the corresponding retrieved documents.

5. CONCLUSION

In the study presented in this paper, we investigate ways to improve a GA for query optimisation. The results presented prove the effectiveness of our GA approach to improve the performance of an information retrieval system. We mainly focused on the interesting use of niching technique to recall relevant documents at various regions of the document space and knowledge based operators to guide the retrieval process by exploiting effective retrieval techniques. In future, we would like to perform further experiments in several directions. First we aim to develop a better merging algorithm and perform adhoc feedback using a greater number of GA iterations. Our next goal is to use our approach on very large collections in order to make a global comparison between the several GA parameters.

References

[1] Ankenbrandt C. : An Extension to the Theory of Convergence and a Proof of the Time Complexity of Genetic Algorithms, FOGA90, pp 53-58, 1990

[2] B.T. Bartell, G.W Cortell, R.K Belew, Optimising Similarity Using Multiquery Relevance Feedback

[3] M. Boughanem & C. Soulé Dupuy : 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

[4] M. Boughanem, C. Chrisment & L.Tamine, Genetic Approach to Query Space Exploration. Information Retrieval Journal volume 1 N°=3 , pp175-192

[5] Chang YK, Cirillo GC and Razon J Evaluation of Feedback Retrieval using Modified freezing, Residual Collection and Test and Control Groups. In : The SMART Retrieval System : Experiments in Automatic Document Processing, Prentice Hall Inc., chap 17, pp 355-370.

[6] H.Chen Machine learning for Information Retrieval : Neural Networks, Symbolic Learning and Genetic Algorithms, JASIS 46(3) : 194-216

[7] D.Evans and R.Lefferts Design and Evaluation of the CLARIT TREC2 System. In Proc of the second Text Retrieval Conference (TREC-2). NIST Special Publication 500-215, 1994

[8] Goldberg D.E : Genetic Algorithms in Search, Optimisation and Machine Learning, Edition Addison Wesley 1989

[9] Goldberg D.E : Algorithmes Génétiques, Exploration, Optimisation et Apprentissage Automatique, Edition Addison Wesley, 1994

[10] M. Gordon : Probabilistic and Genetic Algorithms for Document Retrieval, Communications of the ACM pp 1208-1218, October 1988

[11] M. D. Gordon : User-Based Document Clustering By Redescribing Subject Descriptions with a Genetic Algorithm, Journal of The American Society for Information Science, 42(5) pp 311 - 322, 1991

[12] Grefenstette J. J. : Virtual Genetic Algorithms: First Results, Technical Report AIC-95-013 , Navy Center for Applied Research In Artificial Intelligence, February 1995

[13] Gutman L. What is Who What in Statistics. The Statistician, 26 : 81 :107

[14] D. Haines & W.B Croft : Relevance Feedback and Inference Networks, Conference on Research and Development in Information Retrieval (SIGIR), pp 2-11, 1993

[15] D. Harman : Relevance Feedback Revisited : Conference on Research and Development in Information Retrieval (SIGIR), pp 1-10, 1992

[16] Holland J. : Les Algorithmes Génétiques, Revue POUR LA SCIENCE N°=179 pp 44-51, Septembre 1992

[17] Koza JR A Hierarchical approach to Learning the Boolean Multiplexer function In Rawlins G. Ed Foundations of Genetic Algorithms , Morgan Kauffman, San Mateo, CA, pp 171-192

[18] Kraft DH, Petry FE, Buckles BP and Sadisavan T 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

[19] K.L Kwok : A Network Approach to Probabilistic Information Retrieval, ACM Transactions on Information Systems, Vol 13 N3 pp 324 - 353, 1995

[20] M. Mitra, A.Singhal and C.Buckley, Improving Automatic Query Expansion. In Proc of the ACM SIGIR Conference on Research and Development in Information Retrieval 206-214, 1998

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

[22] Robertson S. and Walker S. On Relevance Weights with Little Relevance Information ACM/SIGIR International Conference on Research and Development in Information Retrieval, pp16-24

[23] G. Salton, The SMART Retrieval System. Prentice Hall, Inc. Englwood Cliffs NJ

[24] G. Salton, C. Buckley : Improving Retrieval Performances by Relevance Feedback, Journal of the American Society for Information Science, Vol 41, N°4, pp 288-297, 1990

[25] M. Sebag & M. Schoenauer : Contrôle d'un Algorithme Génétique, Revue D'intelligence Artificielle, Volume N° 2-3, pp 389-428, 1996

[26] L. Tamine : Reformulation Automatique de Requête basée sur l'Algorithmique Génétique, Actes du Congrès Inforsid, pp 643-662, Toulouse Juin 1997

[27] L. Tamine Les SRI : Reformulation de Requête et Apprentissage basés sur les Algorithmes Génétiques, MasterThesis, University of Tizi-Ouzou

[28] TREC Overview. In Proc of the seventh Text Retrieval Conference (TREC-7). 1999

[29] R. Wilkinson & P. Hingston : Using The Cosine Measure in A Neural Network for Document Retrieval, Conference on Research and Development in Information Retrieval (SIGIR), pp 202-210, Chicago (USA), 1991

[30] S.K.M Wong, Y.J Caï & Y.Y Yao : Computation of Term Association by a Neural Network, Conference on Research and Development in Information Retrieval (SIGIR), pp 107-115, 1993

[31] J. J Yang & R. R Korfhage : Query Optimisation in Information Retrieval Using Genetic Algorithms, ICGA'93