A study on using genetic niching for query optimisation in document retrieval

Mohand Boughanem, Lynda Tamine

A study on using genetic niching for query optimisation in document retrieval

Mohand Boughanem [1] Lynda Tamine [2]

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

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

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

Abstract. This paper presents a new genetic approach for query optimisation in document retrieval. The main contribution of the paper is to show the effectiveness of the genetic niching technique to reach multiple relevant regions of the document space. Moreover, suitable merging procedures have been proposed in order to improve the retrieval evaluation. Experimental results obtained using a TREC sub-collection indicate that the proposed approach is promising for applications.

KEY WORDS: Information retrieval , multiple query evaluation, genetic algorithm, niching

1. Introduction

The web is becoming a universal repository of human knowledge, which has allowed unprecedent sharing of ideas and information in a very large scale. As an immediate consequence, the area of information retrieval has grown well beyond its primary goal of indexing text and searching for useful document in a collection. Nowadays, research in information retrieval includes modelling, system architecture, data visualisation, etc.

The focus of our study is on the retrieval process of an information retrieval system using query operations. In fact, as observed with web search engines, the users might need to reformulate their queries in the hope of retrieving additional useful documents. Several approaches for improving the user query formulation have been proposed in information retrieval area. The approaches are grouped into two main categories. In the first category, relevance feedback methods are used for query expansion and term reweighting [22], [24] ,[21].

In the second category, the global approach is based on information derived from the context of the document retrieved. Two main strategies have been proposed: local clustering [1], [28] and global analysis [20] [25] or a combination of both local and global context [18].

In this work, we propose a strategy for multiple query reformulation using both relevance feedback techniques and context query improvement methods. More precisely, we exploit genetic techniques to handle the process of query optimisation.

Genetic Algorithms (GA) can be viewed as search procedures that try to find in a solution search space S, a solution s^* that maximise a function f called the fitness function. GA use some principle of natural selection and genetics [13]. The GA processes a population of individuals that evolve according to crossover and mutation operators.

Genetic techniques processing query optimisation have been proposed by several authors.

Gordon [12] adopted 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 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.

Yang & Korfhage [29] proposed a GA for query optimisation by reweighting 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 showed that the queries converge to their relevant documents after six generations.

Kraft & al [16] apply GA programming in order to improve the weighted Boolean query formulations. Their first experiments showed that the GA programming is a viable method for deriving good queries.

Horng & Yeh [15] propose a novel approach to automatically retrieve keywords and then uses genetic techniques to tune the keywords weights. The effectiveness of the approach is demonstrated by comparing the results obtained to those using a PAT-tree based approach.

These diffrent works show that the genetic approach is suitable for query optimisation. However, there is still some open questions:

- How to elleviate the genetic query drift in order to reach multiple relevant regions of the document space?
- How to define the optimal strategy of combination results?

In this work, we address these questions. Indeed, our goal is to exploit a suitable genetic technique for solving multimodal problems, named niching [11], [17]. Rather than processing a traditional GA which finally generates a unique optimal query corresponding to similar descriptors of assumed relevant documents, the integration of the niching method will tune the genetic exploration in direction of the multiple relevant documents. Furthermore, we propose some utilities to perform the merging of evaluation results.

The remaining of the paper is organised as follows. Section 2 gives an introduction of genetic niching techniques. Section 3 gives the main principles of our approach for query optimisation. Section 4 presents the results and discussion of experiments carried out on a sub-collection of TREC.

2. Multiomodal optimisation using genetic niching

GA is stochastic optimisation methods based on principles of evolution and heredity [13]. A GA maintains a population of potential solutions to a given optimisation problem. Each individual is defined using a genotype corresponding to its structure characteristics and also a phenotype corresponding to it's meaning representation in the context of the current optimisation problem. The population of individuals is renewed at each generation using both a fitness measure to evaluate the individuals quality and genetic transformations to reproduce the fittest ones. The children of each generation are produced using selection, crossover and mutation operators. At the termination of the process, a classical GA produces a unique optimal solution corresponding to the fittest individual produced at the last generation.

However, the goal of a multimodal optimisation process is to find multiple and diverse optima across the search space of a given problem. Convergence may occur to some degree within local regions but diversity must prevail across the most prominent regions. But, it is well known in GA theory that the selection pressure causes the phenomena of *genetic drift* which corresponds to the convergence in local regions. Thus, various techniques for reducing the selection pressure have been proposed [2], [11], [9] but are not overly selective as they generally enable to reach geographically close solutions.

