Web Information Retrieval using Particle Swarm Optimization based Approaches

Habiba Drias
Department of Computer Science
USTHB, LRIA
Algiers, Algeria

***Abstract*—** When dealing with large scale applications, data sets are huge and very often not obvious to tackle with traditional approaches. In web information retrieval, the greater the number of documents to be searched, the more powerful approach required. In this work, we develop document search processes based on particle swarm optimization and show that they improve the performance of information retrieval in the web context. Two novel PSO algorithms namely PSO1-IR and PSO2-IR are designed for this purpose. Extensive experiments were performed on CACM and RCV1 collections. The achieved results exhibit the superiority of PSO2-IR on all the others in terms of scalability while yielding comparable quality.

Keywords; web information retrieval; scalability; bio-inspired approach; swarm intelligence; PSO;

I. INTRODUCTION

Information retrieval has shown its great importance throughout the history of computer science. It has been widely used and has played a central role especially in large database management. In the Internet era, its interest is increasing more and more as it is considered as the core of many web applications such as web services discovery and B2B e-commerce. The complexity of information retrieval induced by the huge volumes of documents necessitates innovative tools to cope with the problem. Many directions of research are contributing in handling this spiny difficulty. Distributed information retrieval and Personalizing Information Source Selection are examples of these research axes. The recent works are considering the users and sources profiles in order to restrict the search only to the sources that have the same profile as the user [8],[9]. In this manner, a lot of information is pruned and therefore, the respond time of such systems becomes rapid.

In this work, we are considering bio-inspired approaches and more precisely particle swarm optimization (PSO) algorithms as another alternative to palliate the web information retrieval complexity issue. The PSO meta-heuristic belongs to the class of direct search methods and was introduced for the first time in [11]. It is a stochastic population–based approach and is a part of the swarm intelligence. The modeling of this paradigm is founded on social-psychological behaviors where particles are directed by their instinct while they are also reactive for their environment.

The PSO framework is simpler to implement than those of the other evolutionary approaches while at the same time it can present a more powerful tool. Its methodology deals only with the changing of particle positions while moving in the solutions space and on their evaluation with the fitness function. For information retrieval where the solutions space is a collection of documents, it is not possible to build the whole documents from the terms of the collection for the purpose of searching for the most similar document with the query. The meta-heuristics where solutions are built from their basic elements as in genetic algorithms are inefficient for such problem. PSO presents interesting features in handling such situation by avoiding such solution construction. It merely searches for documents and does not take into account the terms while moving from one document to another one.

Recently, a few studies on information retrieval have been interested to artificial intelligence techniques like genetic algorithms [13], neural network and symbolic learning [7] and swarm optimization [3], [4], [5], knowing that these approaches are powerful enough when they are adapted judiciously. The problematic treated in such works is very different from that one handled in this paper. In fact our concern is the design of a search engine that is, developing a search process for retrieving documents whereas in the evoked articles, the authors are attempting to improve and optimize the similarity function and information retrieval learning mechanism.

The idea behind addressing large scale information retrieval with a PSO-based approach is the pruning of the prohibitive search space in a stochastic manner in order to browse documents in a reasonable amount of time. Many successful applications of PSO on public and industrial sectors like scheduling, biomedical, communication networks, financial, image and robotics have been performed. Motivated by the success and the power of this swarm intelligence approach, we have designed two PSO algorithms to explore collections of documents. The algorithms were tested on CACM and RCV1 collections and comparison with the traditional IR method is undertaken.

II. INFORMATION RETRIEVAL

The information retrieval (IR) problem consists in finding from a large collection, documents including information expressed in a query specifying user needs. Any IR process involves a matching mechanism between the query and the documents of the collection. Thus three important components are central in such environment:

- The document which can be a text, a web page, an image or a video. A document is usually represented by a set of terms or keywords extracted from its source.
- The query which represents a need expressed by a user and specified in a formalism adopted by the system.
- The similarity function that measures the degree of resemblance between a document and a query. Two system evaluations are widely used: the precision which is the fraction of retrieved documents that are relevant and the recall

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.225

which is the fraction of relevant documents that are retrieved.

