International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

An Evolving Approach on Efficient Web Crawler using Fuzzy Genetic Algorithm

P.Jaganathan[1], T. Karthikeyan[2]

[1]Professor and Head, Department of Computer application, PSNA College of Engineering and Technology, Dindigul-624 622, India

[2]Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore - 641 046, India

Abstract: ***Today, Web has become one of the largest and most readily accessible repositories and a rich resource of human knowledge. The traditional search engines index only surface Web whose pages are easily found. The focus has now been moved to invisible Web or hidden Web, which consists of a large warehouse of useful data such as images, sounds, presentations and many other types of media. To use such data, there is a need for specialized technique to locate those sites as we do with search engines. This paper focuses on an effective design of a Web Crawler that can optimally discover pages from the by employing fuzzy based genetic algorithm. The fitness function was evaluated using fuzzy fitness finder to produce best population for each iteration. A framework is used to discover the resource discovery problem and the results show the improvement in the crawling strategy and harvest rate.***

Keywords: Web crawler, Fuzzy, Web Crawler, Genetic algorithm, Search engine

1. Introduction

A web crawler (also known as a robot or a spider) is a system for the bulk downloading of web pages. Web crawlers are used for a variety of purposes. A Web crawler is a key component inside a search engine[15]. Most prominently, they are one of the main components of web search engines, systems that assemble a corpus of web pages, index them, and allow users to issue queries against the index and find the web pages that match the queries. A related use is web archiving where large sets of web pages are periodically collected and archived for posterity. A third use is web data mining, where web pages are analyzed for statistical properties, or where data analytics is performed on them. Finally, web monitoring services allow their clients to submit standing queries, or triggers, and they continuously crawl the web and notify clients of pages that match those queries. The web crawlers lies in the fact that the web is not a centrally managed repository of information, but rather consists of hundreds of millions of independent web content providers, each one providing their own services, and many competing with one another.

There are several research problems of information retrieval, far from optimization such as guiding user in order to determine one's needs, the analysis of people's way of using and processing information, accumulating a package of information that facilitates the user to come closer to a solution, representing Knowledge, the ways of processing knowledge/information, the human computer interface for better information retrieval, a better user-enhanced information systems design and an optimal method to evaluate an information retrieval system. The purpose of this paper is to produce an optimized web crawler which overcomes the vagueness in the determination of web link resources effectively with less cost and time.

2. Related Work

The entire internet lies on the search engine and more than 85% of the users use search engines to find their information [13]. This section discusses about the some of the existing techniques for web crawling. Breadth first is well suited for situations where the objective is found on the shallower parts in a deeper tree. It will not perform so well when the branches are so many in a game tree, especially like chess game and also when all the path leads to the same objective with the same length of the path [1] [2].

Andy yoo et al [3] proposed a distributed BFS for numerous branches using Poisson random graphs and achieved high scalability through a set of clever memory and communication optimizations.

Shaojie Qiao [4] proposed a new page rank algorithm based on similarity measure from the vector space model, called SimRank, to score web pages. They proposed a new similarity measure to compute the similarity of pages and apply it to partition a web database into several web social networks (WSNs)

Yongbin Qin and Daoyun Xu [5] proposed an algorithm, taking the human factor into consideration, to introduce page belief recommendation mechanism and brought forward a balanced rank algorithm based on Page Rank and page belief recommendation which ultimately attaches importance into the subjective needs of the users; so that it can effectively avoid topic drift problems. Tian Chong [6] proposed a new type of algorithm of page ranking by combining classified tree with static algorithm of PageRank, which enables the classified tree to be constructed according to a large number of users' similar searching results, and can obviously reduce the problem of Theme-Drift, caused by using PageRank only, and problem of outdated web pages and increase the efficiency and effectiveness of search.

In [7] proposed a novel approach - genetic based apriori algorithm for web crawling to discover interesting patterns or relationship between data in large database Association rule mining is used. Association rule mining can be an

important data analysis method to discover associated web pages. The Apriori algorithm is a proficient algorithm for determining all frequent web pages. The Frequent web pages form the Association rules. The proposed method yields promising results compared to the ordinary apriori algorithm and we present empirical results to substantiate this claim.