Dejong [8] has proposed another technique based on an iterative execution of the GA.Using the assumption that the probabilities of reaching the multiple optima are equal, the number of executions required is computed using the following formula:

$$p * \sum_{i=1}^{p} \frac{1}{i} \cong p * (\alpha + \log p)$$

p : number of optima

α = 0.577, Euler constant

However, this method gives bad results in real life applications [26].

In this study, we restrict our efforts on niching techniques. Various other techniques for promoting genetic diversity are presented in [17], [14]. A niching method is based on the formation of subpopulations which explore different regions of the search space.We present in the following, the most common approaches.

2.1. Sequential niching

The approach is based on a sequential location of multiple niches using an iterative run of a traditional GA.Beasly & al [3] present a sophisticated strategy where at the end of each run, the algorithm proposed depresses the fitness function at all points with a certain radius of the fittest solutions. This transformation encourages the optimisation process to explore other area of the search space.

2.2. Ecological niching

This approach is based on the creation and exploitation of multiple environments of evolution. The basic theory of the ecological niching approach propose a simultaneously coevolution of subpopulations of individuals which are implicitly able to use food resources. Individuals that are unable to properly use resources die. Thus, the environment varies over time in its distribution of food resources, but individuals that are geographically close tend to experience the same environment [17]. The sharing [10] and clearing techniques [19] presented below are based on this ecological inspiration.

2.2.1. Sharing technique. Goldberg & Richardson [10] presented an implementation of the concept known as the *sharing method.* In this study, each individual in a niche can consume a fraction of the available resources: the greater the population size of the niche, the smaller the fraction. This leads towards a steady state in which subpopulation sizes are proportional to the amount of the corresponding available resources. The general formula of sharing fitness function is the following [10]:

$$f'(x) = \frac{f(x)}{\sum\limits_{y \in Pop} sh(dist(x,y))}$$

x,y : individuals of the population Pop
f(x) : initial fitness function
sh(dist(x,y)) : sharing function

The sharing function depends on the distance between two individuals of the population. The simplified version is the following form [10]:

$$sh(dist(x,y)) = \begin{cases} 1 - \left(\dfrac{dist(x,y)}{\delta_{sh}} \right)^{\alpha} & if\ dist(x,y) < \delta_{sh} \\ O & otherwise \end{cases}$$

α : constant
δ_{sh} : dissimilarity threshold

The distance function can be defined in the genotypic or phenotypic space search [9] or their combination [14].

Mahfoud [17] applied the principle of perfect discrimination of the niches which has two main consequences:

- each individual in a given niche, regardless of the distance measure , is always closer to every individual of its own niche than to any individual of another niche,
- the difference measure is able to determine whether two individuals are members of the same niche.

The author concludes that the sharing technique is most effective in cases of no overlap niches.

2.2.2. Clearing technique. The clearing technique [19] is a niching method based on the sharing ecological inspiration. It is applied after evaluating the fitness of individuals and before applying the selection operator.Like the sharing method, the clearing algorithm uses a dissimilarity measure between individuals to determinate if they belong to the same subpopulation or not. In contrast, the clearing procedure fully attributes the whole resource of a niche to a single individual: the winner. The winner takes all rather than sharing resources with the other individuals of the same niche.

Comparatively to the sharing technique, the complexity of the clearing procedure is lower and is more compatible with elitist strategies [19].

3. Our approach: genetic niching for query optimisation

The retrieval process as shown in figure 1, is based on an iterative feedback evaluation of query niches. A niche represents a set of individual queries exploring a specific region of the document space according to their evaluation results. The genotype representation of an individual 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