A. Indexing process

In an IR system, an important step is the indexing process in which initial corpuses are transformed in an internal organization of the documents and the queries in order to access them in an efficient way. The documents and the queries must be described according to a model. Many models for IR are described in the literature among them are the Boolean model, the vector space model and the probabilistic model. The most widely used is the vector space model. In this model, documents as well as queries are represented as vectors of term weights. Each weight denotes the importance of the corresponding term in the document or in the query. The vector space is built during the indexing process and contains all the terms that the system encounters.

1) The inverted file

Consider the following vector space:

$$(t_1, t_2, t_3, \ldots, t_n)$$

where t_i is a term or keyword for $i=1$ *to* n. For each term, we consider a structure that contains all the documents that include the term. The weight of the term in the document is associated with the document in the list. The file containing all these structures is called the inverted file.

2) The collection of documents

The whole collection of documents is represented by a file containing all the documents, that is $C = (d_1, d_2, d_3, \ldots, d_m)$. The structure is indexed by the number of the document. Each element of C points towards a list containing all the terms of the document with their respective weight. The list is sorted according to the term identifier. The query is modeled exactly as a document.

The weight of a term in a document is computed using the expression *tf*idf* where *tf* represents the term frequency in the document and *idf* is the inverted frequency computed usually as follows:

$idf = log(m/df)$ where m represents the number of documents and df is the number of documents that contain the term. The component *tf* indicates the importance of the term for the document, while *idf* expresses the power of discrimination of this term. This way, a term having a high value of *tf *idf* is at the same time important in the document and less frequent in the others.

The weight for a term in a query is computed with the same manner. The similarity of a document d and a query q is then computed using one of the following formulas:

$sim(d, q) = \sum_i (a_i * b_i)$ *(internal product)*

$sim(d, q) = \sum_i (a_i * b_i) / (\sum_i (a_i)^2 * \sum_i (b_i)^2)^{1/2}$ *(Cosine)*

$sim(d, q) = 2 \sum_i (a_i * b_i) / (\sum_i (a_i)^2 + \sum_i (b_i)^2)$ *(Dice)*

$sim(d, q) = \sum_i (a_i * b_i) / (\sum_i (a_i)^2 + \sum_i (b_i)^2 - \sum_i (a_i * b_i))$ *(Jaccard)*

a_i and b_i are the weight of term t_i respectively in the document and in the query. The cosine formula is considered arbitrarily in the experimentations shown in section IV since no research result concerning the use of these formulas is provided in the literature. In [13] the authors propose a process based on genetic algorithm to develop the similarity function for optimization purposes.

B. Traditional Search Approaches

Classical approaches for information retrieval are exact and exhaustive methods. One obvious way to design an algorithm for IR is to browse the whole collection of documents and calculate the similarity between the document and the query. Generally, this algorithm called usually the naïve approach is not considered even for reasonable size of corpuses because there is a smarter way to address this problem. The idea is to execute the above algorithm on the inverted file instead of the whole collection of documents. Only documents that have at least one common term with the query are consulted and this way the complexity of the algorithm is reduced at a phenomenal rate.

It is clear that the complexity of the second approach is more interesting than the one of the naive approach because fewer documents are considered in the search process. For both approaches the principle is to search in the corresponding file for documents that are similar with the query. The consulted documents are sorted according to their similarity with the query. This process has a worst case complexity in $O(n*m)$ where n is the number of terms and m the number of documents. When n and m are reasonable, the inverted file technique is very efficient. However for an environment where the number of documents and the number of keywords are prohibitive, the complexity is exponential because the parameters n and m express exponential magnitudes. This is the reason why it is important to find another alternative for addressing information retrieval in such context. Meta-heuristics enables to get a polynomial response time at a higher computation scale but of course in the detriment of the solution quality. To enhance the latter every component of the search method must be very well thought and implemented and all the difficulty resides in this task. This is what is aimed at in this paper.

III. PSO-IR ALGORITHMS

The PSO process starts by creating at random particles and by initializing the best position for each particle and for the whole population. It then iterates the computation of the positions and velocities of each particle movement and the update of the local best position for each particle and the global best position for the entire swarm. More precisely, the process consists of a loop that computes for each particle two quantities: 1) its new velocity and its new position using its proper intuition expressed by the best position it has already determined and its instinctive feeling in the group denoted by the global best position found by the entire group or swarm and 2) the update of its best position. The global best position of the swarm is also updated inside the loop.

