2009 Third Asia International Conference on Modelling & Simulation

Query Optimization in Relevance Feedback using Hybrid GA-PSO for Effective Web Information Retrieval

Siti Nurkhadijah Aishah Ibrahim, Ali Selamat, Mohd Hafiz Selamat
Intelligent Software System Research Laboratory (ISSLab),
Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.
echoas1306@yahoo.com, aselamat@utm.my, mhafiz@utm.my

Abstract

Due to the rapid growth of web pages available on the Internet recently, searching a relevant and up-to-date information has become a crucial issue. Conventional search engines use heuristics to determine which web pages are the best match for a given keyword. Results are obtained from a database that is located at their local server to provide fast searching. However, to search for the relevant and related information needed is still difficult and tedious. By using the genetic algorithm (GA) in relevance feedback, this paper presents a model of hybrid GA-Particle Swarm Optimization (HGAPSO) based query optimization for Web information retrieval. We expanded the keywords to produce the new keywords that are related to the user search. Experimental results demonstrate that it is very effective to improve the search of the relevant web pages using the HGAPSO.

1. Introduction

Being the most promising information source in the world, the World Wide Web(WWW) is still expanding rapidly. To date, millions of people are seeking information on it, and search engines play a very important role during this process. Search engines aims to process the enormous information in some collection of document then create an index for quick search. Basically, the index is an inverted file that maps each word in the collection to the set of documents containing that word [1]. Normally, the processing step is to improve the performance of the collection such as removing the noise words(stop words; e.g. and, I, was, etc.), the conation of words via stemming and/or the use of thesauri, and the use of the word weighting schemes such as term frequency (TF).

As the internet technology explodes, the world is moving directly to the evolution of information technology. Many types of information from all over the world are at our fingertips. However, according to the Google search engine statistics, until 2008 almost 1 trillion web pages have been indexed. Therefore, the process of information searching and retrieval will take a while and it is difficult for users to get the relevant information. Search engines such as Google, AltaVista, Yahoo!, etc. have been introduced and used to assist in searching the relevant information from the web.

Unfortunately, the current commercial information retrieval system that is usually based on the Boolean information retrieval model has provided unsatisfactory results. There exist three main areas of GA applications to information retrieval: document indexing, clustering and query optimization. The third group of applications, query optimization, is the most frequently discussed. What they all have in common is the use of the GA to perform the technique of relevance feedback.

In addition, most of the current search engines take up an enormous amount of bandwidth and are time consuming while crawling the web pages. Thus, by using the genetic algorithm in relevance feedback, this paper presents a model of hybrid GAPSO (HGAPSO) based query optimization for effective Web information retrieval. We expand the keywords to produce new keywords that are related to the user search and present more results to users. Experimental results demonstrate that it is very effective to improve the search of the relevant web pages using the HGAPSO.

The rest of the paper is structured as follows. Section 2 presents the model of our HGAPSO. Section 3 explains the relevance feedback and HGAPSO based query optimization. Section 4 shows the experimental results. Finally, section 5 describes our discussion and conclusion.

978-0-7695-3648-4/09 $25.00 © 2009 IEEE
DOI 10.1109/AMS.2009.95

2. The Model of Hybrid Genetic Algorithm - Particle Swarm Optimization (HGAPSO)

2.1. Genetic Algorithm (GA)

Currently, the increasing number researches done in the information retrieval field and the developing interest towards applying artificial intelligent tools such as machine learning into the field show that there are a lot of efforts have been put aiming to solve the web searching problems. One of the most focused areas with a considerable growth is evolutionary computation (EC). EC is based on the use of models of evolutionary process for the design and implementation of computer-based problem solving systems [2].

As claimed by Z. Zhu et. al [2], a classical and very important technique in EC is genetic algorithm (GA). GA is not specifically a learning algorithm but it offers a powerful and domain-independent searching ability that can be used in many learning tasks, since learning and self-organization can be considered as optimization problems in many cases.

GA is not new to information retrieval. So, it is not surprising to see many applications of GA into Information Retrievak (IR) [2][3][4][5] recently. As explained in section 2, there exist three main areas of GA applications into information retrieval: document indexing, clustering and query optimization. The third group of applications, query optimization, is the most frequently discussed. What they all have in common is the use of a GA to perform the technique of relevance feedback.

GA is a probabilistic algorithm that simulates the natural selection mechanism of living organism and is often used to solve problems that require expensive solutions. The search space in GA is composed of candidate solutions to the problem whereby each of the candidate is represented by a string known as chromosome. Fitness function is an objective function for each chromosome in GA. A group of chromosome with their fitness function is known as population. At every iteration, this population is called a generation. Figure 1 shows the basic GA process.