The phenotype of an individual query is traduced by its evaluation results in the IRS. 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 ($N^{(s)}$)
   Endfor
   Until a fixed number of feedback iterations
End
```

3.1. The niching method

In the current study, we applied the sharing technique to build the niches. Our choice is motivated by the fact that we attempt to explore widely the document space.We hope that the analysis of our first experiments using this technique will give us suitable utilities in order to exploit in the future, other niching techniques like the clearing one.

Regardless of the niching method used, the fitness function must be correlated with the standard goodness measure in IR that is average and precision. Considering this characteristic,we propose two distinct fitness function formulations. Each one is related to a specific strategy of formation of the niches.

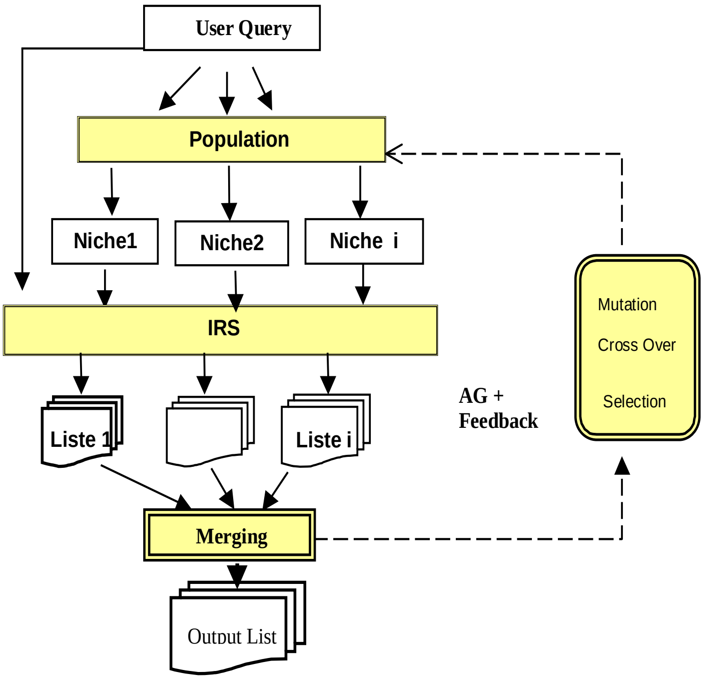

Figure 1: **The genetic retrieval process**

3.1.1. Niching using genotypic sharing. In this case, a niche is a set of individual queries having closed genotypes. The sharing function is the following:

$$sh(dist(Q_u^{(s)},Q_v^{(s)})) = \begin{cases} 1 \quad if \;\; dist(Q_u^{(s)},Q_v^{(s)}) < \;\; \delta \\ \qquad\qquad\qquad\qquad 0 \quad otherwise \end{cases}$$

$Q_u^{(s)}$: individual query at the generation s of the GA
dist : Euclidian distance
δ : niching threshold ($\delta > 0$)

The function has the following properties:

1. $0 \leq sh(dist(Q_u^{(s)}, Q_v^{(s)}) \leq 1$
2. $sh(0)=1$
3. $\lim_{dist(Q_u^{(s)}, Q_v^{(s)}) \to \infty} sh(dist(Q_u^{(s)}, Q_v^{(s)})) = 0$

Furthermore, the niches are perfectly distinct.
The fitness function is computed using the formula:

$$Fitness(Q_u^{(s)}) = \frac{QFitness(Q_u^{(s)})}{\sum_{Q_v^{(s)} \in Pop} sh(dist(Q_u^{(s)}, Q_v^{(s)}))}$$

where :

$$QFitness(Q_u^{(s)}) = \frac{\frac{1}{|Dr|} * \Sigma_{dr \in Dr} J(dr, Q_u^{(s)})}{\frac{1}{|Dnr|} * \Sigma_{dnrDnr} J(dnr, Q_u^{(s)})}$$

dr: relevant document
dnr: irrelevant document
Dr: set of relevant documents retrieved across the GA generations
Dr: set of irrelevant documents retrieved across the GA generations
J(D_j , $Q_u^{(s)}$): Jaccard measure

3.1.2. Niching using phenotypic sharing. In this case, the formation of the niches is based on the results (the documents retrieved) of their individual query members rather on their genotypic similarity. The niche structure is defined according to the coniche operators as following:

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

$Q_u^{(s)}$: indivudial query 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

In order to maintain distinct niches, we assume to affect an individual query once, to the niche of lower capacity. The fitness function is computed using a formula built on the Guttaman model:

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

J: Jaccard measure
$Dr^{(s)}$: 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

3.2. Genetic operators

The genetic operators defined in our approach [27] are not classical ones as they are not based on the basic structure proposed in GA theory [11]. They have been adopted 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. The selection procedure is based on a roulette wheel selection. Crossover and mutation perform a query reformulation using both feedback technique and local context information. The crossover is applied to a pair of individuals that are selected in the same niche, according to the crossover probability *Pc.* The mutation is applied to an individual query according to a mutation *Pm*. It consists essentially of reweighting a query term using a relevance measure formula.

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 the whole ones retrieved 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 formula.

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:

$$Rel_{Ni}^{(s)}(dj) = \frac{1}{|Ni|} \sum_{Q_u^{(s)}\in Ni} RSV(Q_u^{(s)},dj)$$

$RSV(Q_u^{(s)},d)$: RSV (Retrieval Status Value) of the document 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:

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

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

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 is more considered 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:

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

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 were carried out on a sub-collection of TREC-4 corpus. The documents we used are the AP88 newswire. We used 24 queries of TREC-4 (query numbered 1-24). The experiments were run using the Mercure IRS [4] that process the spreading activation technique. Because of the multiple iteration aspect of the search and the use of relevance judgements,the results reported in the paper are based on a residual ranking evaluation [7].