In the paper [8], they use a genetic algorithm with focused crawling for improving its crawling performance. Expands initial keywords by using a genetic algorithm for focused crawling. The results showed that our approach could build domain-specific collections with higher quality than traditional focused crawling techniques.

Bing Liu et al combined the crawling strategy with clustering concepts [9]. For each topic they first retrieve a specific number of top weighted retrieved pages from Google for that topic and then extract some other keywords from them.

3. Materials and Methods

Genetic Algorithm

Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas [12]. Genetic algorithms (GAs) begin with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. Further, solutions are selected according to their fitness to form new solutions, that is, offsprings. The above process is repeated until some condition is satisfied.

Algorithm: Genetic Algorithm

Procedure GA(n, χ, μ)
n : number of individuals in the population;
χ : is the fraction of the population to be replaced by crossover in each iteration;
μ : the mutation rate
// Initialise generation 0:
k := 0;
Pk := a population of n randomly-generated individuals;
// Evaluate Pk:
Compute fitness(i) for each i ∈ Pk;
do
{ // Create generation k + 1:
// 1. Copy:
Select (1 − χ) × n members of Pk and insert into Pk+1;
// 2. Crossover:
Select χ× n members of Pk; pair them up; produce offspring; insert the offspring into Pk+1;
// 3. Mutate:
Select μ × n members of Pk+1; invert a randomly-selected bit in each;
// Evaluate Pk+1:
Compute fitness(i) for each i ∈ Pk;
// Increment:
k := k + 1;
}
while fitness of fittest individual in Pk is not high enough;
return the fittest individual from Pk;

4. Proposed Framework

The problem which exists in the traditional focused crawler URL analysis model described previously is that the local optimal solution is often easily given in the process of searching the relevant pages according to the predetermined theme, namely only crawling around the related web pages, which results in some related web pages which are linked together through hyperlinks with lower degree of relevance are not crawled, then an effective coverage of the focused crawler reduces cost and time.

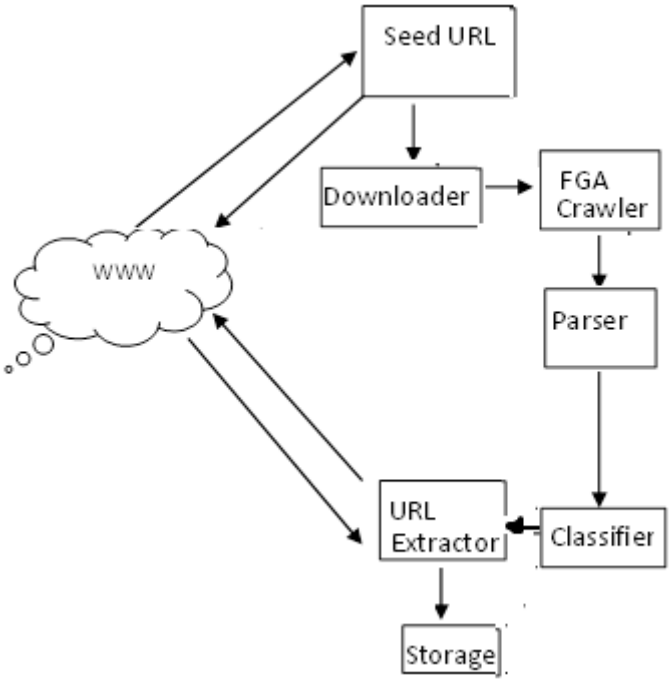


Figure 1: Proposed Framework

5. Proposed Architecture of Fuzzy genetic Crawler

The proposed Fuzzy genetic algorithm Crawler shown in the above figure overwhelms the traditional genetic algorithm in which the population of the genetic algorithm undergoes evolution at every generation, the relatively 'good' solutions reproduce while the relatively 'bad' solutions die. To distinguish between solutions, an objective (evaluation) function is used. In the simple cases, there is only one criterion for optimization for example, maximization of profit or minimization of cost. But in many real-world decision making problems, there is a need for simultaneous optimization of multiple objectives.