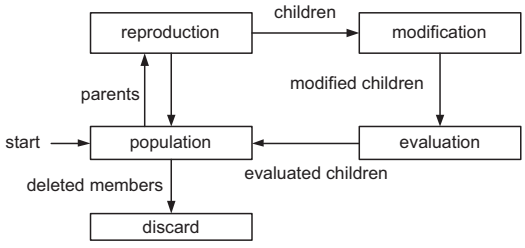

Figure 1. Basic Genetic Algorithm Process

2.2. Particle Swarm Optimization (PSO)

PSO is an evolutionary computation method, which is clearly different from other evolutionary-type methods that does not use the filtering operation (such as crossover and/or mutation) and the members of the whole population are maintained through the search procedure [6]. In order to find an optimal or near-optimal solution to the problem, PSO updates the current generation of particles (each particle is a candidate solution to the problem) using the information about the best solution obtained by each particle and the entire population. Each particle has a set of attributes: current velocity, current position, the best position discovered by the particle so far and, the best position discovered by the particle and its neighbors so far. Each particles start with randomly initialized velocities and positions. Then the *n*th component of the new velocity and the new position for the *i*th particle are updated by using the following equations:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (pbest_i - s_i^k) + c_2 r_2 (gbest^k - s_i^k) \quad (1)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (2)$$

where ω is the inertia weight; c1 and c2 are random numbers, *pbest* is the best position discovered so far by the corresponding particle and *gbest* is the best particle found so far within the neighbours. In a binary space, the v_i is a probability that must be constrained to the interval [0, 1]. A logistic transformation v_i can be used to accomplish this last modification. The resulting change in position then is defined by the following rule:

$$if(rand() < s(v_i))\, then\ s_i = 1; else\ s_i = 0 \quad (3)$$

where the function $s(v_i)$ is a sigmoid limiting transformation and rand() is a quasi-random number selected from a uniform distribution in [0, 1].

PSO aims to share information among individuals of a population. In PSO algorithms, search is conducted by using a population of particles, corresponding to individuals as in the case of evolutionary algorithms. Compared to GA, PSO has no operator of natural evolution which is used to generate new solutions for future generation. Instead, PSO is based on the exchange of information between individuals, so called particles, of the population, so called swarm [6]. Each particle also adjusts its own position based on its previous experience and towards the best previous position obtained in the swarm. Memorizing its best own position establishes the particles experience implying a local search along with global search emerging from the neighboring experience or the experience of the whole swarm.

From [6], there are two variants of the PSO algorithm were developed, one with a global neighbourhood, and other one with a local neighborhood. "In the global neighbourhood, each particle moves towards its best previous position and towards the best particle in the whole swarm, called *gbest* model. On the other hand, according to the local variant, called *lbest* model, each particle moves towards its best previous position and towards the best particle in its restricted neighbourhood" [6]. Since PSO was first introduced by [6], it has been successfully applied to optimize various continuous nonlinear functions [7][8][9].

2.3. HGAPSO

Figure 2 shows the flow of HGAPSO and below are the details of the proposed HGAPSO:

2.4. Population and Chromosomes

In this paper, the chromosomes from the document are represented directly. The weight vector of the document or a query in each test collection is directly encoded as a chromosome. Thus, a chromosome is represented as a weight vector $w = (w_1, w_2, ..., w_n)$ where w_i is a real number and denotes the weight of the keyword t_i for *i=1, 2,...,n*. Each gene represents a weight of a keyword. If $w_i = 0$, it means the keyword t_i is not included in the chromosome. The initial population is then divided into two and feed into GA and PSO algorithm to proceed to the next process.

$$w = \{1 \ if w > 0; 0 \, if w = 0 \tag{4}$$

After the new population have been generated from GA and PSO, they are collected and the genetic operators such as crossover and mutation are executed, respectively. Below is the pseudocode of HGAPSO:

Pseudocode for the HGAPSO

```
Initialize parameters

begin
Initialize populations
    Chromosome one from GA
    Chromosome two from PSO

    Crossover P(g)
    (exchange the information
    between GA and PSO populations)
    Mutate P(g)
    Evaluate P(g)
  end while
end
```

2.5. Fitness Function

Fitness function is the measure of the quality of an individual. It should be designed to provide assessment of the performance of an individual in the current population. In the application of a genetic algorithm to information retrieval, one has to provide an evaluation or fitness function for each problem to be solved. The fitness function must be suited to the problem at hand because its choice is crucial for the genetic algorithm to function well.