Prior experiments [5] allowed us to evaluate the main parameters of the GA: crossover and mutation probability. The best performances have been reached for respectively the following values: 0.7, 0.07 and then were chosen for all the remaining experiments presented in this paper.

4.1. Effect of the genetic query optimisation

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

- the first one 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 Mercure IRS

In order to make sens to our comparative evaluation, we consider that an iterative single query evaluation process may be based on the scanning of the overall output list, beginning from the top in direction of 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.

Finally, we compare the retrieval performance of residual lists issued from the same iteration of both single query evaluation and genetic combination process.

Table 1 presents 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 1: Retrieval performances

Table 2 provides a summary of the performance due to our proposed approach measured by the improvement achieved comparatively 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 2: Improvements of the genetic approach

As the tables illustrate,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. 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.

4.2. Comparative evaluation of the sharing techniques

This experiment compares the sharing techniques proposed. We report in table 3 the number of relevant documents in top 15 retrieved at each iteration of the GA and cumulative number of relevant documents retrieved at that point, using both genotypic sharing and phenotypic sharing.

	Iter1	**Iter2**	**Iter3**	**Iter4**	**Iter5**
Genotypic sharing	177(177)	114(291)	93(384)	69(453)	56(510)
Improvement	38%	41%	24%	25%	22%
Phenotypic sharing	180(180)	88(268)	97(366)	75(442)	78(520)
Improvement	63%	32%	28%	25%	26%

Table 3 : Comparative evaluation of the sharing techniques

Table 3 reveals that the phenotypic sharing technique is more effective than the genotypic one. More precisely, the cumulative number of relevant documents retrieved at the fifth generation of the GA is 510 using the genotypic sharing and 520 using the phenotypic sharing. The number of relevant documents retrieved by iteration is also generally higher in the case of using the phenotypic sharing.

These results are according with previous analyses presented in (Mahfoud, 1995) (Talbi, 1999) on the goodness of the phenotypic sharing technique. The main reason might be due to *the meaning distance* between the genotypic individual representation and its significant phenotypic one.

4.3. 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. In order to evaluate its 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[.

Table 4 shows, for each bin, the cumulative number of relevant documents retrieved at the fifth generation of the GA.

	[20 25[	**[25 30[**	**[30 35[**
No niching	19	263	226
With niching	27	275	219
Improvement	42%	45%	-3%

Table 4: Effect of the niching technique

It can be seen that niching 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 becomes slow.

Considering this assumption, we have developed this experimentation by running the 6th iteration of the GA for especially the third bin of queries. Table 5 shows the effect of the niching technique on the cumulative number in the top 15 retrieved at this iteration.

Query number	No niching	With niching
22	64	*83*
11	41	*40*
25	14	*14*
10	40	*40*
16	6	*9*
12	37	*33*
21	9	*10*
17	55	*53*
14	16	*14*
	282	*296*

Table 5: Effect of niching at the 6th iteration of the GA

We notice clearly that the results are better when using niching technique at the following iteration of the GA (4,9 % of improvement). This suggests that in order to increase the convergence of the GA, it might be interesting to use more suitable combination between the coniche operator definition and prior user relevance judgements.

Conclusion

In this paper, we have described a genetic approach for query optimisation in information retrieval. This approach takes into account the relevance multimodality problem in document retrieval by using an interactive retrieval process based on niching technique.

Prior experiments have been performed on TREC6 comparing genetic query evaluation and single pass search equivalent to Rocchio type search (Boughanem & al, 2000). The results have shown that the genetic approach is more effective particularly to improve recall.

We have showed in this study, that adding niching technique associated with suitable merging formula improves the exploration of the document space. Indeed, the approach has been applied to a sub-collection of TREC4 with success.