In this section, we present the particle swarm optimization algorithms called PSO1-IR and PSO2-IR designed for information retrieval. Let first start with the description of

the problem modeling.

A. Solutions coding

Unlike the other meta-heuristics applied to information retrieval, the PSO modeling is simpler. It takes into account only documents when computing particles positions and not their contents except for their evaluation. Documents are indexed according to the vector space model:

$C = (d_1, d_2, d_3, \ldots, d_m)$

$d = (t_1, t_2, t_3, \ldots, t_n)$

$q = (qt_1, qt_2, qt_3, \ldots, qt_h)$

C is the whole collection of m documents each one of them designated by an identifier which is a number, d is a document involving at most n terms and q a query containing up to h terms.

The solution is evaluated by the fitness function called f and expressed in IR as the similarity between the document and the query.

B. PSO1-IR and PSO2-IR

The first algorithm PSO1-IR considers as input the whole collection of documents. The position of a particle is a document identifier. When a particle moves from one position to another one, the search corresponds to crossing one document then another one. $v[k]$ determines the distance that separates the document visited currently by the particle k from the next one that it will be visited just after. This distance is calculated according to the corresponding formula of the loop of PSO algorithm. The next document to visit denoted $x[k]$ in the algorithm is then obtained by adding to the current document the freshly computed velocity. The sum operation is performed modulo the number of documents in order to not stop the search when the next document identifier is greater than the last document in the collection. This means that the topology of the particle movement network is a ring shaped graph. The algorithm PSO1-IR is then outlined as follows:

Algorithm 1. PSO1-IR

```
begin
    i=1;
    while (i<max-iter) do
        begin
            for each particle k do
                begin
                    rp = random value from [0, cmax];
                    rg = random value from [0, cmax];
                    v[k] = c1*v[k] + rp*(p[k] – x[k])+
                    rg*(g – x[k]);
                    x[k]= ( x[k] + v[k] ) mod (m) +1;
                    if f(x[k]) > f( p[k] ) then p[k] =
                    x[k];
                    if f(p[k]) > f( g) then g = p[k];
                end ;
            i = i+1;
        end;
end;
```

Where:

- i is the loop counter.
- $max\text{-}iter$ is the maximum number of iterations determined by experiments.
- rp, rg: positive random numbers representing the confidence coefficients.
- $cmax$ is defined by the formula: $cmax= ((c_1+1)*(c_1+1))/2$.
- c_1 is the inertial factor set to 2.
- g is the best solution found by the swarm.
- $p[k]$ is the best solution found by the particle k.
- f is the fitness function calculated by the similarity function.

A considerable improvement can be gained if the search space is restricted to the inverted file. PSO2-IR is a second version of the algorithm that is executed on the reduced input size. This algorithm is effective only in the case where the inverted file is available.

IV. EXPERIMENTAL RESULTS

The designed algorithms have been implemented in Java on a personal computer and have undergone a series of extensive experiments. The empirical parameters that yield high solutions quality such as the number of particles and the maximum number of iterations were set by experiments prior to the algorithms performance evaluation tests.

A. Used benchmarks

We have conducted experiments on two collections that are described in the following.

1) CACM

CACM is a collection of article abstracts published in ACM journal between 1958 and 1979. This collection has been used in many research works, but remains too small to show the income brought by the algorithms presented in this paper. It contains 3204 documents and near 6468 terms. The average document size is 2Kbytes.

2) RCV1

Reuters Corpus Volume I (RCV1) is a collection of 804414 documents representing archives published by Reuters and 47236 terms. The average document size is 2Kbytes.

The empirical parameter tests have yielded the following values: for CACM 50 and 20, for RCV1 300 and 80 respectively for max-iter and #particule.

C. Results and Comparison

A good performance of the designed algorithms implies a high similarity of the best found document and a runtime lower than the one yielded by the exact method. This remark has leaded us to use the ratio between the similarity and the runtime to evaluate the algorithm efficiency. A high value of this ratio in comparison with the optimal solution translates a reduction in

running time and a similarity relatively stable.

Since we are comparing algorithms for the search process of information retrieval, we do not need to express the performance criteria in terms of the precision and recall metrics. The latter serve to test the whole system performance and not just retrieval algorithms. This is the reason why all the results we will present below are established on the basis of the similarity and execution time criteria.