In HGAPSO, jaccard coefficient is used in the overall process to measure the fitness of a given representation. The total fitness for a given representation is computed as the average of the similarity coefficient for each of the training queries against a given document representation. Document representation evolves as described above by genetic operators (e.g. crossover and mutation). Basically, the average similarity coefficient of all queries and all document representations should increase.

$$fitness(d_j) = \frac{1}{n} \cdot \sum_{k=1}^{n} \frac{|d_j \bigcap d_q|}{|d_j \bigcup d_q|} \tag{5}$$

2.6. Genetic Operators

After an initial population is formed, a genetic algorithm always uses its genetic operators with configurable probabilities to generate offspring based on the current population. Once a new generation has been created, the genetic process is repeated iteratively until a problem solution is found.

Selection The selection procedure for GA is based on a variant of the usual roulette wheel selection. 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 Random point crossover has been selected to be used in this paper. The simplest methods choose one or more points in the chromosome to mark as the crossover points. Then the parameters between these points are merely swapped between the two parents. Let say we have two parents,

$parent_1 = a_1, a_2, a_3, ..., a_n$
$parent_2 = b_1, b_2, b_3, ..., b_n$

The crossover points are randomly selected and then the parameters in between are exchanged:

$$parent_1 = a_1, a_2, \downarrow b_3 \downarrow, ..., a_n$$
$$parent_2 = b_1, b_2, \uparrow a_3 \uparrow, ..., b_n$$

Mutation In this paper, we use random mutations whereby it alter a small percentage of the bits in the list of chromosomes. Mutation is the second way a genetic algorithm explores a cost surface. It can introduce traits that are not in the original population and keeps the genetic algorithm from converging too fast.

To combine the GA with PSO, the basic elements of PSO algorithm [7] are summarized as follows:

- *Particle:* x_i^k is a candidate solution *i* in swarm at iteration *k*. The i^{th} particle of the swarm is represented by a *d*-dimensional vector and can be defined as $x_i^k = [x_{i1}^k, x_{i2}^k, ..., x_{id}^k]$, where *x*s are the optimized parameters and x_{id}^k is the position of the i^{th} particle with respect to d^{th} dimension. In other words, it is the value d^{th} optimized parameter in the i^{th} candidate solution.
- *Population:* pop^k is the set of *n* particles in the swarm at iteration *k*, i.e. $pop^k = [x_1^k, x_2^k, ..., x_n^k]$.
- *Particle velocity:* v_i^k is the velocity of particle *i* at iteration *k*. It can be described as $v_i^k = [v_{i1}^k, v_{i2}^k, ..., v_{id}^k]$, where v_{id}^k is the velocity with respect to d^{th} dimension.
- *Particle best:* PB_i^k is the best value of the particle *i* obtained until iteration *k*. The best position associated with the best fitness value of the particle *i* obtained so far is called particle best and defined as $PB_i^k = [pb_{i1}^k, pb_{i2}^k, ..., pb_{id}^k]$ with the fitness function $\mathrm{F}\left(PB_i^k\right)$.
- *Global best:* GB^k is the best position among all particles in the swarm, which is achieved so far and can be expressed as $GB^k = [gb_1^k, gb_2^k, ..., gb_d^k]$ with the fitness function $\mathrm{f}\left(GB^k\right)$.
- *Termination criterion:* it is a condition that the search process will be terminated. In this study, search is terminated when the number of iteration reaches a predetermined value, called maximum number of iteration.

3. Relevance Feedback and HGAPSO based Query Optimization

Users of the online search engines often find it difficult and tedious to express their need for information in the form of a query. However, if the user can identify examples of the kind of documents that they require then they can employ a technique known as relevance feedback. Relevance

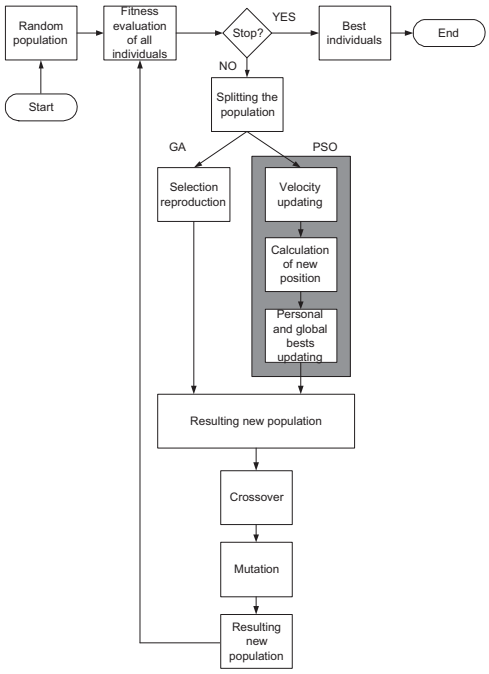

Figure 2. The Flow of HGAPSO

feedback covers a range of techniques intended to improve a users query and facilitate retrieval of information relevant to a users information need.