A single objective optimization model cannot serve the purpose of a fitness measuring index because we are looking at multiple criteria that could be responsible for stringing together data items into clusters. This is true; not only for the clustering problem but for any problem solving using GA that involves multiple criteria. In multi-criteria optimization,

the notion of optimality is not clearly defined. In the classical method of objective weighting, each criterion is given a weight and the weighted sum of the fitness of individual criteria is taken as the overall objective function. The weights are a measure of the significance of a criterion in comparison with the other criteria. For example if one weight is two times the other, it implies that the former criterion is doubly significant compared to the latter. However, a set of fixed weights leads to a constant interrelationship. If values for one or more of these criteria are not available, then the total fitness score will drop drastically. Actually, some of the other criteria may be enough to give a high combined score. Under different combinations of inputs, the weight values can vary. It is here that fuzzy logic comes to the rescue for it specializes in comparing apples and oranges. In a fuzzy-logic based system, a weighting function is replaced by a set of rules. A criterion can take a whole range of values in an interval. The criterion, if modeled as a fuzzy variable can belong to different classes in the interval with different probabilities. Varied combination of different criteria's in fuzzy inputs join together to produce unique effect. A fuzzy logic inferencing system can capture the complex interactions among the criteria which is not possible by assignment of fixed weights to the inputs.

6. The Fuzzy Fitness Finder

The GA calls upon the Fuzzy Fitness Finder (FFF) to evaluate the fitness of the solutions it creates in each population. A solution is a mapping of the whole data set into clusters. A cluster may have one or more data objects. The fitness of an individual is calculated as the summation of the fitness values of all its constituent clusters whose size is greater than one. A single unit cluster is assumed to have fitness equal to zero.

$$\text{Fitness} = \sum_{i=1}^{m} fitness_i$$

Where m = number of non-single clusters in an individual and i fitness is the fitness of the ith cluster of the individual. To find the values for i fitness, a fuzzy inferencing mechanism has been developed.
The FFF is used in two ways : -
i) to find the pair-wise fitness between adjacent data items
ii) to find the fitness of a set of data items (a cluster)

The same FFF can be applied to both cases. In the first case, the cluster size is equal to two and as it is between adjoining data items of the data set, these values can be calculated and stored at the outset. It may be recollected that these values are used by the mutation operator to adjust the cluster borders. The FFF calculates the fitness values of all non-single clusters of individuals in the initial population as well as in subsequent generations.
Number of calls to FFF for calculating the fitness of the

$$\text{population} = \sum_{i=1}^{m} M_i$$

where mi is the number of non single clusters in the i th individual and n is the size of the population. The fitness calculations are done for clusters of the initial population. In each new generation, variation is brought about by crossover and mutation. For crossover, only if the cut-point falls inside a cluster, the fitness values have to be recalculated for the disturbed clusters. Otherwise, it involves only a renumbering of clusters from the cut point till the end of the data set. Fitness values have to be re-calculated for clusters affected by mutation. Whenever cluster constitution changes, the FFF is called to recalculate the new cluster's fitness value. The FFF is a standard fuzzy logic based inference system used in engineering and control systems.

The elements of the FFF are
- Identification of input and output criteria
- Calculation of crisp values of the input criteria
- Fuzzification of input values
- Fuzzy Inference Engine
- Defuzzification of output values

