QUERY OPTIMISATION USING AN IMPROVED GENETIC ALGORITHM
Lynda Tamine, Mohand Boughanem

To cite this version:

HAL Id: hal-00359566
https://hal.archives-ouvertes.fr/hal-00359566
Submitted on 8 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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] [5].

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 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.

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.

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 Pop0
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

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. They used a selection operator based on a stochastic sample, a blind crossover at two crossing points, and a classical mutation. The crossover probability is defined as 0.6. They 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 stage; the GA manipulates input documents and their associated keywords to generate an initial set of optimised concepts. 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 stage; the GA manipulates input documents and their associated keywords to generate an initial set of optimised concepts. The experiments showed that the queries converge to their relevant documents after six generations.

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.

- **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
- \( t_i \) : \( i \)th term
- \( n_t \) : Number of documents containing term \( t_i \)
- \( d_j \) : \( j \)th document
- \( d_j \) : term frequency of \( t_i \) in \( d_j \)
- \( t_i \) : term weight of \( t_i \) in \( d_j \)
- \( Q_{G}^{(s)} \) : query individual at the generation \( (s) \) of the GA
- \( Q_{G}^{ui} \) : weight of the term \( t_i \) in \( Q_{G}^{(s)} \)
- \( P_{pop} \) : Population at the generation \( (s) \) of the GA
- \( \bar{D}_{pop} \) : set of relevant documents retrieved by \( pop^{(s)} \)
- \( \bar{D}_{pop}^{(s)} \) : set of non relevant documents retrieved by \( pop^{(s)} \)
- \( D_{pop}^{(s)} \) : the \( L \) top documents retrieved by \( Q_{G}^{ui} \)
- \( RSV(Q_{D}) \) : Retrieval Status Value of the document \( D \) when submitting the individual query \( Q_{ui}^{(s)} \)
- \( Min\_Size \) : size of a niche
- \( Coniche \) : niche limit
- \( N_j^{(s)} \) : the min number of common documents retrieved by queries of the same niche
- \( N_j^{(s)} \) : size of \( N_j \)
- \( Average\_Fitness \) : 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
4. For each niche of the population do the search
5. Build the local list of documents
6. Build a merged list
7. Renew the niches
8. Judge the top fifteen documents
9. Compute the fitness of each individual query
10. for each niche of the population

#### Repeat
- parent1 = Selection (\( N_j^{(s)} \))
- parent2 = Selection (\( N_j^{(s)} \))
- Crossover (\( Pc \), parent1, parent2, son)
- Mutation (\( Pm \), son, sonmut)
- Add_Niche (sonmut, \( N_j^{(s+1)} \))

#### Until
- \( N_j^{(s+1)} \) = \( N_j^{(s)} \)
- 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:

\[ Q_i = (q_{i1}, q_{i2}, ..., q_{in}) \]

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:

\[ q_{ui}(s) = \frac{1}{\sum_i ((1+\log(fa_i)) \log(\frac{N}{ni}))} \]

- 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:

\[ Q_i \rightarrow Q_j \iff (D_{Q_i}(s)/L_i) \cap (D_{Q_j}(s)/L_j) > Coniche_Limit \]  

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 which 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]:

\[ \text{Fitness}(Q_i(s)) = \sum_{d \in \text{Dis阅v}} J(Q_i(s),d) - J(Q_i(s),d) \]

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.

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:

\[ \text{Relevance}(D) = \frac{1}{|N|} \sum_{Q \in N} RSV(Q, D) \]

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:
4.1. Experiments

The experiments correspond to iteration 0 in our algorithm, and the feedback the document rankings produced by both the initial query, which method, all the documents previously judged are removed from on a residual ranking evaluation [5]. This method is used to of relevance judgement, the results reported in the paper are based 

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.

4.3. 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.

<table>
<thead>
<tr>
<th>Iter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_S=2</td>
<td>3</td>
<td>9</td>
<td>15</td>
<td>3</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>C_L</td>
<td>3</td>
<td>9</td>
<td>15</td>
<td>3</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>C_L</td>
<td>3</td>
<td>9</td>
<td>15</td>
<td>3</td>
<td>9</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 1: 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 smaller number of relevant documents.
We notice that for the first iteration corresponding to the GA processing at all the other succeeded iterations, processing recall a greater number of relevant documents than NoGA. The followed iterations did not perform better. However, the GA processing recall a greater number of relevant documents than NoGA 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.

<table>
<thead>
<tr>
<th>Iter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knl. Op</td>
<td>94(94)</td>
<td>62(156)</td>
<td>40(196)</td>
<td>59(255)</td>
<td>48(303)</td>
<td>45(348)</td>
</tr>
<tr>
<td>Bld. Op</td>
<td>91(91)</td>
<td>43(139)</td>
<td>36(169)</td>
<td>40(209)</td>
<td>27(235)</td>
<td>30(266)</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Iter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mrg. based on fit. val</td>
<td>94(94)</td>
<td>62(156)</td>
<td>40(196)</td>
<td>59(255)</td>
<td>48(303)</td>
<td>45(348)</td>
</tr>
<tr>
<td>Mrg. based on rank. val</td>
<td>91(91)</td>
<td>71(162)</td>
<td>50(212)</td>
<td>41(253)</td>
<td>47(300)</td>
<td>40(339)</td>
</tr>
</tbody>
</table>

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