The technique of relevance feedback has been designed to produce an improved query formulation following an initial retrieval operation. It is an automatic process introduced over thirty years ago and is one of the most popular methods of query optimization [10]. According to [10], the classical Dec-hi method is one of the best traditional relevance feedback methods. The method is very simple yet very effective. The primary idea of Ide Dec-hi method consists of adding directly to the weights of the original query those all the relevant documents of the set of documents provided for feedback and subtracting from them those of the first irrelevant document obtained in the retrieval that belongs to the said set. The query vector is reformulated as follows:

$$Q' = Q + \sum_{all\ relevant} D_i - S \tag{6}$$

where Q is the vector of the original query, D_i is the vector of the relevant document *i* and S is the vector of the first irrelevant document of the ranking.

3.1. Term Vectorization and Document Representation

Vector space model (VSM) is one of the most widely used models in the application of GAs into information retrieval. In this paper, VSM has been chosen as a model to describe documents and queries in the test collections. Let say, we have a dictionary, *D*;

$$D = (t_1, t_2, ..., t_i) \tag{7}$$

where *i* is the number of distinguished keywords in the dictionary. Each document in the collection is described as an *i*-dimensional weight vector;

$$d = (w_1, w_2, ..., w_i) \tag{8}$$

where w_j represents the weight of the *j*th keyword t_j for *j=1, 2,..., i* and is calculated by the Term Frequency Inverse Document Frequency (TFIDF) method. Each query in the collection is also described as a weight vector, *q*;

$$q = (u_1, u_2, ..., u_i) \tag{9}$$

where u_j represents the weight of the *j*th keyword t_j for *j=1, 2,..., i* and is calculated by the Term Frequency (TF) method.

To determine the document terms, we used the following procedure:

- Applying the extraction of all the words from each document.
- Eliminating the stop-words using the stop-words list.
- Stemming the remaining words using the Porter stemmer; commonly used stemmer in English.
- Extracting the words appear in each documents (TF) and storing it as keywords in the database.

4. Experimental Results

In this paper, user will search their interest topic through query optimization search system. After user enters the keyword, the system will search the term related to that keyword from the database. Then, the result will be presented to the user. From the interface, a user will select interest topic that is most related to the keyword entered before. After that, the keyword will be arranged in an array to represent the chromosome in binary so that the fitness value for each documents can be calculated. Document with high fitness value will be picked in the selection operation. The process flow in our system is listed below:

1. User enters query into the system.

Table 1. Example of User Queries

$Queries$	$Information$
Q1	iskandar malaysia development
Q2	iskandar
Q3	iskandar malaysia
Q4	khazanah nasional
Q5	iskandar johor open

2. Match the user query with list of keywords in the database.
3. Results without GA are represented to the users.
4. Users select the relevant results found by the system.
5. Encode the documents retrieved by user selected query to chromosomes (initial population).
6. Then, the chromosomes will be processed by the HGAPSO and the new population will be generated.
7. Population feed into genetic operator process such as selection, crossover and mutation.
8. Step 5 is repeated until maximum generation is reached. Then, get an optimize query chromosome for document retrieval.
9. Decode optimize query chromosome to query and retrieve new document (with GA process) from database.

We evaluate the result based on the system classification and user judgment. They are calculated based on precision, recall, and F1. Precision (P) in this system is defined as the percent of retrieved documents that are relevant to the user search. Recall (R) is defined as the fraction of the documents that are relevant to the query successfully searched by the system. For F1, it is defined as a measure of a test's accuracy.

