Applying Heuristics to Improve A Genetic Query Optimisation Process in Information Retrieval

Lynda Tamine, Mohand Boughanem

Applying Heuristics to Improve A Genetic Query Optimisation Process in Information Retrieval

L. Tamine
IRIT SIG Université de ToulouseIII 118 Route de Narbonne,
31062 Toulouse, France
lechani@wissal.dz

M. Boughanem
IRIT SIG Université de ToulouseIII 118 Route de Narbonne,
31062 Toulouse, France
bougha@irit.fr

Abstract

This work presents a genetic approach for query optimisation in information retrieval. The proposed GA is improved y heuristics in order to solve the relevance multimodality problem and adapt the genetic exploration process to the information retrieval task. Experiments with AP documents and queries issued from TREC show the effectiveness of our GA model

Key words : Information retrieval, genetic algorithm, Relevance feedback, niching technique

1 Introduction

The advent of the world wide web has significantly increased the need for automated information retrieval for large and heteregeonous collections. In this context, finding relevant information becomes a difficult task. This has attracted several researches in information retrieval domain. The main obstacle is the absence of a well defined underlying model to express both documents and user information need. For this reason, strategies and techniques operate usually with any retrieval model in order to improve the retrieval results. The most common used is still query reformulation via relevance feedback [12] [11] [18].

However even though relevance feedback significantly improves the performance of the search, it does not permit the recall of all possible relevant documents.

Recently, there has been a growing interest in applying genetic algorithms to handle the process of information retrieval. Genetic algorithms constitute an interesting category of modern heuristic search. Based on the powerful principle of *"survivle of the fittest"*, genetic algorithms model some natural phenomena of genetic inheritance and Darwinian strife of survival.

Genetic algorithms have been shown to be powerful search mechanism and seem to be suitable in information retrieval for the main following reasons [21]:

- The document space represents a high dimensional space. As GA 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 a genetic query optimisation process. The genetic algorithm 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 need 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 [6] in order to explore the document space by encouraging the reproduction of the queries in different directions rather than reaching a unique optimal query when using a classical genetic exploration.

2- Restrictive application of enchanced operators

Relevance feedback is an effective technique commonly used in information retrieval [12] [16] [17] [3]. Rather than using classical genetic operators, we propose enhanced ones, which aim to expand and reweight individual queries using the 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.

3- Virtual individuals in order to improve the convergence conditions of the GA.

Section 2 describes an overview of genetic algorithms in information retrieval. Section 3 presents the details of the query optimisation model proposed. Finally, experiments performed on documents issued from TREC [23] program and discussions of the results are presented in the last section.

2 An overview of genetic information retrieval

The development of scheme theory invented by Holland [13] 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 [8] 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 [9] 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 [24]. 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 [15]. The documents are viewed as a vector of index terms. A weighted Boolean query is represented as chromosome in Koza's genetic model [14]. 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 a viable method for deriving good queries.

Comparatively to these works, our genetic approach has two main advantages :
- The query optimisation process deals with any retrieval model
- The GA takes advantages from domain techniques by using appropriate heuristics to improve the retrieval task.

3 The genetic query optimisation algorithm

Our GA model handles the process of query optimisation; thus it aims to operate genetic transformations on the individual queries in order to reach the best set of relevant documents according to the user query. For this purpose, the population of queries is organised into niches. Each niche explores a potential direction of the document space and is renewed according to it's goodness computed using the fiteness function.
The general query optimisation process is done as follows :

```
Begin
    Submit the initial query and do the search
    Judge the top thousand documents
    Build the initial population
    Repeat
        For each niche of the population
            do the search
            build the local list of documents
    Endfor
    Build a merged list
    Renew the niches
    Judge the top fifteen documents
    Compute the fitness of each individual query
    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
End
```

The next subsections describe the structure's details of the GA proposed.