Additional work is certainly necessary to analyze the evolution of the niches structure across the GA generations in order to improve the merging procedures.

Finally, we believe that genetic niching provide interesting possibilities to solve the issue of relevance optimisation multimodality in document retrieval.

References

1. A. Attar & S. Franenckel (1977). Local Feedback in Full Text Retrieval Systems. Journal of the ACM, 397-417, 1977
2. J E.Baker (1985). Adaptive Selection Methods for Genetic Algorithm, in Proceedings of the first International Conference on Genetic Algorithm (ICGA) pp 101-111
3. D. Beasly, D.R Bull & R. R Martin (1993). A sequential niche technique for multimodal function optimization, Evolutionary Computation, 1(2) : pp 101-125
4. M. Boughanem (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
5. M. Boughanem, C. Chrisment & L.Tamine (1999). Genetic Approach to Query Space Exploration. Information Retrieval Journal volume 1 N°3 , pp175-192
6. 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
7. Chang Y K, Cirillo G C and Razon J (1971). Evaluation of feedback retrieval using modified freezing, residual collections and test and control groups. In: the Smart retrieval system: Experiments in automatic document processig, Prentice Hall Inc, chap 17, pp 355-370

8. K. A Dejong (1975). An analysis of the behavior of a class of genetic adaptive systems, Doctocal dissertation University of Michigan,. Dissertation abstracts International 36 (10), 5140B. University Microfilms N°76-9381

9. C.M Fonseca & P. J Fleming (1995). Multi-objective genetic algorithms made easy: selection, sharing and mating restrictions, In IEEE International Conference in Engineering Systems: Innovations and Application, pp 45-52, Sheffield, UK

10. Goldberg D.E & Richardson (1987). Genetic algorithms with sharing for multimodal function optimization, in Proceedings of the second International Conference on Genetic Algorithm (ICGA) , pp 41-49

11. Goldberg D.E (1989) : Genetic Algorithms in Search, Optimisation and Machine Learning, Edition Addison Wesley 1989

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

13. Holland J. (1962). Concerning Efficicent Adaptive Systems.In M.C Yovits, G.T Jacobi, &G.D Goldstein(Eds) Self Organizing Systems pp 215-230 Washinton : Spartan Books, 1962

14. J. Horn (1997). The nature of niching : Genetic algorithms and the evolution of optimal cooperative populations, PhD thesis, university of Illinois at Urbana, Champaign

15. 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

16. 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

17. Mahfoud S. W (1995). Niching methods for genetic algorithms, PhD thesis, university of Illinois at Urbana, Champaign, 1995

18. R. Mandala, T. Tokunaga & H. Takana. Combining multiple evidence from different types of thesaurus for query expansion, In Proceedings of the 22 th Annual International ACM SIGIR, Conference on research and development in information retrieval, August 1999, Buckley USA

19. Petrowski A. (1997) . A clearing procedure as a niching method for genetic algorithms. In the Proceedings of the IEE International Conference on Evolutionary Computation (ICEC), Nagoya, Japan

20. Y. Qiu & H.P. Frei, (1993). Concept Based Query Expansion. In Proceedings of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval, 160-169, Pittsburg, USA 1993

21. S. Robertson, S. Walker & M.M Hnackock Beaulieu (1995): Large test collection experiments on an operational interactive system: Okapi at TREC, in Informatio Processing and Management (IPM) journal, pp 260-345.

22. Rocchio(1971). Relevance Feedback in Information Retrieval, in The Smart System Experiments in Automatic Document Processing, G.Salton, Editor, Prentice-Hall, Inc., Englewood Cliffs, NJ, pp 313-23, 1971

23. G. Salton (1968). Automatic Information and Retrieval, Mcgrawhill Book Company, N. Y., 1968

24. G. Salton & C.Buckley (1990). Improving Retrieval Performance By Relevance Feedback, Journal of The American Society for Information Science, Vol. 41, N°4, pp 288-297, 1990

25. Schutze H.& Pedersen J. (1997). A Cooccurrence- Based Thesaurus and two Applications to Information Retrieval, Information Processing & Management, 33(3) : pp 307-318, 1997

26. E.G Talbi (1999). Métaheuristiques pour l'optimisation combinatoire multi-objectifs : Etat de l'art, Rapport CNET (France Telecom) Octobre 1999

27. L. Tamine & M. Boughanem (20001). Un algorithme génétique spécifique à une évaluation multi-requêtes dans un système de recherche d'information, journal Information Intelligence et Interaction, volume 1 n°=1, september 2001

28. J. Xu & W.B. Croft (1996). Query Expansion Using Local and Global Document Analysis. In Proc. ACM SIGIR Annual Conference on Research and Development, Zurich, 1996

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