5. Discussion and Conclusion

In this paper, we have shown how an evolutionary algorithm can help to reformulate a user query to improve the results of the corresponding search. The evolutionary algorithm is in charge of selecting the appropriate combination of terms for the new query. To do this, the algorithm uses fitness function, a measure of the proximity between the query terms selected in the considered individual. Then, the top ranked documents are retrieved using these terms. We have carried out some experiments to have an idea of the possible improvement that the HGAPSO can achieve. In these experiments, we have used the precision obtained

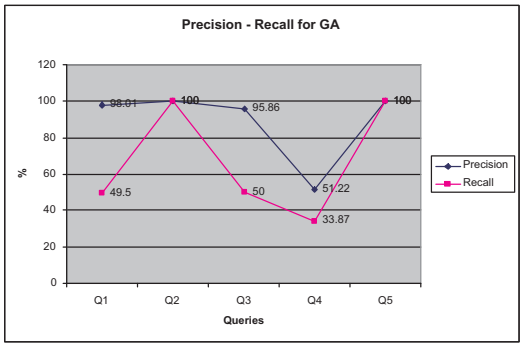

Figure 3. Precision - Recall for GA

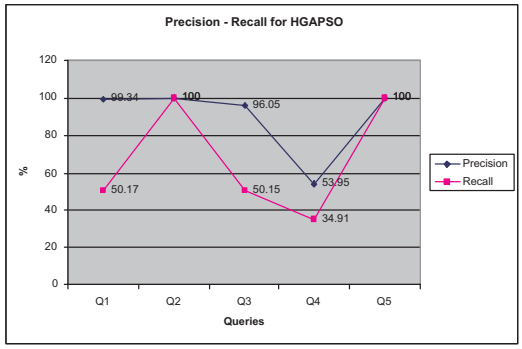

Figure 4. Precision - Recall for HGAPSO

from the user relevance judgement as fitness function. Results have shown that in this case, the HGAPSO achieve a slight improvement compared to the conventional GA. However, we want to emphasize that this feedback mechanism improves the search system by considering users suggestions concerning the found documents, which leads to a new query using HGAPSO. In the new search stage, more relevant documents are given to the user.

As a conclusion, since this is still a work in progress, the results are not really satisfying even though the proposed model was improved. In the future, we hope that the system can be improved further and the results can achieve higher accuracy rate in solving the data mining problems.

5.1. Acknowledgment

The authors wish to thank the reviewers for their helpful suggestions. Also thanks to all current and previous members of the Intelligence Software Engineering Lab (ISELab) at the Universiti Teknologi Malaysia (UTM). This work is supported by the Ministry of Science & Technology and Innovation (MOSTI), Malaysia and Research Management Centre, Universiti Teknologi Malaysia (UTM) under the Vot 79267.

References

[1] Menczer, F. and Belew, R. K. 2000. Adaptive Retrieval Agents: Internalizing Local Contextand Scaling up to the Web. Mach. Learn. 39, 2-3 (May. 2000), 203-242.

[2] Zhengyu Zhu, Xinghuan Chen, Qingsheng Zhu, Qihong Xie: A GA-based query optimization method for web information retrieval. Applied Mathematics and Computation 185(2): 919-930 (2007).

[3] M. Koorangi, K. Zamanifar, A distributed agent based web search using a genetic algorithm, IJCSNS, International journal of computer science.

[4] Horng, J.-T., Yeh, C.-C.: Applying genetic algorithms to query optimization in document retrieval. Inf. Process. Manage. 36(5), 737759 (2000).

[5] Lopez-Pujalte, C., Bote, V.P.G., de Moya Anegon, F.: A test of genetic algorithms in relevance feedback. Inf. Process. Manage. 38(6), 793805 (2002).

[6] Eberhart, R. C. and Shi, Y. 1998. Comparison between Genetic Algorithms and Particle Swarm Optimization. In Proceedings of the 7th international Conference on Evolutionary Programming VII (March 25 - 27, 1998). V. W. Porto, N. Saravanan, D. E. Waagen, and A. E. Eiben, Eds. Lecture Notes In Computer Science, vol. 1447. Springer-Verlag, London, 611-616.

[7] M. Fatih Tasgetiren, Yun-Chia Liang. A Binary Particle Swarm Optimization Algorithm for Lot Sizing Problem. Journal of Economic and Social Research 5 (2), 1-20.

[8] Song Liangtu, Zhang Xiaoming. Web Text Feature Extraction with Particle Swarm Optimization. IJCSNS International Journal of Computer Science and Network Security, Vol. 7 No. 6 pp. 132-136.

[9] Shi, X.H.; Wan, L.M.; Lee, H.P.; Yang, X.W.; Wang, L.M.; Liang, Y.C., "An improved genetic algorithm with variable population-size and a PSO-GA based hybrid evolutionary algorithm," Machine Learning and Cybernetics, 2003 International Conference on , vol.3, no., pp. 1735-1740 Vol.3, 2-5 Nov. 2003.

[10] G. Salton, C. Buckley. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41 (4) (1990), 288-297.