The function of FFF is shown in the figure below

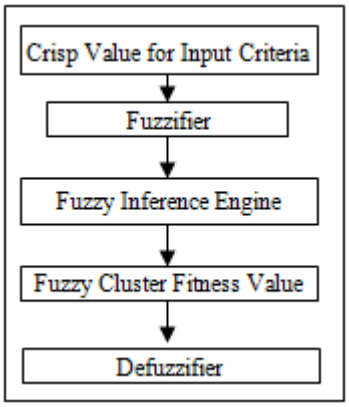


Figure 2: Function of FFF

Algorithm (Fuzzify)

begin
initialize // *input data and variables*
k = number of input variables
m [i], i = 1, k // *number of fuzzy sets of the kth variable*
inpval [i], i =1, k // *k input values*
lowval [i][j], i = 1,..,k , j = 1,..,m [i] // *lower limits of fuzzy sets*
midval [i][j], i = 1,..,k , j = 1,..,m [i] // *peak values of fuzzy sets*

```
highval [i][j], i = 1,..,k , j = 1,...,m [i] // upper limits of fuzzy sets
memb [i][j], i = 1,…,k, j = 1,…,m [i]
// memb[][] membership of ith variable in its jth fuzzy set
i = 0
do i = i + 1 // for all k variables
j = 0
do j = j + 1 // for each fuzzy set
if ( lowval [i][j] < inpval [i] < highval [i][j] ) then
if ( inpval [i] < = midval [i][j] ) then
memb [i][j] = ( inpval [i] - lowval [i][j])/ ( midval [i] - lowval [i][j])
else memb [i][j] = (highval [i] - inpval [i][j])/( highval [i] - midval [i][j])
until j = m [i]
until i = k
return memb[][] // membership functions
end
```

Algorithm (Fuzzy Inference Engine)

```
begin
initialize // input data and variables
k = 3 // number of input variables
m [i], i = 1, k // number of fuzzy sets of the kth variable
memb [i][j], i = 1,…,k, j = 1,…,m [i]
// memb[i][j] is the membership of ith variable in its jth fuzzy set
isetname[i][j], i = 1,…,k, j = 1,…,m [i] // input set names
// linguistic code e.g. HIGH, LOW
antecedents [i], consequent [i], i = 1,…,r // r rules
// antecedents concatenated e.g.HIGHLOW
// output e.g. LOW
osetname[i], i = 1,…, q // output set name
osetvalue [ii] = 0, ii = 1,…, q // output strength
i = 0
do i = i + 1 // for first variable
j = 0
do j = j + 1 // for second variable
k = 0
do k = k + 1 // for third variable
// look at all combinations
if (memb[1][i] > 0 and memb[2][j] > 0 and memb[3][k] > 0) then
begin
addsets = isetname [1][i] + isetname [2][j] + isetname [3][k]
p = 0
do p = p + 1
if (addsets = antecedents [p]) then
begin
poutput = min(memb[1][i], memb[2][j], memb[3][k])
if ( consequent [p] = osetname[ii] ) then //ii = 1,…,q
osetvalue[ii] = osetvalue[ii] + poutput*poutput
end
until p = r
end
until k = m [3]
until j = m [2]
until i = m[1]
osetvalue[ii] = sqrt(osetvalue[ii]) //ii = 1,…,q
```

7. Proposed Fuzzy Genetic Algorithm

By integrating genetic algorithm with fuzzy set [10], a fuzzy genetic algorithm [11] is a fuzzy set-coded genetic algorithm, where each individual (chromosome) is composed of a set of membership functions. In designing fuzzy genetic algorithms, issues in conventional genetic algorithms are put into fuzzy context and converted into fuzzy versions, for instance, fuzzy representation, fuzzy genetic operators, etc. In particular, fuzzy genetic algorithms need to consider the validation and ranking of the created fuzzy sets. In mining web links in a given url, we adapted the following fuzzy genetic algorithm framework. It deals with all basic issues such as initialization, selection, crossover, mutation and evaluation of web links in fuzzy context. For initialization, all individuals are sampled randomly within the valid domain.