The traditional approach has been implemented in order to get the optimal solution. Besides in the experiments we have considered the algorithm that has the most interesting features from the design point of view that is PSO2-IR. PSO1-IR algorithm can be considered as an alternative in the case when the inverted file is not available.

1) Results for CACM

The results of the comparison of PSO2-IR with the inverted file classical approach are shown in Table I. Through these results we observe clearly that PSO2-IR is more interesting than the classical algorithm even for a small collection like CACM.

TABLE I. PSO2-IR VERSUS THE CLASSICAL APPROACH

	query	Achieved document	sim	Time (ms)	sim/ time
optimal	1	2945	1.6	10.3	0.16
	2	1030	1	4.5	0.22
	3	3012	1.18	12.1	0.09
	4	1621	43.37	3.2	1.05
PSO-IR2	1	2945	1.63	3.3	0.49
	2	137	1	1.6	0.62
	3	1629	1	2.1	0.48
	4	3142	3.09	1.7	1.82

2) Results for RCV1

The same experiments as those for CACM have been performed on RCV1 where the number of documents is more than 250 times larger. Table II shows the numerical results obtained from retrieving documents for 4 queries.

TABLE II. PSO2-IR VERSUS THE EXACT ALGORITHM FOR RCV1.

	query	Achieved document	sim	sim/ time
optimal	1	4040	0.31	0
	2	93576	0.62	0
	3	511263	0.79	0
	4	536823	0.56	0
PSO-IR2	1	4040	0.31	0.31
	2	697610	0.63	0.01
	3	382463	0.77	0.02
	4	536823	0.56	0.01

VI. Conclusions

In this paper, we have developed two different innovative particle swarm optimization algorithms called PSO1-IR, PSO2-IR for information retrieval. The algorithms consider as search space respectively the whole collection of documents and the inverted file.

Through the experiments we have performed, we have shown the robustness and superiority of the designed algorithms on the classical approaches. The experimental results have shown a significant reduction in response time relatively to the classical approaches. Finally increasing scalability levels are reached when applying respectively PSO1-IR and PSO2-IR.

REFERENCES

[1] R. Baeza-Yates, B. Ribiero-Neto, “Modern Information Retrieval,” Addison Wesley Longman Publishing Co. Inc., (1999)

[2] E. Bonabeau, M. Dorigo, G. Theraulaz: Swarm Intellignce from natural to artificial systems, Oxford University Press, 1999

[3] X. Cui, T.E. Potok, P. Palathingal: document clustering using particle swarm optimization, IEEE Swarm Intelligence Symposium, pp. 185-191 (2005)

[4] E. Diaz-Aviles, W. Nejdl, L. Schmidt-Thieme: Swarming to rank for information Retrieval, Gecco, ACM, pp. 9-16 (2009)

[5] H. Drias, H. Mosteghanemi: Bees Swarm Optimization based Approach for Web Information Retrieval, IEEE/WIC/ACM: pp 6-13, (2010)

[6] A. P. Engelbrecht: Fundamentals of Computaional Swarm Intelligence, Wiley (2005)

[7] C. Hsinchun: Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning and Genetic Algorith, Journal of the American Society for Information Science. Vol 46, n° 3, pp. 194-216, (1995)

[8] S. Kechid, H. Drias: Personalizing the Source Selection and the Result Merging Process. International Journal on Artificial Intelligence Tools 18(2): pp. 331-354 (2009)

[9] S. Kechid, H. Drias: Mutli-agent System for Personalizing Information Source Selection. Web Intelligence, pp. 588-595 (2009)

[10] J. Kennedy, R.C. Eberhart: Swarm Intelligence, Morgan Kaufmann (2001)

[11] J. Kennedy, R.C. Eberhart: Particle Swarm Optimization. In Proc of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942-1948 (1995)

[12] C.D. Manning, P. Raghavan, H. Schutze: Introduction to Information Retrieval, Cambridge University Press, (2008)

[13] P. Pathak, M. Gordon, W. Fan: Effective Information Retrieval using Genetic Algorithms based Matching Functions Adaptation,33rd IEEE HICSS, vol.1 8 pages (2000)