3.1 Individual Encoding

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 :

$$\mathbf{Q_u^{(s)} (q_{u1}, q_{u2},...,q_{uT})}$$

T : Total number of stemmed terms automatically extracted from the documents

$Q_u^{(s)}$: query individual u at the generation (s) of the GA
$q_{ui}^{(s)}$: weight of the term i in $Q_u^{(s)}$

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 :

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

N : Total number of documents
n_i : Number of documents containing term t_i
tf_{ji} : frequency of t_i in d_j
t_i : ith term
d_j : jth document

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 Population generation and distribution

The population is organised into several subpopulations of individual queries, called niches.
A niche is a set of individual queries exploring the document space in a potential region . The theory of genetic niching technique [7] 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 :

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

Coniche limit : the min number of common documents retrieved by queries of the same niche
$D_{QU}^{(s)}/L$: the L top documents retrieved by $Q_u^{(s)}$

Thus, queries belonging to the same niche have a significant number of common documents resulting from the evaluation. The size and the structure (individual components) of the niche evolve at each generation due to both the retrieval process and genetic transformations.

3.3 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 [10] :

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

J : Jaccard measure
$D_r^{(s)}$: set of relevant documents retrieved at the generation s
$D_{nr}^{(s)}$: set of non relevant documents retrieved at the generation s
dr : relevant document
dnr : irrelevant document

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

3.4. 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 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 [7] [20]. 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 :

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

$q_{pi}^{(s+1)} = \mathrm{Max}\,(q_{ui}^{(s)}, q_{vi}^{(s)})$

if importance $(t_i, D_r^{(s)}) \geq$ importance $(t_i, Dn_r^{(s)})$

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

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

d_{ji} : term weight of t_i in d_j

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 $L^{(S)}$ 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 mutation probability Pm . The $L^{(s)}$ 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 :

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

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

3.5 Evolution heuristics

In order to improve the search results at few generations of the GA, we propose to add at each generation, a niche composed of two virtual individual queries.

The first one is issued from the apply of the elitist reproduction strategy where the fittest individual of each generation is passed unaltered to next generation.

The second one is composed of a combination of the best concepts retrieved at the current generation, according to the score formula :

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

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

3.6 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 method.

Indeed, the ranked list of documents is obtained using the rank formula :

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

$Pop^{(s)}$: population at the generation (s) of the GA
$RSV(Q_u,D)$: assumed RSV of the document D in the niche N at the generation (s) of the GA
$N_j^{(s)}$: jth niche at the generation (s) of the GA

The main feature of the relevance measure formula, is the use of the fitness value of the individual queries in order to adjust the global ranking value of the output list of documents. Thus, ranking order given by the fittest queries 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 with 25 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 knowledge based operators comparing with blind ones, niching technique and the integration of virtual individuals on the search results.

Prior experiments [4] [22] allowed us to evaluate the main parameters of the GA : crossover and mutation probability, population size and coniche limit value. The best performances have been reached for respectively these values: 0.7, 0.07, 4, 0.6 and then were chosen for all the remain 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 [12] [19].

- 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 niches are delimited by computing the common documents on the top fifty selected by each individual query.

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 manual 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

The goal of the first experiments was to evaluate the effectiveness of our genetic query optimisation approach. Table 1 compares the outcomes of the system when applying GA processing to those obtained by a classical query evaluation using Mercure IRS.

Generation	*1*	*2*	*3*	*4*	*5*
With GA	180 (180)	88 (268)	97 (366)	75 (442)	78 (520)
No GA	110 (110)	114 (225)	77 (302)	69 (371)	65 (437)
Improvement/ Cum_Doc	**63%**	**32%**	**28%**	**25%**	**26%**

Table 1 : ***Results for GA vs. No GA retrieval process***

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

Table 1 shows that the genetic optimisation process improve the results of the search. Indeed, we note an improvement varying between 25% to 63% depending on the GA generation. The following experiments show the effect of the genetic heuristics proposed on the retrieval results.

4.3.1. Effects of the knowledge based operators

Table 2 compares the results of the GA using the knowledge based operators and the blind ones.

Generation	*1*	*2*	*3*	*4*	*5*
Bld Op	171 (171)	79 (250)	65 (315)	65 (380)	68(449)
Knl. Op	180(180)	88 (268)	97 (366)	75 (442)	78 (520)
Improvement/ Cum_Doc	**5,2%**	**7,2%**	**16%**	**16%**	**15%**

Table 2 : ***Results for knowledge based operators vs. blind operators***

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

4.3.2. Effects of the 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 and combined effects on the retrieval results and then plot an histogram presented in figure 1. A bar indicates the percentage difference between the apply of solely each virtual query and both them comparatively to the baseline witch traduces 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.