Algorithm (fuzzy genetic algorithm)

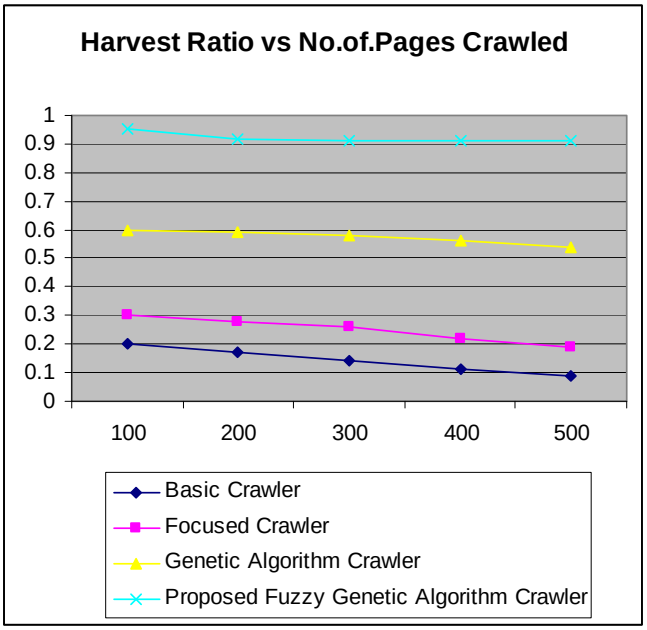
```
Input: real number set X
Output: optimal fuzzy set Y for decision support
Procedure: FGA(μ , X (t), X̃ (t) , X̃ '(t) ,Y )
//start with an initial time
t := 0;
//initialize a fuzzy random population of individual's X̃ (t) by fuzzifying the real number sets X (t) with proper membership functions μX̃ ,
initialize
X̃ (t) = {(x, μX̃(x))| x ∈ X̃ (t), μx, X (t) → [0,1]};
//evaluate the fitness of all initial individuals of population based on fuzzy evaluation
evaluate X̃ (t) ;
//test for termination criterion
While (not done) do
//increase the time counter
t := t + 1;
//select a fuzzy sub-population set X '(t) for offspring production
X̃ '(t) := select X (t) ;
//crossover the "genes" of the selected parents X '(t)
crossover X '(t) ;
//perturb the mated population stochastically
mutate X̃ '(t) ;
//fuzzily evaluate its new fitness
evaluate X '(t) ;
//select the survivors Y from actual fitness
Ỹ := survive X̃ (t), X̃ '(t) ;
End
//fuzzily rank the survivors
rank Ỹ ;
//defuzzify and export the final survivors
export Y ;
```

8. Experimental Result

The efficiency of Focused crawler is calculated by harvest rate. Harvest Rate measures the rate at which relevant pages are crawled and how effectively irrelevant pages are filtered off from the crawl

$$\text{Harvest ratio} = \frac{\text{No.of relevant web pages crawled}}{\text{Total no of web pages crawled}}$$

For better crawling performance, harvest ratio should be high. The harvest rate of the proposed Fuzzy genetic algorithm crawler is compared with simple crawler, focused crawler and genetic algorithm crawler. The 500 web pages crawled on topic Computer, Science, Arts and Sports, and average of harvest rate is taken. The figure below shows the harvest rate of simple crawler, focused crawler, priority based focused crawler and priority based semantic crawler. The overall performance gain over simple crawler is 95%, focused crawling is 75% and genetic algorithm crawler is 20%.



In the field of information retrieval, precision is the fraction of retrieved web pages that are relevant to the search.

$$\text{Precision} = \frac{(\text{relevant wepage}) \cap (\text{retrieved webpage})}{\text{retrieved webpage}}$$

Recall in information retrieval is the fraction of the web pages that are relevant to the query that are successfully retrieved.

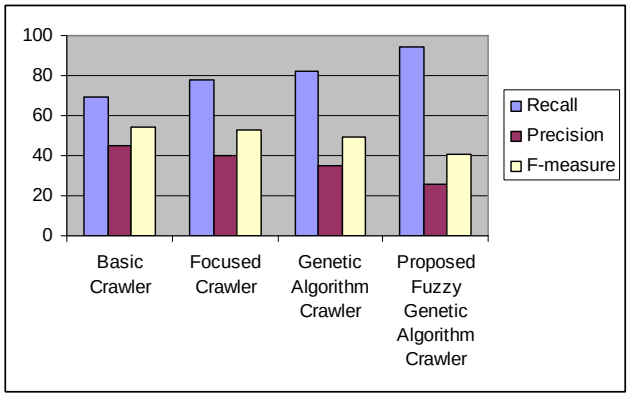
$$\text{Recall} = \frac{(\text{relevant wepage}) \cap (\text{retrieved webpage})}{\text{relevant webpage}}$$

A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table: Performance comparison of Proposed work with other existing approaches

Techniques	Recall	Precisio	F-measure
Basic Crawler	69	45	54.4737
Focused Crawler	78	40	52.8814
Genetic Algorithm	82	35	49.0598
Proposed Fuzzy Genetic	94	26	40.7333



The experiment shows that the focused crawler URL analysis model based on Fuzzy genetic algorithm proposed in this paper can improve accuracy rate, recall rate effectively, and avoid getting into the local optimal solution. Table 2 shows the comparison among crawlers.

9. Conclusion and Future Work

In this paper, a fuzzy based genetic algorithm crawler has been proposed that keeps all URLs to be visit based on the maximum optimization approach. The queue always returns URL with highest fitness value in each iterations. The performance of crawler is evaluated on topic Computer, Science, Arts and Sports. The experimental results show that the proposed FGA crawler gives 20% improved results over classical genetic algorithm, 75% improved results over simple focused crawler and 95% improved results over simple crawler. The proposed FGA method utilizes the membership value of each link fetched and overcomes the degree of vagueness is identification of relevant web links.

Corresponding Author

T. Karthikeyan
Research Scholar
Department of Research and Development Centre,
Bharathiar University,
Coimbatore - 641046, India
E-mail: karthik.rt@gmail.com

References

[1] Steven S. Skiena "The Algorithm design Manual" Second Edition, Springer Verlag London Limited, 2008, Pg 162.
[2] Ben Coppin "Artificial Intelligence illuminated" Jones and Barlett Publishers, 2004, Pg 77.
[3] Andy Yoo,Edmond Chow, Keith Henderson, William McLendon, Bruce Hendrickson, "A Scalable Distributed Parallel Breadth-First Search Algorithm on BlueGene/L" ACM 2005.
[4] Shaojie Qiao, Tianrui Li, Hong Li and Yan Zhu, Jing Peng, Jiangtao Qiu "SimRank: A Page Rank Approach based on similarity measure" 2010 IEEE
[5] Yongbin Qin and Daoyun Xu "A Balanced Rank Algorithm Based on PageRank and Page Belief recommendation"
[6] TIAN Chong "A Kind of Algorithm For Page Ranking

Based on Classified Tree In Search Engine" Proc International Conference on Computer Application and System Modeling (ICCASM 2010)

[7] J. Usharani, Dr. K. Iyakutti, Mining Association Rules for Web Crawling using Genetic Algorithm, International Journal Of Engineering And Computer Science ISSN:2319-7242, Volume2 Issue 8 August, 2013 Page No. 2635-2640

[8] Chain Singh , Ashish Kr. Luhach , Amitesh Kumar Improving Focused Crawling with Genetic Algorithms, International Journal of Computer Applications (0975 – 8887) Volume 66– No.4, March 2013

[9] Bing Liu, Chee Wee Chin, Hwee Tou Ng. "Mining Topic-Specific Concepts and Definitions on the Web" in proceeding WWW, May 20-24, Hungary, 2003 .

[10] L. Zadeh, "Fuzzy sets," Information and Control, 83: 338–353, 1965

[11] J. Buckley, Y. Hayashi, "Fuzzy genetic algorithm and applications," Fuzzy Sets and Systems, 61: 129-136, 1994

[12] L. Davis, (Ed.). Handbook of genetic algorithms. New York: Van Nostrand Reinhold, 1991

[13] S.Lawrence, C. L. Giles, "Accessibility of Information on the Web," Nature, 400, 107-109, 1999.

[14] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, Marco Gori, " Focused Crawling using Context Graphs," Proceedings of the 26th VLDB Conference, Cairo, p. 527–534, 2000.