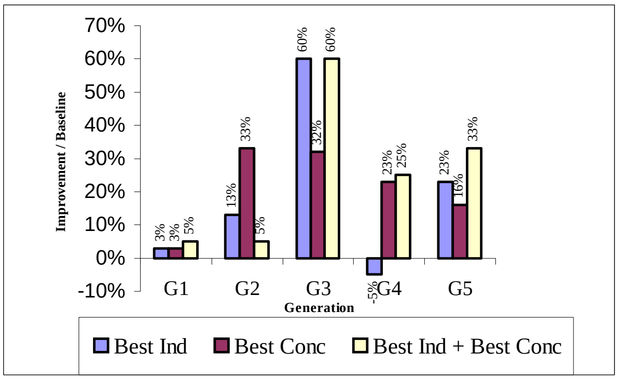

Figure 1 : Effect of the virtual individual queries

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

4.3.3. Effect of the 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 to recall relevant documents with quite different descriptors. Table 3 compares the results of the GA across five generation when using niching technique to those obtained when using a classical genetic processing on the whole population.

Generation	*1*	*2*	*3*	*4*	*5*
With Niching	180 (180)	88 (268)	97 (366)	75 (442)	78 (520)
No Niching	177 (177)	124 (302)	84 (387)	64 (451)	56 (507)

Table 3 : ***Effect of the niching technique***

This table shows globally the effectiveness of the niching technique. In order to evaluate it's precise effect on the search results, we have organised the query collection test into bins. Each bin 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 2.

This figure show that nichnig technique improves the results for the first and the second bin with respectively 42% and 45% comparatively 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 related quite important distance between relevant documents, the convergence of the GA become slow. Indeed, the results obtained at the sixth generation for this specific bin seem to be better with an improvement evaluated to 16%. Thus, the results might be improved by using more suitable combination between the coniche operator definition and prior user relevance judgements.

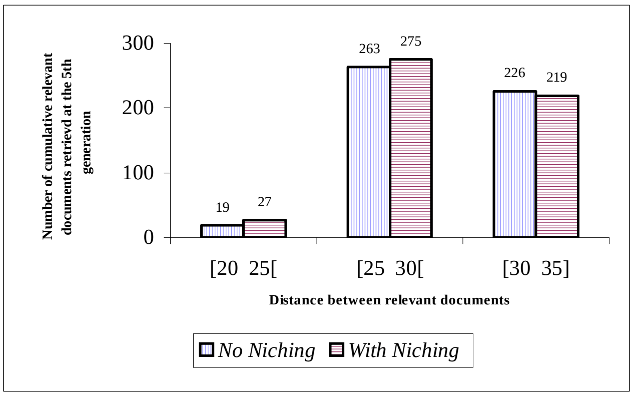

Figure 2 : Effect of the niching technique

Conclusions and future work

The experiments reported in this paper are promising. The results presented show the effectiveness of our GA approach for query optimisation in information retrieval and allow us to conclude on the interesting use of genetic heuristics to improve the results. More precisely, we focused on the effectiveness of using niching technique to recall relevant documents in various regions of the document space, knowledge based operators to guide the retrieval process by exploiting effective retrieval techniques and virtual individuals to improve the convergence conditions of the GA.

Future work should consist of more experiments on very large collections, as well as more elaborated experiments comparing the effectiveness of various niching and merging formula.

Bibliography

[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,1998

[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, 1997

[4] M. Boughanem, C. Chrisment & L.Tamine, Genetic Approach to Query Space Exploration. Information Retrieval Journal volume 1 N°=3 , pp175-192,1999

[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, 1971.

[6] H.Chen Machine learning for Information Retrieval : Neural Networks, Symbolic Learning and Genetic Algorithms, JASIS 46(3) : 194-216, 1995

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

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

[9] 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

[10] Gutman L. What is Who What in Statistics. The Statistician, 26 : 81 :107,1978

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

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

[13] Holland J : Les Algorithmes Génétiques , Revue Pour La Science, n°=179 pp 44-51, 1992

[14] 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, 1991

[15] 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, 1995

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

[17] 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

[18] 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, 1997

[19] 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

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

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

[22] L.Tamine & M. Boughanem : Query Optimisation Using an Improved Genetic Algorithm, Conference on Information knowledge and Management , Washington 6-11 November 2000

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

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